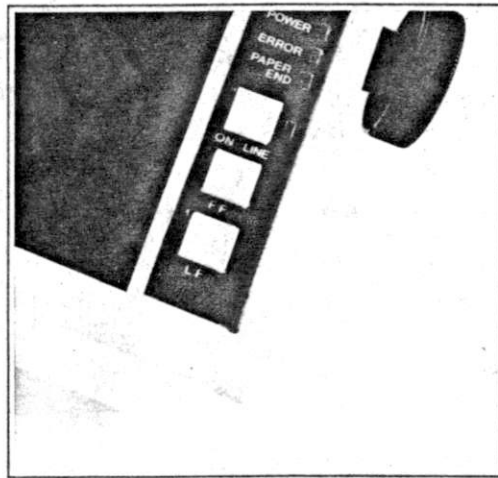


EPSON  
EPSON  
EPSON  
EPSON  
EPSON  
EPSON  
EPSON  
EPSON  
EPSON  
EPSON  
EPSON  
EPSON  
EPSON  
EPSON  
EPSON  
EPSON  
EPSON  
EPSON



EPSON AMERICA, INC.

# **User's Manual for MX-80 Printer**

**by  
David A. Lien**

**COMPUSOFT™ PUBLISHING**

**A Division of CompuSoft, Inc.**

**P.O. Box 19669 • San Diego, California 92119 U.S.A.**

This Book was prepared especially for Epson America, Inc. by CompuSoft Publishing. All rights, domestic and international are reserved by CompuSoft, Inc. Requests for permission to reproduce or distribute this User's Manual should be addressed to:

COPYRIGHT DEPT  
CompuSoft, Inc.  
P.O. Box 19669  
San Diego, CA 92119

*Copyright © 1980 by CompuSoft Publishing,  
A Division of CompuSoft, Inc.  
San Diego, CA 92119*

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without the prior written permission of the publisher. No patent liability is assumed with respect to the use of the information contained herein. While every precaution has been taken in the preparation of this book, the publisher assumes no responsibility for errors or omissions. Neither is any liability assumed for damages resulting from the use of the information contained herein.

CompuSoft is a registered trademark of CompuSoft, Inc.

\*TRS-80 is the registered trade mark of Radio Shack, a Division of Tandy Corporation.

International Standard Book Number: 0-932760-04-X  
Library of Congress Catalog Card Number: 80-70122

10 9 8 7 6 5 4 3 2 1

*Printed in Japan*

## **A Personal Note From the Author**

Congratulations on your decision to buy an EPSON MX-80 printer! In my opinion, it is the best dollar-value on the printer market today.

***“WHO NEEDS A LEARNER’S MANUAL FOR A PRINTER?”***  
The answer — ***ALMOST EVERYONE EXCEPT A COMPUTER PROFESSIONAL.***

Today’s printers are very sophisticated compared to those of even a year ago. Most are not fully utilized because the instructions are too vague and complicated. We’re doing our best to eliminate that problem with the EPSON MX Series.

I encourage you to learn ALL about your remarkable MX-80. You paid for it. Put it to work

Dr. David A Lien  
San Diego — 1980

# Table of Contents

Introduction .....	7
<b>Chapter 1:</b> The Starting Line .....	9
<b>Chapter 2:</b> Send it a Message .....	23
<b>Chapter 3:</b> More Print Control Commands .....	35
<b>Chapter 4:</b> Introduction to Graphics .....	41
<b>Chapter 5:</b> How Dot Matrix Printing Works .....	47
<b>Chapter 6:</b> Advanced Printing and Endless Form Control .....	55
<b>Chapter 7:</b> Advanced Graphics .....	63
<b>Chapter 8:</b> TRS-80 Mode and Misc .....	69
<b>Chapter 9:</b> Switching to Foreign Language Characters .....	71
<b>Appendix A:</b> ASCII Charts for MX-80 and TRS-80 Modes .....	79
<b>Appendix B:</b> Control Codes .....	81
<b>Appendix C:</b> Internal Switch Explanation Tables .....	83
<b>Appendix D:</b> Use with Other Computers .....	87
<b>Appendix E:</b> Use with a Radio Shack printer cable .....	91
<b>Appendix F:</b> Special considerations for Model II TRS-80 .....	93
<b>Appendix G:</b> Character and Graphic Fonts .....	95
<b>Appendix H:</b> Technical Specifications .....	99
<b>Appendix I:</b> Function Codes .....	101
<b>Appendix J:</b> Logic Block Diagram .....	103
<b>Appendix K:</b> Pinout Signal Chart .....	105
<b>Appendix L:</b> Data Transmission Sequence .....	107
<b>Index</b> .....	109

## Introduction

"But do I REALLY have to take a course on 'How to Operate A Printer' to use this one?" No — not if you only want to use it for mundane printing. It prints "mundanes" very nicely.

If you want to learn to use the exotic features it offers, approach your new Printer in the same way as your first computer — with a healthy curiosity and an open mind. It probably far exceeds what you ever expected from a printer inexpensive enough to call your own.

The MX-80 Printer works with virtually any computer properly interfaced to it. This "Learner's Manual" uses the popular Model I TRS-80 as its teaching model since the TRS-80 graphics set is similar to the one built into the MX-80. What's taught here applies to all computers, tho not all are powerful enough to utilize every MX-80 feature. More on that in good time.

Impatient readers should head for the Appendices and usual reference materials at the back. The rest of us will believe that this is no ordinary Printer and take the time to learn to use it right — *the first time*.

A good working knowledge of the BASIC\* language is all that's required as we begin, with Chapter 1.

\*Users who feel their BASIC skills are a little rusty are referred to the following books by the same author:

*The BASIC Handbook, An Encyclopedia of the BASIC Language.* Available thru CompuSoft Publishing, Box 19669 San Diego, CA 92119.

*TRS-80 Level I User's Manual.* Available thru all Radio Shack stores as Catalog Number 62-2016 or 26-2101.

*LEARNING LEVEL II.* CompuSoft Publishing.

## Chapter 1

### The Starting Line

This important Chapter shows how to unpack the Printer, set it up and test it. Read it **before** you get into trouble. The time spent will be your best investment since buying the Printer itself.

### Counting the Parts

Open the box carefully and remove the contents. There should be:

1. This User's Manual (obviously . . .)
2. The EPSON MX-80 Printer
3. A long box containing the ribbon cartridge
4. A wire rack to guide the paper

You also need an EPSON cable to connect the Printer to your particular computer. Computers which do not use the Centronics Standard Parallel Interface scheme need the proper EPSON matching kit. Your EPSON dealer can supply whatever is needed.

### Setting It Up

Lift the printer lid and locate the 2 shipping screws. See Figure 1.

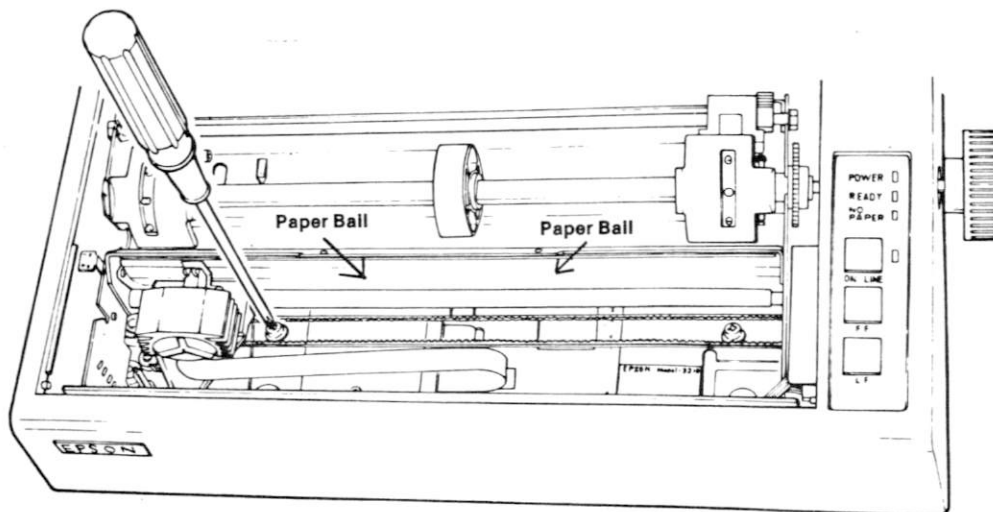


Figure 1-1

These screws are to protect the MX-80 from damage during shipping. Let's remove the printer lid so we can get at the screws without breaking something.



lay the Printer flat on a firm surface and raise its lid to the full vertical position. See Figure 1-2.

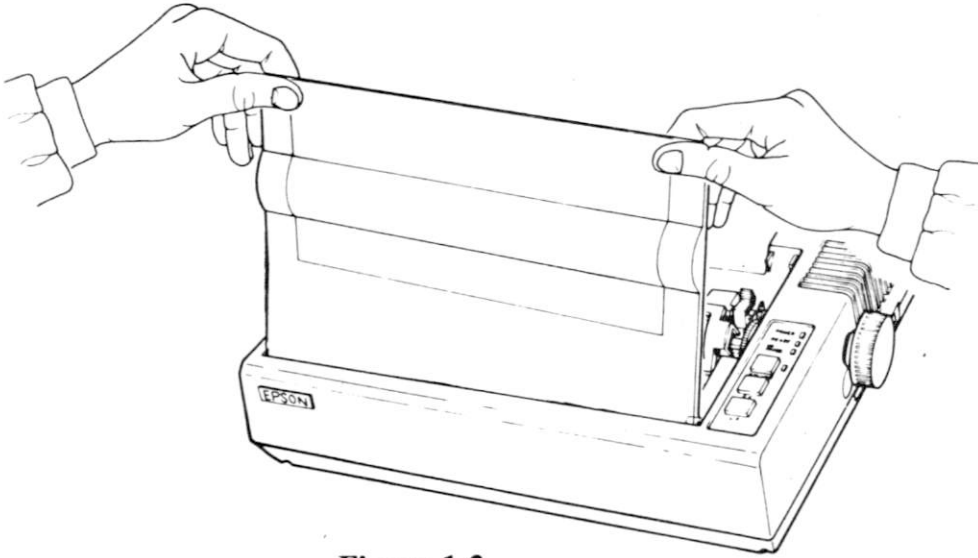


Figure 1-2

With the Printer facing you, as shown in Figure 1-3, grasp the lid firmly on its top side with your **right** hand. Use your **left** hand to hold the Printer down on the table and “spread” the case open just a bit. Pull the lid off as shown in Figure 1-3.

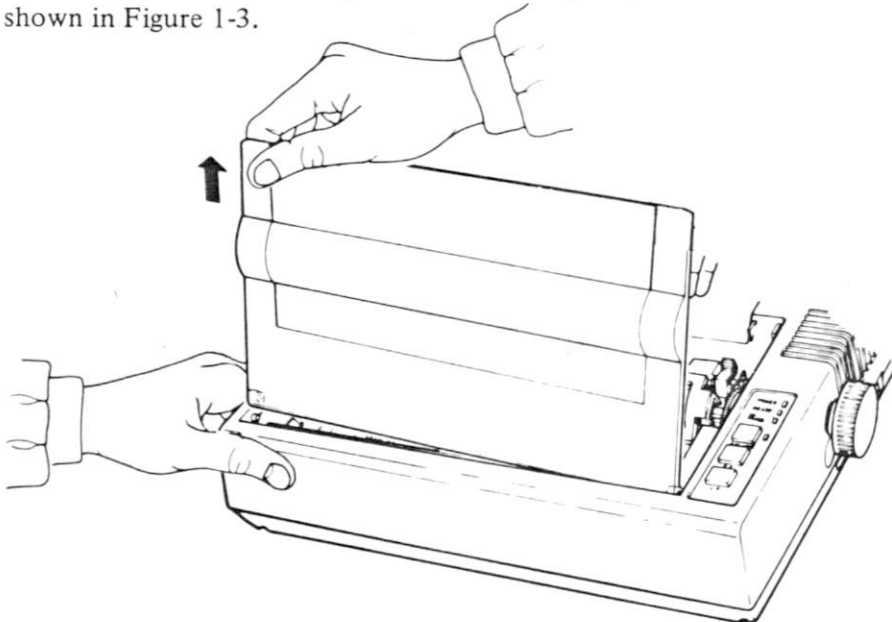


Figure 1-3

While it isn't absolutely necessary to remove the lid to get at the screws, it does reduce the chance of accidental damage. It also gives us an opportunity to learn how to remove the lid so we can learn about other interesting things.



Using a Phillips type screwdriver, remove the packing screws and store them for possible future use. If it's necessary to move the print head to the right or left to get at the screws, do so with great care. (If you've not followed the directions and have already tried to run the Printer, be careful of the print head! It runs very HOT under normal operation.)

### Installing the Ribbon

Position the Printer with the EPSON label (front) facing you. Locate the paper bail (the moveable metal bar with numbers 1-8 $\phi$  on it) and push it towards the back, away from you.

Remove the ribbon cartridge from its box and turn the plastic knob counterclockwise so the ribbon is tight. (Figure 1-4A)

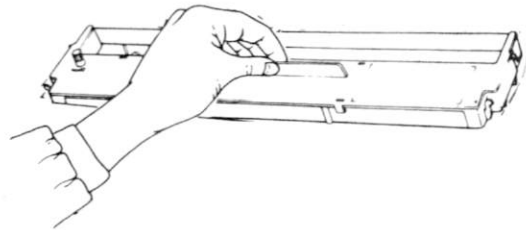


Figure 1-4A

Holding the cartridge by its vertical fin (sounds like a shark), steer the 4 tabs on its sides into the 4 slots in the Printer's metal frame. (Figure 1-4B) Press the cartridge firmly into place.

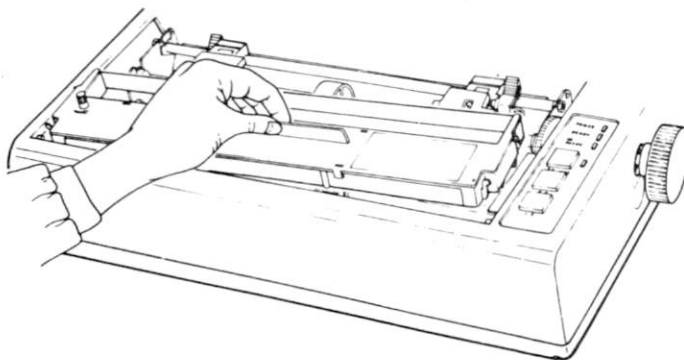


Figure 1-4B

Using a pencil (or your fingers) lift the ribbon into the slot in front of the print head as shown in Figure 4-1C. With a little practice, you can do all the above in one quick operation.

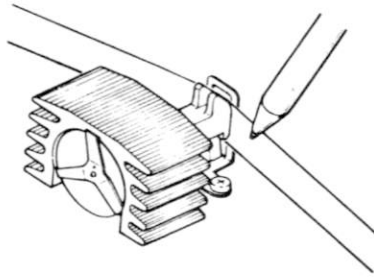


Figure 1-4C

Wind the ribbon tight again with the little plastic knob, and it's all set to go.

## Matching the Printer to Your Computer

There are 12 different switches inside the Printer case. They can be set to make the MX-80 match a wide variety of computers and computer interfaces. We now have to set them to match our computer. In a later Chapter we will reset some of them and do some very unique things.

## Opening the Case

To get at the switches we have to open the case. The transparent lid should still be off. Remove the roller knob by pulling straight out, with firm but steady pressure. (Figure 1-5)

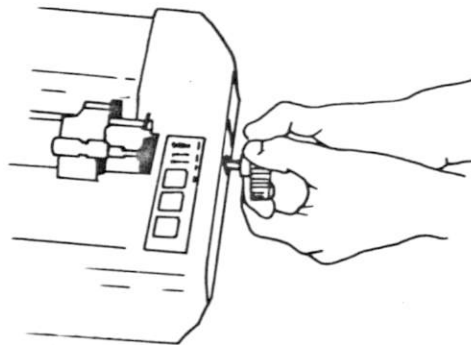


Figure 1-5

Turn the Printer upside down on a soft surface. With a Phillips head screwdriver, completely loosen (but don't remove) all 4 corner screws as shown in Figure 1-6. Place tape over the 4 holes so the screws won't fall out when we tip the Printer rightside up again.

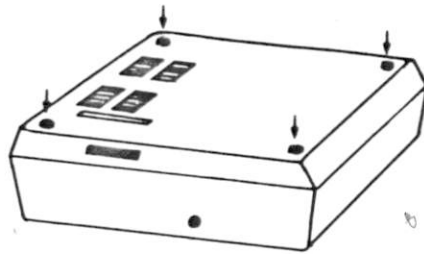


Figure 1-6

Now tip the printer rightside up again.

Gently wiggle the top cover loose. Wires are hooked to it! We are not going to completely remove the cover -- only open it enough to gain access to the switches.

Lift the cover up, mostly from the left side, and slide it **gently** to the right to free it from the roller shaft. **Be careful** not to pull the wires on the right hand side. With just a bit of class we can maneuver the cover so it stands firmly in place, as a sentinel guarding the goodies.

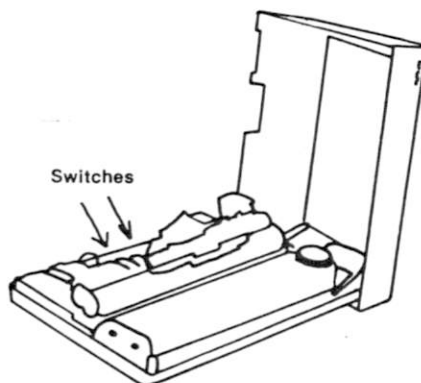


Figure 1-7

Take a minute to gawk at all the stuff in the box. Wow! As Custer said, "Look at all those . . ." well, anyway.

Wonder how they sell it as cheap as they do? (Hope it prints as good as it looks.)

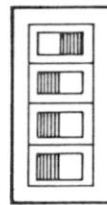
## Setting the Switches

Position the Printer as shown in Figure 1-7. Look for the 2 clear plastic caps. They are dust covers, and snap right off. **Gently**, remove them.

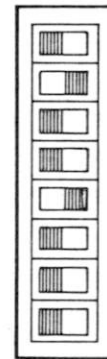
The assembly on the right contains 8 switches, and is labeled "SW1". The smaller one to its left contains 4 switches and is called "SW2". Switches set to the **left** are ON -- to the **right** are off. So far, so good.

They will probably leave the factory burn-in and checkout station set something like this:

SW2	SW1
	1-8 L
	1-7 R
	1-6 L
	1-5 L
2-4 R	1-4 R
2-3 L	1-3 L
2-2 L	1-2 L
2-1 L	1-1 L



SW-2



SW-1

If they are not like this, and you are connecting the Printer to a TRS-80 with an EPSON cable, set the switches as above. Other computers using the Centronics standard PARALLEL interface matching scheme will want the switches set the same way.

If you are using a Model I printer cable supplied by Radio Shack, go directly to Appendix E for more information. (Do not pass GO. Do not collect \$200.) If you are using a TRS-80 Model II, go directly to Appendix F.

If you are using an Apple, Pet or S-100 buss computer, go directly to Appendix D. Instructions for correct switch settings for these and other popular computers are included with the EPSON cables and interface kits designed especially to connect them to the MX-80. A complete description of what each switch does is found in Appendix C (for those who simply can't wait to find out). The rest of us will continue on.

A plastic plate fell out of the access hatch on the inside back of the lid when we opened the case. We can either replace this plate as we replace the lid, tape it to the OUTside of the hole, or add it to our growing collection of printer parts. The switches can be REset through this hatch without having to reopen the case (by using a flashlight and small screwdriver or ball-point pen).

Got all the settings correct for your Computer? Put the case back together and tighten the screws. The Printer cover goes back on the way it came off — very carefully. Same with the transparent lid.

## Printer Cable Connection

We're getting closer to the good stuff.

Be sure both the computer and Printer are turned OFF. Connect the cable to the Printer **only**. EPSON supplies a variety of cables to match different computers. Be sure you have the **right** one. Do **not** connect the other end of the cable to the computer yet.

Take your time! Double-check to ensure that the cable is in place **and locked**. It may take a firm push on the connector to get the clips secured. Without this tight connection, printing may be erratic. Again — leave the other end disconnected.

## The Paper Rack

The wire rack with the plastic tube is a paper separator. It allows the paper to feed smoothly into and out of the printer. To install it, simply spring the wires a bit so they pop into the holes in the printer frame, as shown in Figure 1-8.

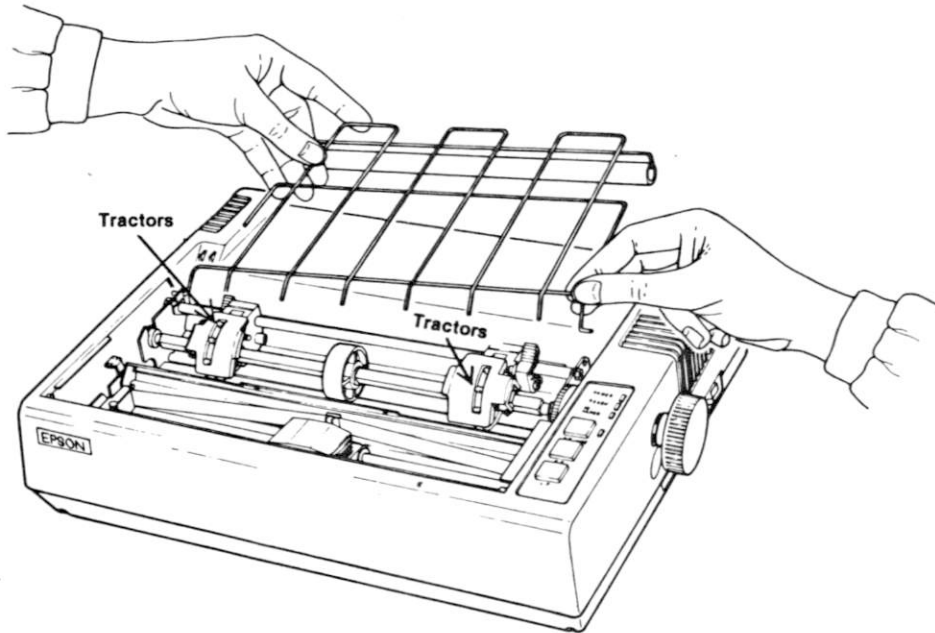


Figure 1-8

## Feeding the Paper

The MX-80 takes pin-feed paper between 4" and 10" wide. Both the left and right hand TRACTORS are adjustable to match the spacing between the holes. (We never get too far from the land, do we?)

Position the paper on the floor behind and below the Printer. Reliable operation depends somewhat on the weight of the paper keeping itself taut. Raise the printer lid and move the bail towards you. Open both tractor covers and center the plastic roller between them. (Figure 1-9)

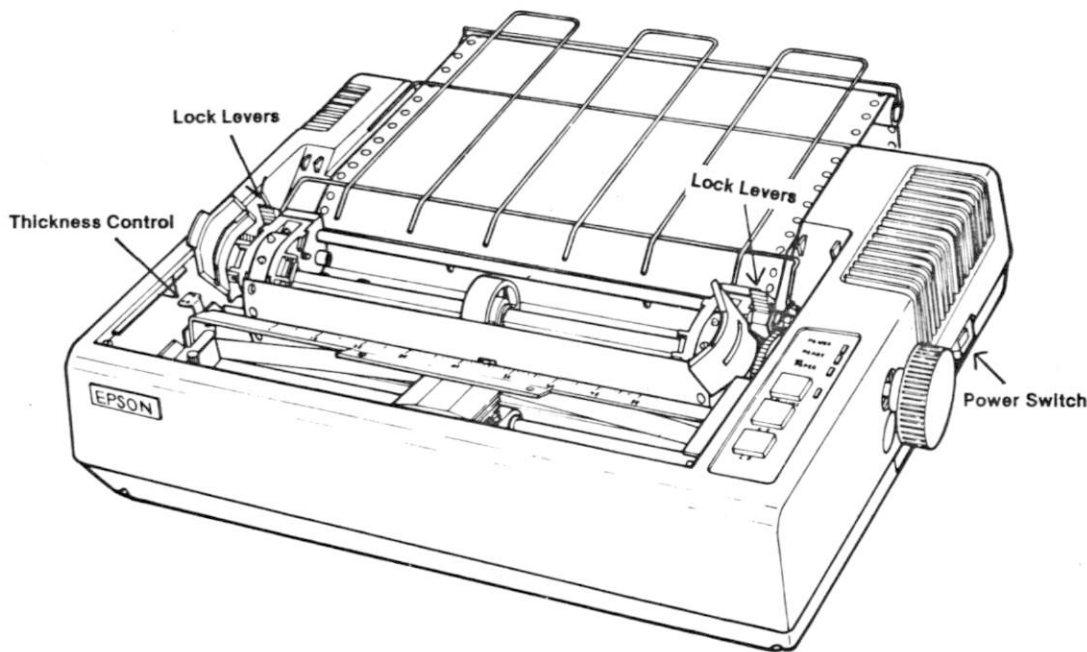


Figure 1-9

Feed the paper above the plastic tube but below the wire frame, into the slot and right on around to the tractors. Move the tractors as necessary to match the paper hole spacing. The tractor position lock levers are shown in Figure 1-9.

Position the paper holes on top of the tractor teeth and close the tractor covers. Adjust one or both tractors so the paper is centered as you wish it, and is held firmly in place. Push the bail back up against the paper.

Roll the paper forward with the roller knob.

**CAUTION:**

Never attempt to turn the roller knob when the power is ON. It's done electrically then, and gears are engaged (. . . you might damage them).

The Printer moves paper **forward** only, and never looks back. If you must turn the roller back manually (power OFF), it helps to pull lightly on the paper.

Pull the bail back out of the way and set the paper so that the top of the guide on the print head is positioned right at the top of a new sheet of paper (Figure 1-10). Push the bail back up against the paper.

## Adjusting for Paper Thickness

The MX-80 can print on all types of pre-printed multiple copy forms and carbons, as well as ordinary printer paper. That's what the 7 position thickness control lever is for (Figures 1-9 and 1-10). It moves the print head closer to or farther from the paper. For ordinary single thickness paper, start out with the control lever at about the middle position.

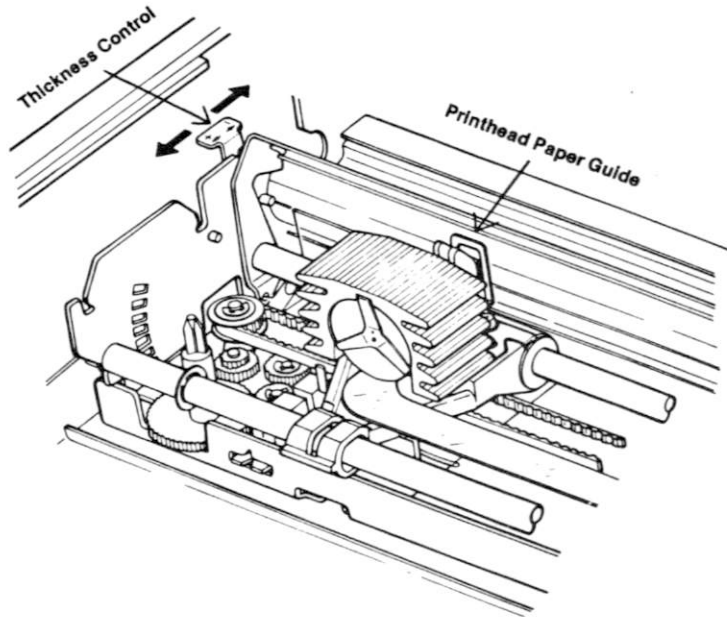


Figure 1-10

## Plugging It In

The big moment is getting closer.

Printers sold in the U.S. and Canada are designed for a standard 120V 60Hz outlet, and have a 3 wire grounding plug. **Do not attempt to defeat the grounding.** When you've located a proper outlet, see that the POWER switch on the right-hand side (Fig. 1-9) is OFF, then plug the cord in!



## At Last

The big moment is here.

### T U R N I T O N !

Mmmm! Look at that! A little startup sound, 3 green lights, and it just sits there waiting. The old days were never like this.

Where is all the motor noise? How about some big levers, lights and alarms? Looks like dynamite in a small silent package!

If we did everything right, the POWER, READY and ON LINE lights should be lit. If we got the paper in wrong, the NO PAPER light will be lit too.

Press the ON LINE button several times. It “toggles” the printer-to-computer connection, alternately hooking us up and disconnecting us from the computer (which we haven’t connected to, yet). Finish fiddling with the Printer ON LINE and READY.

Now try pushing the FF (Form Feed) and LF (Line Feed) buttons.

Nothing! What’s wrong? Nothing. We have to “toggle” the Printer “OFF LINE” to use them. Oh.

Press the ON LINE button so the READY light goes out. Now press FF.

Good grief — look at that paper go. Now look at where the paper stopped. Right at the top of the next page. Unless (until) we change it otherwise, the Form Feed button will always advance the paper to the top of the next Form (or piece of paper).

Now tap the Line Feed button.

It advanced the paper 1/6 of an inch. Unless (until) we tell the printer otherwise, it prints 6 lines to an inch, and 66 lines to a page. Now hold the Line Feed button down for a bit.

How about that? Both manual and automatic transmission!

Finally, press the FF button again.

Where did the paper stop this time? Yep. At the top of the next form.

And that just about covers all the “obvious” buttons and switches on the MX-80. Learning how to use it is not very formidable at all, if we take our time.

## The Final Checkout

The final part of this checkout takes about 10 minutes. It’s an important thing to do, and shows us many of the print characters that are available.

Turn the Printer OFF. Hold the LF button down and turn the Printer back on at the same time.

GO!

Wow! Look at that sonofagun go. Open the lid and watch the head printing in both directions. Move the paper bail out of the way to see better.

We’re looking at all of the characters in its memory (except certain replacements we can switch in using some of the 12 switches inside). Included is the full TRS-80 graphic character set, (not just the light blocks which are usually SET and RESET). Yep, we can print graphics on the paper.

Notice that when there is a graphics character in a print line, it only prints from left to right, returning back to the left when the last graphic character is printed. It then prints the rest of the graphic character, along with the usual letters, numbers and punctuation. Nothing earth-shaking, perhaps, but it demonstrates some of the Printer’s “internal logic” compared to the old “washing machines” of not so long ago.

When this test is finished, we will automatically be switched ON LINE to the computer cable.



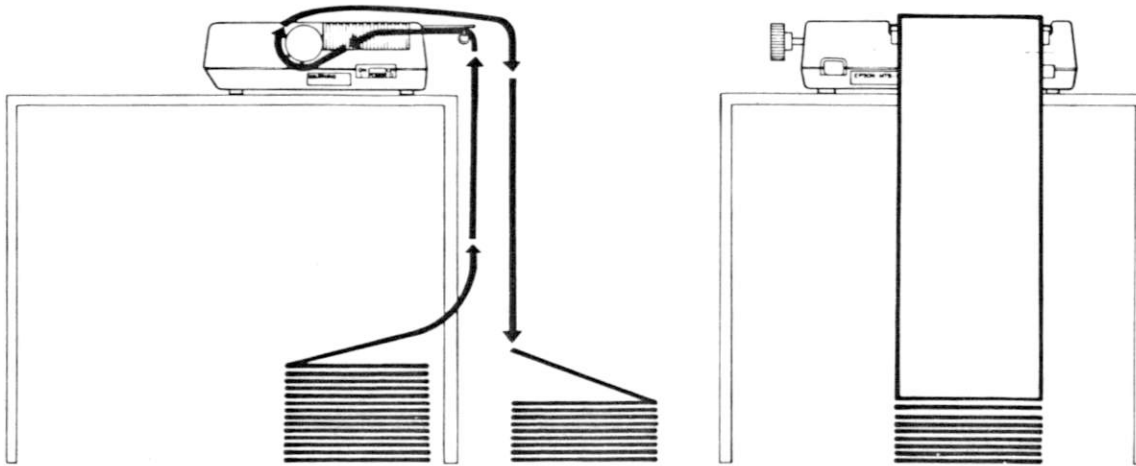
## Memory Test

How's your memory? Do you remember how to advance the paper to the top of the next form? Do it.

Had to go OFF LINE, right? Tear off the test run using the edge of the lid as a guide, and hang it up on the wall as a souvenir.

*WHEW!*

That's enough for this Chapter. Take a short walk to vent the exhilaration. In the next Chapter we'll hook it to the computer, turn IT on, and see what happens.



Correct Paper Feeding

1

2

3

## Chapter 2

### Send It A Message

The MX-80 Printer is smart. It knows how to follow instructions.

#### Any Code Devised By Man Can Be Broken By Man

Many instructions are sent to the Printer. Every letter, number, and other character travels from the computer through the printer cable in the form of a code made up of numbers. We know it as the ASCII code (American Standard Code for Information Interchange), pronounced ASK-Key.

Let's take a quick glance at Appendix A to refresh our memory. The decimal number 65 stands for the letter A -- etc.

The ASCII code numbers for **upper** case letters, numbers and punctuation are pretty well standardized around the world. Unfortunately, the remaining code numbers are not very well standardized, even among computer manufacturers within a single country.

We can also send "special" codes to make the Printer print **narrow** letters, **wide** letters, actual **graphic** characters from the screen, plus do many other things. To take advantage of all these features, however, the **computer** has to be able to send these codes.

As we will see, each with his own computer, not all computers can send all codes. With printer technology advancing faster than computer technology, the computer is replacing the Printer as the weak link in the system.

#### The Code Courier

In many cases, the easiest way to send these special codes for special things is to "build" them into the computer program along with its "regular" codes for letters and numbers. We can do this quite easily using programs written in the BASIC language.

We can also send these special codes at the BASIC "command level" **before** running a program. A program can also contain codes to change earlier codes, allowing us to print things **the way we want them, when we want them.**

The route to success in all cases is via CHR\$.



## The Fort Tandy Connection?

**CAUTION: MAKE ALL CONNECTIONS WITH EVERYTHING TURNED OFF!**

If you are using a TRS-80, continue straight ahead. If you have an Apple, a Pet, or any other computer, head straight for the special instructions that came with your EPSON interfacing cable. When you have followed those instructions, return here, ignoring future sections that are clearly written for the TRS-80, like the next paragraph.

The other end of the Printer Cable hooks to the Printer Port on the Model I Expansion Interface. Be sure the cable is positioned so the wires lead downward from the plug, as shown in Figure 2-1.

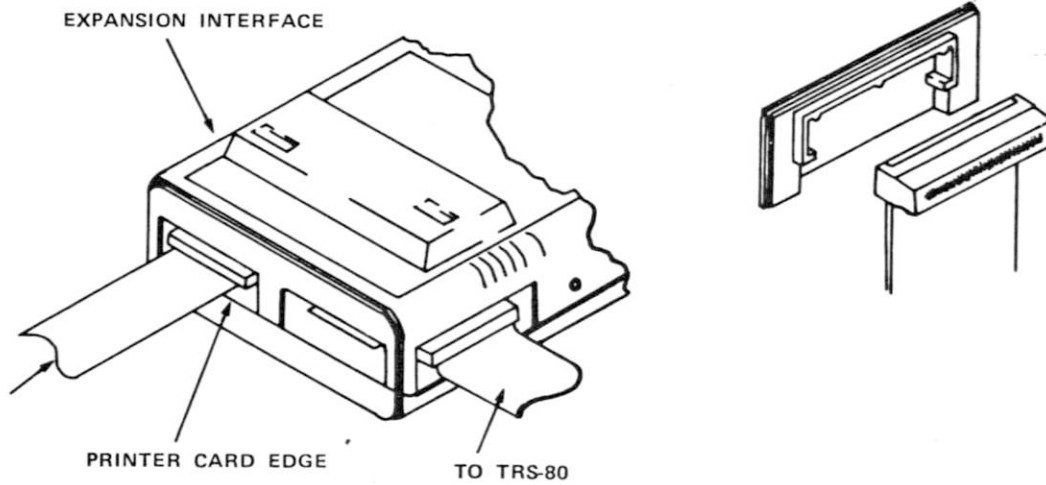


Figure 2-1

If you are using some other “special” printer interface, follow any special directions that come with it.

## All Systems Are “Go”

Fire up the computer, loading in BASIC as usual. Turn the Printer ON. The 3 lights, POWER, READY and ONLINE should be lit.

Type in this simple program, but do NOT RUN it yet:

```
10 LPRINTTAB(13);"GREETINGS FROM THE VERSATILE"  
20 LPRINTTAB(22);CHR$(14);"MX-80"  
30 LPRINTTAB(40)CHR$(15)"BY"CHR$(14)CHR$(27)CHR$(71)" EPSON"  
40 LPRINTTAB(36)CHR$(198) + STRING$(18,211) + CHR$(185)  
50 LPRINT : LPRINTTAB(20)CHR$(18)"YOU'LL LIKE ME!"  
60 LPRINTCHR$(27)CHR$(72)
```

### Time Out For Emergency Training

CAUTION: If you make a typing error which causes the program to crash, or look weird, be sure to turn the Printer OFF then ON again before running the corrected program. The reason is, the faulty program may have sent an unwanted code "down the line". It may even have sent something unpatriotic like "don't listen to the computer". The Printer will reset itself to "normal" by our simply turning it OFF and ON.

It may also be necessary to shut the computer down COLD, and start over from the beginning. The Printer has its own internal "computer", and the two computers talk to each other. If one should decide to throw a temporary snit, we have to get in between them and cool things down. Doesn't happen often, but as with any computer, a glitch on the power line or a static electricity discharge can cause all sorts of heartburn. Best to know how to handle it when it comes.

### Operator Now Certified

Now we're prepared to make that program:

RUN



There it is! We finally strapped these two pieces of high technology into harness and got us a real convoy.

```
GREETINGS FROM THE VERSATILE
      MX-80
      BY EPSON
      ██████████
YOU'LL LIKE ME!
```

Figure 2-2

RUN it again. This time open the lid and pull the paper bail forward to get a better look at the print head in action.

Did you see it printing in both directions (except for the graphics characters)? Did you see the head print the last line **twice**, to make the print darker? If not, RUN it again and watch more carefully.

There is a lot going on here, so we'd better dissect the program to learn all about it. Type LLIST to make a "hard copy". Use the line feed (LF button, remember?) to roll the print-out past the lid. Tear it off and keep handy so we can study the program as we go along.

SAVE the program on disk or tape to avoid having to type it in later.

### Line By Line

Type NEW to erase the program. We're going to load it back in a line at a time, analyzing it as we go. (An alternate strategy is to load the entire program back in from disk or tape, then DELETE all the lines except 10.)

Line 10 is straightforward. We LPRINTed starting 13 spaces from the left. Type it in and RUN it, watching the head action as it prints.

```
GREETINGS FROM THE VERSATILE
```

Figure 2-3

Well, that was rather "pedestrian".

Line 20 has 3 LPRINT statements on the same line, separated by semicolons, which are usually optional. The first LPRINT TABs over 22 spaces, the second sends one of those special CHR\$ (character string) codes we mentioned earlier, and the third prints "MX-80".



CHR\$(14) stands for:     print DOUBLE WIDTH.

Everybody back to look at Appendix B. It contains all the special codes, often called CONTROL CODES. The ASCII code number we need to use to print (or do) what we want is shown on the same line with its explanation.

It's important to note that not all codes actually print on either the screen or the Printer. Most special or "control" codes don't really PRINT anything, even tho we must precede them with an LPRINT to "push them down the line" to the Printer. It's just the means to the end.

For example, type the following at the command level:

```
LPRINT CHR$(14); "HELLO THERE"
```

and see the words appear in double width (not double spaced).

HELLO    THERE

**Figure 2-4**

LLIST the program again. Notice that the double width feature is no longer switched on. The message —

EACH TIME WE WISH TO PRINT SOMETHING IN THE DOUBLE  
WIDE MODE, WE MUST PRECEDE IT WITH CHARACTER CODE 14.

When that line is finished printing, the double width feature is **automatically turned off**. It's the only "control code" we have with a special automatic shut-off feature.

Now that we understand line 20, let's add it in and RUN our 2 line program.

```
20 LPRINTTAB(22);CHR$(14); "MX-80"
```

GREETINGS FROM THE VERSATILE  
MX-80

**Figure 2-5**

Pretty nice — eh? If you wish, do another LLIST to feel confident that DOUBLE WIDTH really got switched off.

The programming in line 30 is a bit more exotic. We can quickly see that it consists of only one LPRINT statement to “push” one TAB, and four control characters to the Printer. The control codes are of course in Appendix B, but are also listed here “in order of appearance”.

- 15 stands for COMPRESSED CHARACTER mode
- 14 stands for DOUBLE WIDTH mode
- 27 stands for “ESCAPE”
- 71 is ASCII for the letter “G”, code for DOUBLE STRIKE

*which forces us to gosub to another topic.*

### The Great Escape

The MX-80 recognizes 13 so called “Escape Codes”. They are similar to (and really part of) the control codes under discussion. The Escape codes should not be confused with the escape key which some computers have, tho they are shirt-tail relatives.

We send the special code CHR\$(27), which means “ESCAPE”, down the line immediately before another code which requires that ESCAPE. These special code clusters are logically called “escape codes”. We can either build them into a program or send them from the command level, just as we did before with the simple control codes. We will learn how to use each “escape code” as it is needed.

### Our First Escape Code

Escape “G” stands for DOUBLE STRIKE. (Not double width.) Everything on a program line which contains an Escape G is printed TWICE by the print head. Before the second time it is printed, however, the paper is rolled up just 1/216th of an inch to help fill in the spaces between the dots printed on the first pass. The result looks very bold and solid, almost like it came from a regular typewriter. Pretty Clever.

Further, once the “G” switch is thrown, everything which follows is DOUBLE PRINTED until an ESCAPE “H” message is sent, returning printing to normal. (The ASCII code number for “H” is 72.)

*Return*

*RETURNING from our gosub, add line 30 and RUN.*

```
30 LPRINTTAB(40)CHR$(15)"BY"CHR$(14)CHR$(27)CHR$(71)" EPSON"

  GREETINGS FROM THE VERSATILE
    MX-80
  BY EPSON
```

**Figure 2-6**

Pretty impressive. Just like they do it in the big city! The letters BY in line 30 are COMPRESSED to 132 per line. CHR\$(15) caused it. The usual spacing is 80 Characters Per Line (80 CPL), or 10 characters to the inch.

In the DOUBLE WIDTH mode, we can print either 40 or 66 characters per line, depending on whether we are in COMPRESSED character or NORMAL mode. Since we are in COMPRESSED, the word EPSON was printed DOUBLE WIDTH/COMPRESSED, 66CPL. CHR\$(14) is the reason.

STOP now and think this idea through! It's really very easy, even if some of the words are similar. Like having a normal and an overdrive, it doubles our options.

### Comparing Our Options:

NORMAL WIDTH	=	80 CPL
NORMAL WIDTH/COMPRESSED	=	132 CPL
DOUBLE WIDTH	=	40 CPL
DOUBLE WIDTH/COMPRESSED	=	66 CPL

DOUBLE STRIKE – Unrelated to the above.

Pretty simple when laid out this way, right?

In addition, the print head did its thing twice, caused by CHR\$(27)CHR\$(71) = ESCAPE "G". In fact, it was so impressive, lets RUN it again. (C'mon, do it! Stay with the program.)

```
  GREETINGS FROM THE VERSATILE
    MX-80
  BY EPSON
```

**Figure 2-7**

## Trouble at the Old Corral?

Oh-oh. Now what did we do? You might know, just when things were going so well. Looks pretty cute tho, doesn't it?

Remember, whenever any "switch" is thrown by sending a Code, (except DOUBLE WIDTH), it stays thrown until reversed back to normal. In this case, Code 15 was sent, and all characters became COMPRESSED (132 CPL). It takes a Code 18 to switch it back to NORMAL (80 CPL). (Everybody gallop back to Appendix B again. We'll know it well before we're done.)

The compression principle also applies to the TABs. If we are printing 132 characters per line, then the width of each TAB must also be compressed compared to the normal 80 CPL. That's why the first 2 lines are shifted to the left. Everything is working quite normally. (What WOULD you do if you wanted to align the first 2 print lines with the 3rd?)

(Sure — change the TAB values in lines 10 and 20. Not now, tho. We're busy.)

## Never Look Back — It Might Be Gaining

Line 40 might be the toughest, but it's not bad if we hang in there. It obviously consists of one LPRINT statement, one TAB, two character strings, and one STRING-string. The STRING\$(18,211), in case you forgot, prints the ASCII character 211 18 times in a row. Everybody back to the ASCII chart in Appendix G to see what 211 stands for.

Users with computers which do not have the STRING\$ feature may substitute a FOR-NEXT loop system to do the same thing.

Line 40 can be placed with:

```
40 LPRINTTAB(36) CHR$(198) ;  
42 FOR N = 1 TO 18  
44 LPRINTCHR$(211) ;  
46 NEXT N  
48 LPRINTCHR$(185)
```

Hmm. Very interesting. The graphics set is the same as the TRS-80 uses, but the ASCII numbers are all increased by +32. Hmmm — some more. (We'll have plenty of time to explore that whole situation in Chapter 4.)

Guess we can pick any of these and the MX-80 will print them. Let's see what #211 looks like.

```
>LPRINT CHR$(211)
```

```
=
```

Figure 2-8

Well, that's sure a strange one. It must be what we printed 18 times to get the fancy parallel bars in our print-out. Wonder how we got the fancy "caps" on the ends of those bars. Probably used graphics characters. Looking at Appendix G, suppose #198 would fit on the left end, and #185 would fit on the right? Let's look at program line 40 again.

Who would have guessed it! CHR\$(198) gets printed, then character 211 is printed 18 times, then CHR\$(185). And the whole works is printed **twice** because we're still in the Escape "G" DOUBLE STRIKE mode. All of a sudden this is starting to make sense!

### Michaelangelo Should See This

Let's add line 40 (or its substitute, lines 40-48), RUN it, and watch the print head to see if it really happens that way.

```
40 LPRINTTAB(36)CHR$(198) + STRING$(18,211) + CHR$(185)
```

```
GREETINGS FROM THE VERSATILE
MX-80
BY EPSON
```



Figure 2-9

It did do it alright, but why did it print the top part of the graphics decoration 2 times, then come back and print the bottom part 2 times? Well,

**FIRST:** the graphics character is bigger than the printhead. So, pretending to be a narrow paint-brush on a tall barn, it made several passes.

SECOND: it was smart enough to figure that out by itself.



Figure 2-10

## 2 Easy Pieces

Only 2 more easy lines. It's downhill now.

Line 50 is a multiple statement line. The first LPRINT just inserts a vertical space of one line.

```
50 LPRINT : LPRINTTAB(20)CHR$(18)"YOU'LL LIKE ME!"
```

Figure 2-11

The second LPRINT tabs over 20 spaces and shoots the ASCII code #18 down the line before printing "YOU'LL LIKE ME!"

18 is code for "turn off 15". No more COMPRESSED CHARACTERS.

It returns printing from 132 to 80 characters/line.

Add line 50 and RUN, watching the printhead.

Yep! Its print size is more like what we're used to. But why did it print twice? Oh yes, it's that old ESCAPE G again. We'll take care of that in our final program line.

But first, let's double-check to see that we are really out of the COMPRESSED CHARACTER mode. What would make a good test to find out? Think about it. Think back a little ways.

## Testing, Testing

What happened to the size of all our characters after we went COMPRESSED? Ummm-hmmm. It goofed up the TAB. If size is now "normal", the TABs in lines 10 and 20 should also act "normal". If we RUN the program again... let's try it.

```

GREETINGS FROM THE VERSATILE
  MX-80
  BY EPSON
  _____
YOU'LL LIKE ME!

```

Figure 2-12

There it is! CHR\$(15) means COMPRESSED ON, and CHR\$(18) is its opposite.

One more step. Line 60 resets the "G" switch. As we've seen repeatedly over many RUNs, the printhead is printing everything twice. We need to turn off the DOUBLE STRIKE to restore the Printer to the same condition as when we began this folly. It's a common-sense principle to follow in all our programming with this sophisticated Printer. Add line 60:

```
60 LPRINTCHR$(27)CHR$(72)
```

Figure 2-13

... and RUN, watching the printhead.

Well, why did it print the last line twice? Wasn't it supposed to shut off the double-strike? But it DID shut it off, after the entire program had been printed. Think of some way to prove that claim.

An easy way is to just LIST the program. LLIST it.

```

10 LPRINTTAB(13); "GREETINGS FROM THE VERSATILE"
20 LPRINTTAB(22); CHR$(14); "MX-80"
30 LPRINTTAB(40)CHR$(15) "BY"CHR$(14)CHR$(27)CHR$(71) " EPSON"
40 LPRINTTAB(36)CHR$(198) + STRING$(18,211) + CHR$(185)
50 LPRINT : LPRINTTAB(20)CHR$(18) "YOU'LL LIKE ME!"
60 LPRINTCHR$(27)CHR$(72)

```

Figure 2-14

Just like the book said!

This has been a feature-packed Chapter, but it gave us a good introduction to some powerful capabilities we can put to work. Better take a good break before tackling Chapter 3. It wouldn't hurt a thing to come back and go through this Chapter once more before moving on, now that you know the plot.

<b>Summary of LPRINT Command Codes Learned So Far</b>
---

- |   |
|---|
| CHR\$(14) – Turns ON double width characters.<br>(Goes OFF by itself after end of line) |
| CHR\$(15) – Turns ON compressed character mode.<br>(Stays ON until turned OFF)          |
| CHR\$(18) – Turns OFF compressed character mode.  |

<b>Special 'Escape' Codes</b>
-------------------------------

- |  |
|--|
| CHR\$(27) – The ASCII symbol for "Escape".<br>(Special code used with letter codes)                |
| CHR\$(71) – ASCII number for the letter "G". (Turns ON double strike<br>when preceded by escape.)  |
| CHR\$(72) – ASCII number for the letter "H". (Turns OFF double strike<br>when preceded by escape.) |

Example: LPRINT CHR\$(27)CHR\$(71)      Turns double strike mode ON.



## Chapter 3

### More Print Control Commands

That last Chapter was a heavy one, but it gave us a good overview of many MX-80 features. In this Chapter we'll explore a few of the features in more detail, plus learn to use additional features related to print size and quality.

As always, CHR\$ is the magic wand. Continue to refer to Appendix B as needed to help maintain perspective on what we're doing and where we're going.

#### A Cheap Buzz

Type:

```
>LPRINT CHR$(7)
```

Sorry if that scared the dog! Just couldn't resist it. It really has nothing to do with printing, but is widely incorporated in programs as an alarm to indicate something. The built-in alarm is a pretty standard feature on quality printers.

Sure . . . do it again. We'll wait.

If your computer has a Real Time Clock, we can use it as a "time is up" alarm. Just include LPRINT CHR\$(7) in the program. It also makes a great audio prompter, telling the operator it's time to do something. The applications are virtually endless. Most expensive alarm clock in the place.

Technically, what we heard is called The BELL, and ASCII 7 makes it ring. Buzzers have pretty well replaced bells, so we'll usually refer to it as a buzzer, alarm or something more contemporary.



#### Back to Printing

We learned in the last chapter that COMPRESSED Characters and Normal Width characters cannot be printed on the same line. There is, however, an easy way to print 2 sizes of characters on the same line. We simply switch back and forth between single and double WIDTH. The result is quite impressive.

Type in this simple program:

```
10 LPRINT TAB(10);
20 LPRINT CHR$(14) "DOUBLE "CHR$(20) "WIDTH CHARACTERS";
30 LPRINT " ADD"CHR$(14) " EMPHASIS!"
```

and RUN.

```
DOUBLE WIDTH CHARACTERS ADD EMPHASIS!
```

**Figure 3-1**

Note carefully in lines 20 and 30 where the spaces are positioned. Now study the spaces in the print-out. When a space is placed after an ASCII 14 (double width), that space prints as 2 spaces. If placed after ASCII 20 (single width), it takes up only a single space.

Don't forget, double WIDTH mode is automatically turned off at the end of each line, even if that line ends with a semicolon (;). (The program in the last chapter demonstrated that rule.) If we want to print a long line consisting entirely of double width characters, every program line holding a part of that message must have its own CHR\$(14).

## 2 Ways Down, 2 to Go

Let's switch from the traditional 80/40 combination we used above (80 Characters Per Line – 40 CPL in double width) to Compressed mode. Type this directive at the command level:

```
>LPRINT CHR$(15)
```

then RUN.

```
DOUBLE WIDTH CHARACTERS ADD EMPHASIS!
```

**Figure 3-2**

Very nice! CHR\$(15) shifted the Printer into COMPRESSED mode – 132 CPL. Our program lines 20 and 30 tell it to print the words DOUBLE and EMPHASIS in DOUBLE WIDTH. Half of 132 is 66, so those words are printed at 66 CPL and the rest at 132 CPL. It all really makes sense, doesn't it.

Now let's RUN the program again.

```
DOUBLE WIDTH CHARACTERS ADD EMPHASIS!
```

**Figure 3-3**

Hmmm. Same thing. Why did it stay in the compressed mode? We didn't send another ASCII 15 down the line.

Think back to the last Chapter. Double Width reverts to Normal Width at the end of each program line. (It's the only printing command that has automatic shut-off.) Compressed mode stays switched in until we send an ASCII 18, or turn the printer off. If we want to change modes, we have to say so.

## Why?

Good question. Simple answer.

Double Width is quite unique but loses its visual impact if used all the time. We get “bumped” out of it automatically at the end of each line and have to tell the Printer each time we actually need it.

Compressed mode, on the other hand, is the sort of thing we might use hour after hour. A typical application is the printing of a big 132 column sheet of financial reports on 8½” wide paper, instead of the bulky 14” paper. This scheme gives us an original small enough for making copies on an ordinary duplicating machine. To have to write the computer program so each line told the printer to **stay** in compressed mode just wouldn’t make good sense.

Let’s return the Printer back to “normal” either by typing:

```
>LPRINT CHR$(18)
```

or turning it OFF, then ON. (The latter re-initializes all software inside the Printer.)

## Double Striking Revisited

Recalling our complex example from the last chapter, we **DOUBLE PRINTED** part of the message. Double printing (or double striking) is really more sophisticated than it sounds. Instead of simply going over what was printed the first time and hitting it again, the Printer actually rolls the paper up 1/216th of an inch, then prints the text again. The effect is to fill in much of the space between the dots and make the characters look more solid. Sort of a “poor man’s” typewriter.

Type NEW, then the following program:

```
20 LPRINT "AIN'T SCIENCE GRAND?"
```

```
then RUN.
```

```
AIN'T SCIENCE GRAND!
```

**Figure 3-4**

is typed once.

Add this line:

```
10 LPRINT CHR$(27) "G"
```

and RUN.

```
AIN'T SCIENCE GRAND?
```

**Figure 3-5**

Nice and bright isn't it. Almost as good as if typed on an electric typewriter.

### Dissecting Line 10

Line 10 says LPRINT Escape G. We saw that idea used in the last Chapter as LPRINT CHR\$(27)CHR\$(71). 27 is of course the ASCII number for ESCAPE, and 71 the ASCII number for the letter G. (Everybody back to Appendices A and B.)

Most printers require escape commands to be via ASCII numbers. With the EPSON MX series printers, the command can be sent either via CHR\$(##) or with the letter itself enclosed in quotes. We used Escape "G" here since it's shorter to type, and perhaps easier to remember.

Try to memorize these codes as we go along. They are really not very complex once we understand the "big picture", and constantly referring to Appendix B is not all that exciting.

Like most other commands, Double Strike stays in effect until cancelled. We return the Printer to normal single-strike by sending it an Escape H. Let's do it:

```
>LPRINT CHR$(27) "H"
```

and type LLIST. Check the copy to see that we are really back in the single-strike mode.

### The WALLBANGER Mode

We have one final print "quality" command. It is, in nice company, called the EMPHASIZED PRINTING mode. Others have called it "the Wallbanger", "Jackhammer", "Super Whammy", etc. It is a double charged, plug it in the 440V outlet, sock it to the paper, mode. Change our existing program to read:

```
10 LPRINT CHR$(27) "E"
```

```
20 LPRINT "PUT ON YOUR HARDHAT!"
```

```
30 LPRINT CHR$(27) "F"
```

and RUN.

**PUT ON YOUR HARDHAT!**

**Figure 3-6**

Well, that wasn't as violent as expected, but look how nice the print looks. Escape E turned it ON and Escape F turned it OFF. A quick LLIST will confirm that it's off.

We'll come back to this EMPHASIZED PRINTING mode in Chapter 5 when we address the problem of printing multiple copies of invoices, letters, etc.

## The Ultra Machine

How about combining some of these commands to give us the **ultra** in high quality dot-matrix printing? Let's put the Printer in both the wallbanger and double strike modes at the same time and see what happens. Add lines 5 and 35 to incorporate double-strike, making our program read:

```
5 LPRINT CHR$(27) "G"  
10 LPRINT CHR$(27) "E"  
20 LPRINT "PUT ON YOUR HELMET AND GOGGLES"  
30 LPRINT CHR$(27) "F"  
35 LPRINT CHR$(27) "H"
```

and RUN.

**PUT ON YOUR HELMET AND GOGGLES!**

**Figure 3-7**

That's about as good as it's going to get, and that's not too shabby.

## Caveat Printer

Only one minor restriction on Emphasized Printing. It cannot be used in the Compressed mode. If we try, the Printer automatically switches to normal width but stays in Emphasized mode.

It's only logical, if we think about it. The Compressed mode has the dots spaced very close together. If we turn all the engines up to Warp 7 it might just blow the print needles right thru the paper.

Oh, yes. Emphasized printing is only half as fast as regular printing. Don't want to overheat the engines you know. If we then add double strike, we cut the speed in half again. Quality has its price!

## A Word About Word Processing

Several very good word processing programs are widely used on microcomputers, including Scripsit™ and The Electric Pencil™

They share a common weakness, however. As of today, they do NOT allow us to send OUR choice of CHR\$ commands to the Printer. We can partially overcome this deficiency by sending our commands to the Printer in advance of printing from the word processing program.

Let's say, for example, we want to print a high quality business letter by using double-strike, and make several carbons. All we have to do is send the following instructions before calling up the word processing and printing the letter:

```
>LPRINT CHR$(27) "E"
```

```
>LPRINT CHR$(27) "G"
```

Escape E turned on Emphasized Printing, and Escape G turned on Double Strike, for the carbons. If we'd wanted, we could have sent a CHR\$(15) to set up Compressed Mode, in place of Emphasized Printing.

We recall that those 3 commands turn features on that stay on until specifically turned off by their "opposite" Escape codes. Sending the computer off into a word processing program will not affect what the Printer has stored in its own memory.

Remember, if we lose track of what Printer features are ON and OFF, just switch the Printer OFF, then back ON -- and it reinitializes itself, returning all the software controllable features to their original or "default" conditions.

## Chapter 4

### Introduction to Graphics

The MX-80 prints all the graphics characters that come with the standard Model I TRS-80. To use this unique feature we need some careful explanation and clarification, so don't try to read too fast.

Any program which a Model I prints on its screen with the conventional SET, RESET statements can be printed by the MX-80. In addition, there are many other TRS-80 graphics that are less frequently used. Every one of these graphics characters also has an ASCII number so each can also be printed. Any ASCII graphic code number sent down the line will cause the corresponding graphics character to be printed.

#### Only By a TRS-80?

*NO!* ANY computer which can send the proper ASCII numbers to the MX-80 makes it print those graphics characters on paper. Any such computer will act like it's a TRS-80, as far as graphics are concerned.

**VERY IMPORTANT POINT:** The graphic characters **printed** are the ones built into any Printer, not those built into any computer! Read that again . . . slowly.

The MX-80 graphics just happen to exactly match those which are inside the Model I. Since the **printer** graphics are the same as the **COMPUTER** graphics, they can be duplicated exactly by the Printer.

The same holds for all other characters except the 4 arrows – up, down, left and right. (Any computer capable of sending the ASCII codes can print exactly the same thing on paper, even tho it can't print them on its own screen.)

#### And in Either of Two Ways

The most obvious way to push an ASCII number down the line is with an LPRINT and CHR\$. If you're using a Model I, let's print the same graphics character on both on the screen and the Printer. (If not, follow along and you'll get the point O.K.)

Type in this program:

```
10 PRINT TAB(30) CHR$(191)
20 LPRINT TAB(30) CHR$(191)
```

and RUN.

Oops! What went wrong here? We sent the same ASCII number to both the computer and the Printer, but the results are different. The screen shows



Figure 4-1

while the Printer says



Figure 4-2

Some explanation is in order here, and it better be good!

*WELL, AHHH, ER, UMMM*

It's like this.

By tradition, ASCII code numbers 0 through 31 are reserved for control codes. It is in this area that different computer makers do their own thing, sometimes agreeing on little else than #7 should be reserved for the BELL (buzzer). The Model I is also a "free spirit".

By further tradition (but less deeply steeped), ASCII numbers 128 through 159 are reserved for those **same** control codes, more or less. In the case of the Model I -- more. In fact, the Model I control codes found between ASCII 128-159 are **more** complete than those found between ASCII 0-31. Terrific! We really needed that, didn't we.

### But What Does It All Mean?

It means that the upper set (higher numbers) of control codes is **better** than the lower set, that's what it means. In order to get the maximum capability from the Model I, we have to use the upper set of control codes.

The problem is, the TRS-80 Model I uses those **same** upper ASCII numbers **inside the computer** to put graphics characters on the screen. When they are sent **outside** the computer via LPRINT, they are properly interpreted as control codes. EPSON had to move the graphics characters up out of the way and give them new ASCII numbers.

### Who's Responsible For This Mess?

Now, now. There'll be no finger pointing. However, if it wasn't screwed up to start with, EPSON wouldn't have had to take evasive action to fix it.



## OK. What's the Bottom Line?

The bottom line is -- to print the same graphics characters on paper as seen on the screen thru use of `LPRINT CHR$(###)`, we have to add 32 to the regular Model I graphics ASCII code number. The new EPSON graphics ASCII codes for printing on paper are between 160-223. See Appendix G.

## OK, What Are My Options?

There are 3 options:

1. Accept the fact that we have to add 32 and stop fighting it.
2. Move internal switch 2-4 to the ON position. This deliberately switches the graphics down 32 ASCII spaces, so the Printer graphics have the same ASCII numbers as Model I video screen.

That might sound good, but it's really a poor option. It replaces the more versatile control codes that usually reside there and eliminates many of the really fine features the MX-80 offers.

3. Print the desired graphics display on the screen, then "dump" the screen's contents to paper. Hmmm. It has its points. What are the trade-offs?

On the plus side, **anything** shown on the Model I screen can be "dumped" to paper. On the negative side, the screen can't display compressed characters, or other non-printing control characters. They have to be set up prior to the "screen dump".

## Exercising the Options

We've already discussed option 1 and learned the reasons behind it. It's a good option and will probably be used for most serious programming.

Option 2 is a loser since it robs us of the Printer's power. It does work fine however for mundane LPRINTing and LLISTing.

Option 3 is worth some study. The "screen dump" approach has real value, and we don't have to be highly skilled to use it. The necessary software program is coming up next.

## The Screen Dump

The computer's memory locations which hold the information printed on the screen have address numbers 15360 to 16320. To print what they're holding (and thus what's on the screen), all we need do is PEEK at these locations and LPRINT their contents. It's almost that simple.

Type in this important program. It's our "DUMP" program:

```
30000          REM  * SCREEN DUMP PROGRAM *
30010 FOR L = 15360 TO 16320 STEP 64 'BEGINNING OF EACH LINE
30020  FOR P = 0 TO 63   'EACH POSITION ON THAT LINE
30030   A = PEEK(L+P)   'A = ASCII # OF ADDRESSES CONTENTS
30035    IF A<32 A=A+64 '(OPTIONAL) ADDS MSB 4TH BIT
30040    IF A>127 AND A<192 A=A+32 'OFFSET THE GRAPHICS ONLY
30050    LPRINT CHR$(A); 'SEND ASCII # TO PRINTER
30060  NEXT P          'MOVE OVER TO NEXT POSITION
30070  LPRINT         'MOVE PAPER UP ONE LINE
30080 NEXT L          'START ALL OVER ON THE NEXT LINE
```

Save it on disk or tape. You may use it frequently.

Now let's give it a try. Press:

CLEAR to clean off the screen.

Then LIST the program, to put something on the screen.

Then RUN the program, watching the screen contents dump out to the Printer. Yes, it's a bit slow, but we're doing the dump in BASIC to keep it simple — and it does work.

Not too bad, eh? Note that the Printer even "prints" those screen lines that don't have anything on them. Since the dump program is in BASIC, we can stop it anytime by just hitting the BREAK key.

## Putting Dump To Work

Well, the above was a lot of work just to get a program listing. We can do that with a simple LLIST. Let's put our dump program to a **real** test.

Add these program lines. They fill the screen with random characters of all sorts, then our Dump program sends them to the Printer:

```

1 CLS
10 FOR N = 1 TO 200      'PRINT 200 CHARACTERS ON SCREEN
20  R = RND(159) + 32    'RANDOM # TO 159 BUT JUMP UP 32
30  L = RND(1024) -1     '1024 SCREEN LOCATIONS, INCLUDING 0
40  PRINT@L,CHR$(R);    'PRINT THE RANDOM CHARACTER AT L
50 NEXT N

```

then RUN.

Amazing. Quite a display, including graphics characters we've probably never used before. Since this program's purpose is to just put 200 arbitrary characters at arbitrary locations, some characters may get replaced by others. No matter. The proof is in the dump.

Check the finished print-out against the screen to be satisfied that all this is true. Once done, we can have confidence in our dump program.

Yep! It is as represented. The print-out seems a mite wide and a bit short, compared to what we see on the screen. (Picky Picky!) But it's true. The reason is that the "aspect ratios" are not the same.

### *Aspect Ratio?*

Yes. In TV engineering parlance, that's the ratio of the picture's width to its height. The perfect TV picture is one and one-half times as wide as it is tall. It's aspect ratio is 3:2.

With the Printer in the normal 80 CPL mode, our printout looks like maybe its aspect ratio is about 2:1. Wonder how we could make our screen dumps look more like the screen? Ponder that one. How would YOU solve it?

## The Next Envelope, Please

Sure, we could try switching the Printer into Compressed Character mode. That would certainly make the print-out narrower. Let's try it.

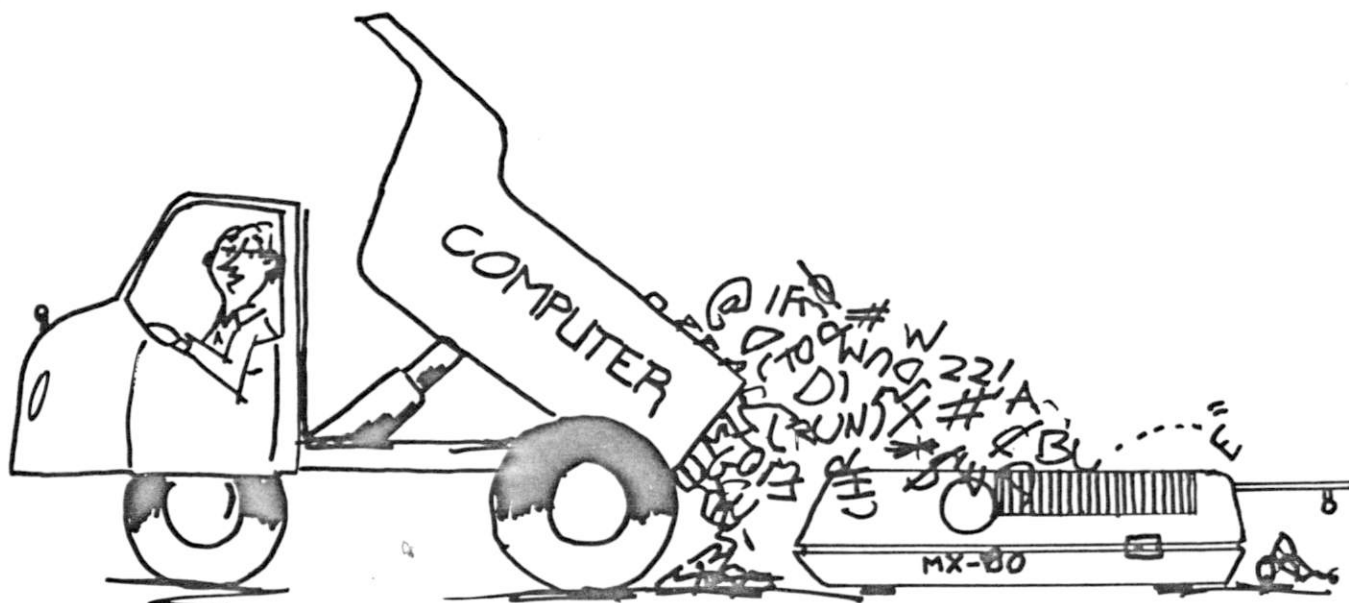
```
>LPRINT CHR$(15)
```

and RUN again.

That's more like it. They look pretty close, don't they?

That's enough for this Chapter. I'll bet you have some favorite graphics programs you can't wait to dump to paper.

Practice with the Dump program, loading in your regular program first, then MERGEing the dump program behind it. If your system doesn't have the ability to MERGE, APPEND or otherwise hook two programs together, the dump program can be typed in behind each program to be printed, and called as a sub-routine when desired.



## Chapter 5

### How Dot Matrix Printing Works

To really understand DOT MATRIX PRINTING, we should understand how the print head works. It contains 9 “needles” or “wires” positioned one above the other, as seen in Figure 5-1.

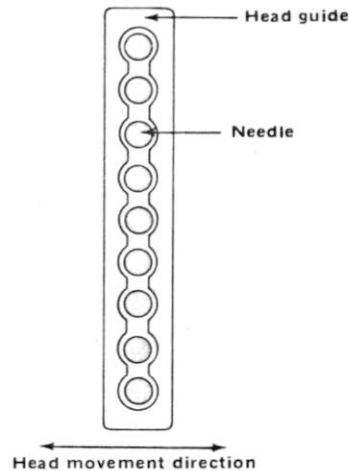


Figure 5-1

Each needle is driven by a “gun”, or electromagnet which “fires” when told to do so by the electronics inside the Printer. The Printer electronics responds to ASCII code instructions received from your computer.

As an Example, to print the letter “I”, wires 1 and 7 are first fired. They hit the ribbon. That makes marks on the paper where the left top and bottom of the letter “I” should be.

Then the head shifts over a bit and wires 1,2,3,4,5,6, & 7 all fire at once, printing the center of the letter “I”. One more shift and one more shot of needles 1 & 7 and the letter “I” is completed. (See Figure 5-2)

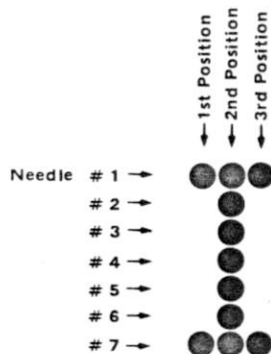


Figure 5-2

This is called "dot matrix" printing. Dots are printed according to a pre-designed "matrix" or "grid" system, where each letter, number and punctuation mark is formed by an arrangement of dots. As we have seen, this complex printing process takes place very quickly.

The needles fire only in groups or clusters. The firing patterns are already programmed inside the Printer, matching the Alpha/Numeric (Letters and Numbers) and graphics characters found in our ASCII charts. (No, we cannot fire an individual needle.)

Appendix G (Part I) shows all our Alpha/Numeric characters constructed within a 9 dot high by 5 dot wide matrix. Every letter, number and punctuation character is designed to fit within that box. (See the lower case letters.)

Notice that the 6th column is always empty. In fact, it isn't even shown. It's reserved for horizontal spacing between characters.

No Alpha/Numeric character is wider than the "standard" 5 dots. If you look closely, it might appear that letters, like the "O" are wider. Actually, they contain more than 5 dots across the width, but those dots are spaced a bit closer than those in a "T". A "W" has even more horizontal dots, but its overall width does not exceed the width of the "standard" 5 dot pattern. The middle dots are just compressed to make more attractive characters.

Our smallest Alpha/Numeric character, the period, requires only 4 dots — 2 by 2. That makes the highest character resolution "4 dots".



Figure 5-3

Graphic characters are a bit different, as shown in Appendix G (Part II) and Figure 5-4.

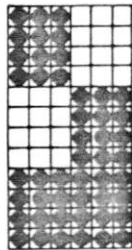


Figure 5-4

There is no empty column on the right because there are no spaces between graphics characters, either on the side, top or bottom.

Since the gun has only 9 wires to fire at a time, graphic characters taller than 9 require 2 passes of the print head. We saw this in Chapter 2 as program line 40 printed graphics, and repeatedly during the Chapter 1 checkout printing.

A quick glance at Appendix G Part II shows that even the smallest Graphic character (ASCII 161) requires the firing of 4 needles, 3 times. Our best (or highest, as they say) Graphics resolution is therefore 12 dots.

Our smallest Alpha/Numeric character, the period, requires only 4 dots — 2 by 2. That makes the highest non-graphic resolution 4 dots. For a complete listing of all our standard MX-80 characters, look at the printout from Chapter 1. It is still hanging on your wall, isn't it?

### How It All Lays Out On Paper

To get a better idea of how these and the remaining characters are distributed on paper, type in this program:

```
10 LPRINT CHR$(222) ; "T" ; CHR$(222) ; "g" ; CHR$(222)
20 FOR N=1 TO 4 : LPRINT CHR$(222) ; : NEXT N
```

(Note to TRS-80 owners. If you have not installed a lower case modification kit, you can still type the lower case "g" required in LINE 10 by pressing the SHIFT key when typing "G". It will show on the screen as a Capital letter, but print out as a small one, having sent ASCII 103 instead of 71. This is true for all the letters. Without the modification, the SHIFT key acts just the opposite of a regular typewriter.)

and RUN.



Figure 5-5

Figure 5-5 shows that each Graphics character took 12 vertical positions and 6 horizontal positions, maximum. The letter "T" took 7 vertical and 5 horizontal.

(If you have a magnifying glass or super-good eyeballs, take a close look at your print-out to verify Figure 5-5.)

The letter “g” has what is called a “descender”. The bottom of the letter must descend below the usual “bottom line”, the 7th line, in order to fit. A “g” or a “y” looks pretty funny if it can’t descend below the “line”. Descenders usually drop 2 dots below the line, requiring the firing of wires 8 and 9. See Appendix G.

Careful examination of our print-out verifies that each print position is 12 dots high by 6 dots wide. Each graphics character occupies the entire space, whether it prints a dot in every space or not. Alpha/Numeric characters use only the top 7, unless there is a descender, in which case they can use the top 9, and all are the equivalent of 5 blocks wide. The 6th column is blank, used for spacing between Alpha/Numeric characters. Verify these facts in your own mind before continuing on.

One last idea. There are ordinarily 5 dot spaces between lines. If a character needs a descender, however, only 3 dot spaces are left between the bottom of that character and the top of the line below it.

This all really makes sense if we think it through, doesn’t it? You don’t have to be a genius to understand it. Just willing to think it through.

### The Line Spacing Is Variable

Now that we understand the dot matrix concept and the standard 12 dot “top-to-top” vertical spacing, we’ll learn how to modify it.

First off, we have a choice of 3 “easy to use” line spacings:

HEIGHT (INCHES)	VERT DOTS	SOFTWARE CODES
1/6	12	Esc “2”
1/8	9	Esc “0”
7/72	7	Esc “1”

Let’s try a simple test program and vary the spacing.

Type in:

```
10 LPRINT "LINE ONE"
```

```
20 LPRINT "LINE TWO"
```

```
30 LPRINT "LINE THREE"
```

and RUN.



Nothing new here. Just like we're used to.

At the command level type:

```
>LPRINT CHR$(27) "0"
```

and RUN it again.

```
LINE ONE
LINE TWO
LINE THREE
```

Figure 5-6

That's what 1/8" spacing looks like. From the top of one character to the top of one on the next print line is only 9 dots, instead of our usual 12. Getting a little snug.

Let's tighten it up a bit more by typing:

```
>LPRINT CHR$(27) "1"
```

and RUN again.

```
LINE ONE
LINE TWO
LINE THREE
```

Figure 5-7

That's too tight for comfort. 7/72 of an inch spacing with only 7 dots from top to top. let's go back where it's more comfortable.

```
>LPRINT CHR$(27) "2"
```

and RUN.

```
LINE ONE
LINE TWO
LINE THREE
```

Figure 5-8

That's better. Any of those commands can be built into a program as well as used at the command level.

### For the Afficianado

The 3 options above cover the needs of most users, but there are many more "fine grain" options that can be used to "tweak" the vertical spacing. We will lightly explore how it's done to demonstrate the capability. Those who get really jazzed by this sort of thing can let their own imagination fly them away to where it feels good.

## Life in the Fast Lane

Everybody add line 5:

```
5 LPRINT CHR$(27) "A" CHR$(140) CHR$(27) "2"
```

and RUN.

Same as before. We changed nothing. Then what is line 5 all about?

Line 5 contains the following 3 distinct commands:

1. An Escape "A". It opens Pandora's box to let us set the vertical line spacing accurate to a **single dot** — 1/72nd of an inch. Mercy!
2. A CHR\$(## + 128). Here we can specify how many **dots** we wish to roll down between the top of one line's printing and the top of the next. We pick our number of **dots** and add 128. Remember, our "standard" 6 lines per inch means we need 12 vertical dots.  $12 + 128 = 140$ . (Aha!)
3. An Escape "2". This activates the charcoal. We must "hit" an Escape 2 for the previous scheme to work. It doesn't have to be on the same program line as we have it here. It can be placed in a later line when we actually want the change in spacing to take place (if we want to). This can allow us to cut back and forth between different line spacings. *Oh*.

To dramatically illustrate the possibilities this feature opens up, add or change the following lines:

```
2 S=129      ' 1 + 128
```

```
5 LPRINTCHR$(27) "A"  CHR$(S)  CHR$(27) "2"
```

```
10 LPRINT"LINE ONE",S-128
```

```
40 S=S+1    :  GOTO 5
```

Line 2 starts us off with a dot spacing of 1.

Line 5 brings the dot number "S" in as a variable

Line 10 is changed to print out the dot spacing

Line 40 adds one dot to S, and sends us back to 5

Let it RUN it until the dot spacing reaches at least 24.

WOW!

## But What Do It Mean, Boss

*(Well, Rochester, it's like this . . .)*

Where the number "1" is printed, the lines are spaced only one dot apart, instead of the usual 12. That's why the letters are printed all over each other.

Check the print-out and find where the spacing is 12. Doesn't that look more like what we're used to?

Now look at "24". Aha! Double spacing. By sending the high-powered command in line 5, we plug in our own value of S and make the Printer give us just about any spacing we want between lines. And, we can do it either inside a program or at the command level. That ought to kick the mind into overdrive!

LLIST the program.

The line spacing stayed where it was when we stopped the program. It is now the new "standard", replacing our power-up value of 1/6" and 12 dots.

We can however, shift back and forth to the other 2 more or less "standard" spacings (options "0" and "1"). Whenever we select option "2" again, the spacing returns to wherever we set it. Once the spacing is set with a high-powered option 2 command, there are only 2 permanent ways to reset it:

1. Set it back to 12 dots per line using option 2 itself

```
>LPRINT CHR$(27) "A" CHR$(140) CHR$(27) "2" or
```

2. Turn the Printer OFF, then ON.

Option "2" in the above format is powerful indeed! Turn the Printer OFF, then ON, before continuing to the next topic.



## Underlining

Many computers do not have the capability of sending an ASCII character to temporarily suppress the Line Feed when a Carriage Return is sent. It is therefore only with great agony that we can do underlining. The zealots among us can refer to Appendix C and throw switches 1-2, 1-3 and 2-3 as necessary to do it. For all practical purposes, it's impractical!

Underlining should not be confused with just moving down one line and printing a series of dashes. True underlining requires that we print the line to which we want to add whole or partial underlining. We then Return the Carriage (print head), suppress the Line Feed so the paper does not advance, and print the underline character, ASCII 95. This is called "overstriking". The underline character is printed in position 10 or 11 in our 12 x 6 matrix. Users with LF suppression can also do such things as slashing zeros, sevens, etc.

In its absence, there is a somewhat cumbersome way to underline using the vertical spacing tricks we just learned. It is practical for such things as report headings or special emphasis, where the result is deemed worthy of the extra programming effort.

Type in this NEW program:

```
10 LPRINT CHR$(27) "A" CHR$(133) CHR$(27) "2"  
20 LPRINT "UNDERLINE"  
30 LPRINT CHR$(27) "1" STRING$(9,45)  
40 LPRINT"IS FAKED." ;  
50 LPRINT CHR$(27) "A" CHR$(140) CHR$(27) "2"  
60 LPRINT "IS NOTHING SACRED?"
```

and RUN.

That result isn't too bad, is it? Here is how the program works:

- Line 10 sets line spacing to 5 dots (5+128) & activates
- Line 20 prints something we want to underline
- Line 30 sets the line spacing to mode "1" - 7/72", and prints 9 dashes (not underlines) via STRING\$
- Line 40 prints some more text to show that it looks OK
  - The ; suppresses the LF
- Line 50 returns the line spacing to 12 dots
- Line 60 prints more text to show all is back to normal

... and that's plenty to think about in this Chapter.

## Chapter 6

### Advanced Printing and Endless Form Control

#### Establishing the Facts

The standard “letter size” piece of paper in North America is 11” long. A maximum of 10” is usually used for printing, the remaining inch divided between the top and bottom margins. “Legal size” paper is 14” long, of which no more than 13” is used. Both sizes are 8½” wide, with no more than 8” used for printing.

Standard typewriter or printer spacing permits 6 lines per inch, or 60 lines per 10 inch page. (78 lines for a legal length page.) This means a new line is started each 1/6”.

As we’ve already seen, standard character spacing permits 80 characters per line, or 10 per inch. In Compressed Mode (Control Code 15) we of course have 132 per line, or 16.5 characters per inch.

Having reviewed and documented these dull facts and traditions, it’s important to know that by simple software commands we can change many of these “standards” to suit our needs.

#### So Much For Tradition

Load the Printer with plenty of paper since we are going to use a pile of it. Chalk it up as tuition.

First, let’s see what is really meant by page length. Adjust the paper so the perforation between two sheets is in the middle of the impact platen. Let’s remove the lid for the rest of this Chapter since we’ll be peeking at the platen often. See Figure 6-1.

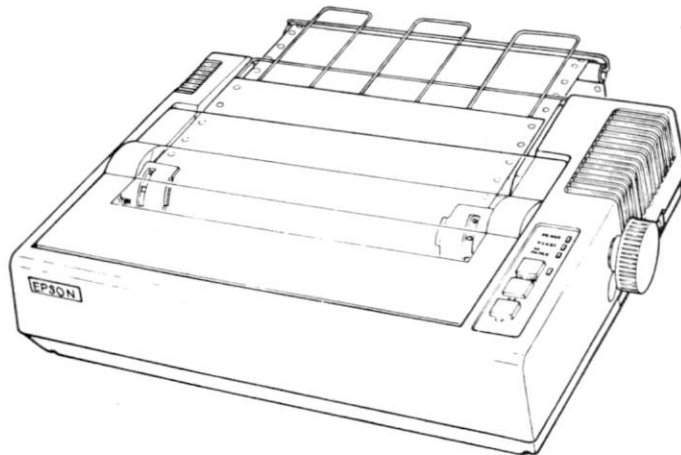


Figure 6-1



This might take a little practice to get perfect. We need to be able to do it right, especially for matching lines and boxes on pre-printed forms such as invoices, bills of sale, purchase orders and the like.

Some users find it easier to position the paper in the ball park, then finish positioning by reference to one of the tractor pins, something else that moves with the paper, or some point over which the paper travels. Practice will make perfect.

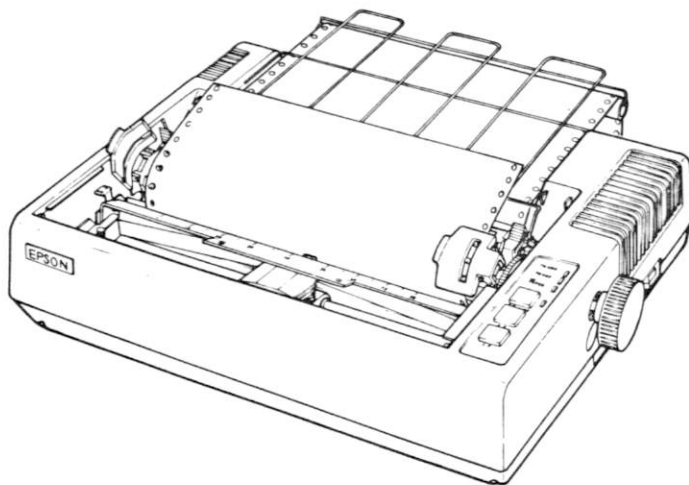


Figure 6-2

Type in this program:

```
10 FOR N = 1 TO 66
20 LPRINT N
30 NEXT N
```

and RUN.

Well, that was singularly unexciting. What did it tell us?

1. It told us that the paper accepts **exactly** 66 rows. No more and no less.
2. Since the numbers all line up in a nice column, and there's no vertical spacing between them, the Printer does not automatically insert margins at the top or bottom of a page.

## The Form Feed — From the Outside

With the paper set where we left it, a perforation in the middle of the platen, let's send the Form Feed command, 12+128=140. Type:

```
>LPRINT CHR$(140)
```

*GOOD GRIEF! THE TRAIN OVERSHOT THE STATION!* Doesn't life have enough trials without this?

Apparently not. An external Form Feed command acts the same as the FF button on the Printer, with one extremely important exception. Since we have to shove commands down the line from BASIC with an LPRINT, and each LPRINT automatically sends a CR and a Line Feed (over which we have no control), each time we send an LPRINT we toss one extra Line Feed into the soup. Terrific!

## So Now What Do We Do?

Well, the problem isn't going away, so we'll confront it head-on.

First, to console ourselves a bit, let's go back to the Printer, push it OFF LINE, push FF, and see what happens.

Yep, the paper went right back to its proper place at the top of the next form. (Yes, yes, this is using up paper. Time is more valuable than the paper, so forget it.)

The whole point of this exercise is:

1. With every LPRINT comes a Line Feed.
2. The Printer counts every LF and knows where the paper is at all times, (even if WE don't).

## The Form Feed — In BASIC Software

Let's modify our resident program to print only 25 lines, then have it automatically roll the paper up to the top of the next form. Change it to read:

```
10 FOR N = 1 TO 25
20 LPRINT N
30 NEXT N
40 LPRINT CHR$(140) ' FF CODE 12
```

and RUN

*CURSES!* (the villain said), there's that extra Line Feed again. We overshot the top of the next form.

Well, it's back to the Printer buttons again. OFF LINE then FF will get us to the top of the next sheet.

### A Little Homespun Philosophy

Having discovered the problem, and it is a very real and vital one, let's analyze the needs of the average user of the Form Feed (FF) command. Typically, he is a businessman sending out a long string of bills. Or, perhaps it's an organization sending out a mess of letters, or printing address labels. In any case, what happens is repetitive, with perhaps certain places on a pre-printed form being hit by the Printer to fill in certain information.

This means the Printer runs continuously. The program will certainly not stop after each form is printed. The program printing all these forms will have the computer in a continuous loop, READING in and processing new information from Disk, Tape, Data Lines, or even from the keyboard, then printing it in a specific format on the forms.

Understanding and appreciating that concept is vital to understanding how FF is used.

Now, if we can **delay** execution of the extra Line Feeds caused by sending form control commands, we can effectively forget them. Sort of like taxes. Delay them long enough and they don't matter.

We learned long ago in our study of Elementary BASIC that a semicolon (;) at the end of a PRINT line temporarily suppresses its Line Feed and Carriage Return. But when the program comes to its END, the suppressed LF/CR catches up. What would happen if we put our LPRINT CHR\$(140) Form Feed command into the continuous loop with the forms printing statements? You got it!

(NOTE: The suppressed LF/CR's don't just keep adding up and dump out at the END, spilling all over the floor. Only 1 of them can hang over our head, awaiting disposition.)

### *CHARGE!*

Let's modify our resident program to do it. It requires 2 steps:

```
40 LPRINT CHR$(140); ' FF CODE 12
50 GOTO 10
```



1. Addition of a semicolon in line 40.
2. Closing the loop so the program can't END

and RUN.(Forget about the paper!)  
 (If you have to fuss, worry about Dutch  
 Elm disease.)

When the floor is strewn with paper and the point is made, you may hit BREAK, and let a silent smile creep across your ugly puss. Success is sweet, eh Crock! (*CRUNCH!*)



## ONWARD

Suppose we're not using full 66 line forms. When's the last time you got a statement on a sheet of paper that big? Oh really? And it started out "Greetings from the President . . ."? Well, anyway.

Let's suppose we're printing on forms that are only half that size, 33 lines per sheet. How do we set the Form Feed to automatically go to a form that isn't 66 lines long? Tho't you'd never ask. (Tho't we'd never get here!)

Escape "C" opens the Printer door for setting the form length. See Line 5:

```
5 LPRINT CHR$(27) "C" CHR$(33) ;
```

Then, CHR\$(##) specifies how many lines are on the form. In this case we set it to 33. The semicolon at the end delays the LF until the program ends. Type in line 5:

```
and RUN (for several sheets)
```

Remarkable! All this really does work.

There's not a lot more to say about FF and LF. It's easy to let them get us into trouble, yet so easy to avoid. If the concept is giving you problems, go back and restudy the HOMESPUN PHILOSOPHY, and start our learning program again from the beginning.

## Horizontal Tabs

All Printer features must receive a message from the computer telling them to get to work. Type in this simple NEW program:

```
10 FOR N = 0 TO 79
20 LPRINT TAB(N) ;N
30 NEXT N
```

and RUN.

OK gang. Different folks are getting different results. Most users will, while making loud and angry noises, see the Printer TAB over to 63, then the carriage return to print position 0 and continue from there. This is a computer software deficiency — it's not sending the right messages to the Printer. Very common.

Those users who find their Printer merrily printing numbers in 78 different columns can give their computers a little pat on the — ah, keyboard. (TRS-80's released after Spring 1980 with the abbreviated power-up message MEM SIZE? R/S L2 BASIC work fine.)

Computer users who are experiencing success can go one step further. Let's go into compressed character mode and see just how many positions we can TAB over to. Change the program to read:

```
5 LPRINT CHR$(15);  
10 FOR N = 0 TO 132  
20 LPRINT TAB(N);N  
30 NEXT N  
40 LPRINT CHR$(18)
```

and RUN.

How sweet it is. We TAB to 131. Everyone else just groan, while your Printer drops back to start again at the left with numbers 64 and 128.

All is not lost if your computer can't match the Printer's ability. If printing on a form that uses the full 80 (or 132) column width, use program TABs as long as possible, then insert spaces the rest of the way. It's frustrating (believe me, I know), but it's about the only universal way out. And, it does work.

## The PRINTER has Horizontal TABs

Yes, the MX-80 itself has the ability to internally store Horizontal Tab positions. This is not the same as just being told by the BASIC program to TAB over so far. Once set, we can quickly shift to each internal TAB point by use of a control code, CHR\$(137)

Enter this NEW program:

```
10 LPRINT CHR$(27) "D" CHR$(138)CHR$(158)CHR$(188) CHR$(128);  
20 LPRINT CHR$(137) "SEE";  
30 LPRINT CHR$(137) "SPOT";  
40 LPRINT CHR$(137) "RUN"
```

then RUN.

Here's what it's all about:

Line 10, using Esc"D" opened the door to accept H TABs  
Line 10 set TABs at 10, 30 and 60 using CHR\$(## + 128)  
Line 10 closed the TAB door with a NULL, CHR\$(0 + 128)  
Line 20,30 and 40 called the TABs, with CHR\$(9 + 128)

If you find the advantages of this system a bit elusive, welcome to the crowd. It seems like a lot of work to TAB when we can (the computer allowing) do the same thing in the BASIC program.

There are some advantages to weigh, however.

FIRST: It does let us TAB beyond the limits of many computers without having to resort to cumbersome PRINT statements with lots of blank spaces. This is especially valuable when printing in the 132 column mode.

SECOND: By setting all the TABs in one line, changes to all lines which call them can be made by changing that single line. (True, the same thing could be done in BASIC.)

Anyway, that's how we can set TABs in the **Printer**, as distinguished from "TABbing" built into the program software.

## Vertical Tabs

Vertical TABS are similar to Horizontal ones. While we can get to any line on the paper by using enough LPRINTs, VTABS can simplify programming.

Replace the current program with this NEW one:

```
10 LPRINT CHR$(27) "C" CHR$(33) ;
20 LPRINT CHR$(27) "B" CHR$(138) CHR$(148) CHR$(128) ;
30 LPRINT "VERTICAL" CHR$(139) "TABS" CHR$(139) "WORK"
40 LPRINT CHR$(140) ; : GOTO 20
```

and RUN

Smooth as silk. The explanation of how the program works is almost as simple:

Line 10 sets for form length at 33 lines.

Line 20 Esc"B" opens the door to receive V TABS

Line 20 sets Vert TABs at 10 (10+128) and 20 (20+128)

Line 20 closes the TAB door with a null 0. (0+128).

Line 30 prints "VERTICAL", TABs down to the 10th line and prints, TABs down to 20 and prints. CHR\$(139) is the TAB "key"

Line 40 sends the Form Feed and keeps the loop closed.

Note the semicolons at the end of the commands in lines 10,20 and 40. Without them we would slip a line here and there. The Printer will count every LF, but unless we are printing on a form requiring every line to be positioned just so, we would miss one now and then.

Did you notice that use of Vertical TABs sends the paper upward in a nice smooth flow? If we did the same thing in BASIC with LPRINTs, the paper would move jerkily as each Line Feed was executed.

Carefully and properly done, Vertical TABs combined with Form Feeds are very powerful programming tools for business applications.

## Chapter 7

### Advanced Graphics

In the earlier Chapter on Graphics, we concentrated on the graphics character set itself, how its ASCII numbers are offset to allow use of more reliable control codes, and how to dump displays from screen to printer. In this Chapter we will practice using the Printer's unique capabilities to create graphics, some of which cannot even be duplicated on the screen.



We have several special things going for us:

1. The Printer's vertical spacing can be adjusted to increment as small as 1/72 of an inch.
2. We can switch between 80 and 132 characters per line.
3. We can switch between single and double width.

We have learned how to utilize all 3 of the features above, but let's experiment a bit more with the first one. Type in this program:

```
10 LPRINT CHR$(27) "A" CHR$(1) CHR$(27) "2"
20 FOR N = 1 TO 20
30 LPRINT TAB(40); "."
40 NEXT N
50 LPRINT CHR$(27) "A" CHR$(12) CHR$(27) "2"
```

and RUN.

**Figure 7-1**

Here's what happened:

- Line 10 set the vertical spacing to 1/72 of an inch.
- Lines 20-40 printed 20 periods in a vertical stack.
- Line 50 reset the vertical spacing back to 1/6 of an inch.

We created our own new character. It may not be award winning, but the possibilities this sort of thing opens up for the creative graphics designer are nearly endless.

## The Big Hammer

We of simpler mind, however, better practice more everyday things, such as the following "notice". It can be easily added to any outstanding bill. What it lacks in subtlety is compensated for in forthrightness.

Type in:

```
10 LPRINT CHR$(27) "E"      ' SETS EMPHASIZED MODE
20 LPRINT CHR$(27) "G"      ' SETS DOUBLE STRIKE MODE
30 LPRINT CHR$(14)TAB(10) "WHERE IS YOUR PAYMENT?"
40 FOR N = 1TO13 : LPRINTTAB(45+N)CHR$(172); : NEXT N
50 LPRINT CHR$(27) "F"      ' TURNS OFF EMPHASIZED MODE
60 LPRINT CHR$(27) "H"      ' TURNS OFF DOUBLE STRIKE MODE
```

and RUN.

**WHERE IS YOUR PAYMENT?**

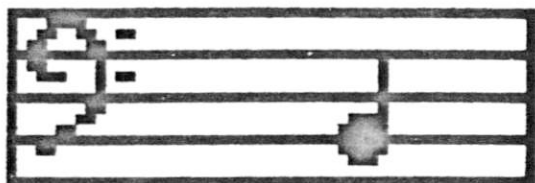
Figure 7-2

Pretty hard to miss that one! Line 30 turns on double width and delivers the message. Line 40 underlines PAYMENT with the graphics character ASCII 172. If we spend a little time to really understand how very simple programs like this are created, we really don't need much talent to create similar ones to match our individual graphics printing needs.

## Genius At Work

We will spend the remainder of this Chapter studying how to apply a very simple "universal style" program to create a rather stunning company symbol or "Logo". This same programming style can be used with an endless variety of other ideas, and is well worth our very careful study.

Examine Figure 7-3 very carefully. It is the end product we are going to create. We will develop it, a line at a time, dissecting it in minute detail. When we're done, you should be able to print your company logo, family coat of arms, fraternal symbol, or whatever suits your fancy.



**BASS CLEF MUSIC CO.**  
 816 HARMONIC WAY  
 MUSICTOWN, USA

Figure 7-3

Beautiful, isn't it. (We've squeezed it up a little here in the Manual to make it fit, but on the Printer it should be 7-11/16" wide.) Hitch up your pants and tighten your belt. Here we go -- a line at a time:

Let's start our NEW program by carefully typing in these lines:

```

20 READ A : IF A>100 LPRINT CHR$(A+32); : GOTO 20
30 IF A>0 READ B : LPRINT STRING$(A,B+32); : GOTO 20
40 IF A=0 LPRINT : GOTO 20
50 ON ABS(A) GOSUB 60,70,80 : GOTO 20
90 DATA 151,163,159,175,147,163,147,19,131,171,0

```

and RUN.



Figure 7-4

## Analysis of the Program Logic

Look carefully at lines 20,30,40 and 50. Their logic is very straightforward.

We begin by reading a value for A from DATA. That value is then tested up to 4 times, once in each line.

"A"'s value will fall in one of 4 ranges. When its value matches a range, something will happen in the line where the match takes place:

Line 20	greater than 100
Line 30	greater than 0 (but less than 100)
Line 40	exactly zero
Line 50	none of the above, (which only leaves a negative number)

Correspondingly, these 4 categories of numbers are "coded" into our DATA lines. Now look at line 90. How many of the 4 categories of numbers can you find?

ght! The first 3 categories are present. Now, what coding scheme does each category represent?

1. Greater than 100: Reserved for graphic characters. 32 added to the number (for ASCII offset) and printed. Users in restricted TRS-80 mode (SW# 2-4 ON), must remove the +32 in lines 20 & 30.
2. Between 0 and 100: Reserved for the "counter" in STRING\$. If "A" reads a "counter", "B" then reads next DATA number which will be >100, thus graphic. STRING\$ prints B+32 a total of A times.
3. 0: Signals a CR and LF via a plain old LPRINT.
4. If none of above, defaults to line 50 which converts negative value to positive, then ON-GOSUB to a print line (which we have yet to type in).

Even tho we haven't yet completed the program, isn't the logic flow nice and easy to follow?

### Micro-Analysis

Let's now go inside DATA line 90 and see what is being READ into the logic system we examined above.

```
90 DATA 151,163,159,175,147,163,147,19,131,171,0
```

#### One Piece of Data At a Time

151 is a graphics character.  $151 + 32 = 183$ . Check Appendix G to see what it should look like.

Using the Editor, change(C) the comma following 151 in line 90 to a colon, making it look like this

```
90 DATA 151:163,159,175 . . . . etc.
```

and RUN.



Figure 7-5

Sure enuf, that's what we get.

Again using the Editor, move the colon to after the 163. Replace the comma after 151.

What does ASCII 163 stand for?  $163+32 = 195$ .

now RUN, READING 2 pieces of DATA



Aha! The picture is starting to take shape.

Continue in this manner, moving the colon over one step at a time, then RUNning. "Step thru" the entire line making only one change at a time, then RUNning.

GO AHEAD AND DO IT. WE'LL WAIT.

Did you catch the subtle difference in the Error message when STRINGS didn't READ both its pieces of DATA? Did you see LPRINT do its thing when the "Ø" was acted upon? If not, best you go back thru line 9Ø and do this assignment again to get the full impact of the program's development.

There's a lot of work involved in such a project, isn't there? It obviously requires careful planning (or an awful lot of cut-and-try). Since you get the idea, step thru the rest of the DATA entries (coming up) in this same fine detail only if you really want to.

## Continuing On

Everybody add 2 more lines:

```
6Ø LPRINT TAB(4Ø) CHR$(14) "BASS CLEF MUSIC CO." : RETURN
1ØØ DATA 157,191,14Ø,14Ø,175,14,14Ø,156,6,14Ø,174,-1
```

and RUN.

Very nice. The "-1" at the end of the DATA line sent us off to LPRINT the message in line 6Ø -- in Double Width. Double width, as we remember, shuts itself off at the end of any line in which it is used.

Let's type in 2 more lines:

```
7Ø LPRINT TAB(5Ø) "816 HARMONIC WAY" : RETURN
11Ø DATA 181,178,179,176,186,178,177,12,176,181,6,176,186,-2
```

and RUN.

Getting better. The "-2" was snagged by the ON-GOSUB in line 5Ø and execution routed to line 7Ø, which very routinely printed it.

Two more lines:

```
8Ø LPRINT TAB(51) "MUSICTOWN, USA" : RETURN
12Ø DATA 149,128,16Ø,152,135,12,128,184,188,149,6,128,17Ø,-3
```

and RUN.

Routine again, but the amount of design work that went into the program is obvious. The beauty of having this “generic” sort of program is that as long as the general format is followed, big changes need to be made only in the DATA lines, not in the program logic.

Only 2 more DATA lines to go. Type carefully:

```
130 DATA 151,139,135,14,131,175,191,135,6,131,171,0
```

```
140 DATA 141,25,140,142,0
```

Beautiful! Now all we need to do is put the Printer in the Wallbanger mode, and make the printout darker. Add:

```
10 LPRINT CHR$(27) "E" : ON ERROR GOTO 150
```

```
150 LPRINT CHR$(27) "F"
```

and RUN.

PERFECTO! Nice crisp printing. When the DATA runs out, the line 10 ERROR trap sends us to line 150 which returns printing intensity to normal.

### Some Final Thoughts

To quote from my earlier books and speeches, (why not — everyone else does), “its use is limited only by your imagination”.

This sample user program was intended to show how a custom letterhead can be designed and printed. ‘Tis a simple matter to print, leave the paper in place, shift into a word processor program, and print the letter beneath it.

We placed the arbitrary cutoff number for graphics characters at a nice round 100 — for easy learning.

Almost every program is ultimately a custom program. The truly “universal” program may NEVER exist. Have fun designing your own graphics.



## Chapter 8

### TRS-80 Mode, Plus Misc Features

#### TRS-80 Mode

The MX-80 provides a TRS-80 simulation-mode option. Its capabilities are very restricted compared to the MX-80 mode, but features graphics located at the same ASCII addresses as the Model I TRS-80. If you purchased software written for a standard Radio Shack Line Printer and do not wish to modify it to operate in the superior MX-80 standard mode, this can be a powerful advantage.

The trade-off for this advantage is a significant loss. Refer to Parts I and II of Appendix B. Also compare parts I and II of Appendix A. If you are willing to give up the advantages shown, this mode is quite satisfactory.

To select the TRS-80 mode, locate switch 2-4 and flip it to the ON position. Thru the hatch, **that's to your right**. As always, be sure the power cord is unplugged when flipping internal switches.

#### TRS-80 Mode Notes

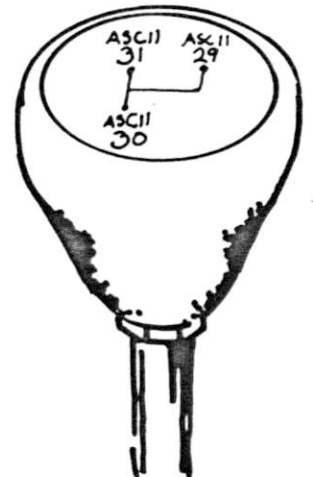
Never add 128 to the ASCII number indicated, since that would put us right in the middle of the graphics set.

The following 3 modes act something like a 3 speed transmission. Shifting into one cancels out the others. Printing can only be in one gear at a time. Modes cannot be mixed on the same line. The Printer is always in **one** of the 3 modes.

ASCII 29 -- COMPRESSED CHARACTER MODE. It behaves generally in the same way as ASCII 15 in the MX-80 mode. It is necessary to LPRINT CHR\$ 30 or 31, switching to another mode, to cancel it.

ASCII 30 -- NORMAL WIDE CHARACTER MODE. In this mode, printing is quite conventional. 10 characters per inch; 80 CPL. The same as when initially powered-up.

ASCII 31 -- DOUBLE WIDTH CHARACTER MODE. Unlike the MX-80 mode, it stays in this mode until switched to one of the above. 5 characters per inch; 40 CPL.



## TRS-80 Mode Escape Codes

The following 4 Escape Codes cluster in two “toggled” pairs. One cancels the other.

ESCAPE “A” --- Causes printing of a normal 8 inch line length. It’s the default mode.

ESCAPE “B” -- Moves both Left and Right margins in about 3/4”. leaving about 6-1/2” for printing. This translates to:

64 Characters in Normal Width

32 Characters in Double Width

105 Characters in COMPRESSED MODE. Indents about 13 characters on the left and 14 on the right.

ESCAPE “6” -- Returns LINE SPACING to 1/6”. Opposite of Esc 8.

ESCAPE “8” -- Sets the LINE SPACING to 1/8”.

Several other ASCII codes are itemized in Appendix B1.

## Other Minor Miscellaneous Codes — MX-80 Mode

The CANCEL and DELETE codes do the same thing. Not all computers can send both, so having both provides a choice.

ASCII 24 -- The CANCEL code, erases all printing data in the PRINT buffer, plus the control codes for DOUBLE WIDTH and COMPRESSED CHARACTER. It seems to be a sort of after-thought for people who change their mind. Not an everyday sort of control code. Activated only if switch 1-4 is OFF.

ASCII 127 -- The DELETE code. Does the same thing as the CANCEL code. Activated only if switch 1-3 is OFF.

ASCII 128 -- (or ASCII 0). The NULL code. Is used to terminate H or V TAB setting command sequences.

## Chapter 9

### Switching to Foreign Language Characters

A multi-lingual Printer? Yes. Switches 2-1, 2-2 and 1-7 allow us to select character sets for 5 different languages. All our work to this point has been in the U.S.A. mode.

#### JAPANESE

The Japanese Katakana character-set is substituted for the TRS-80 graphics-set by flicking a single switch. To make the dramatic comparison, let's start by typing in a simple program and print out the current contents of ASCII 160 to 224 -- where the graphics are stored:

```

10 C = 159
20 FOR N = 1 TO 5
30 C = C + 1
40 IF C = 225 END
50 LPRINT C;" = ";CHR$(C),
60 NEXT N
70 LPRINT : LPRINT : GOTO 20

```

and RUN.

160 =	161 = ◻	162 = ◻	163 = ◻	164 = ◻
165 = ◻	166 = ◻	167 = ◻	168 = ◻	169 = ◻
170 = ◻	171 = ◻	172 = ◻	173 = ◻	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 =

Figure 9-1

That looks familiar.

Next, TURN OFF THE PRINTER, and UNPLUG IT FROM THE WALL.

Using a small pointed object such as a tiny screwdriver, thin ball-point pen, hair-pin, or paper clip, flip switch 1-7 ON. (That's to your right, looking at it from the access hatch. Don't get the wrong switch or we'll have a messy time rechecking all 12 switches.)

Now, plug the Printer back in, turn it back ON and RUN the program again.

160 =	161 = .	162 = 「	163 = 」	164 = 、
165 = ・	166 = フ	167 = 7	168 = イ	169 = 9
170 = 工	171 = 4	172 = 7	173 = 1	174 = 3
175 = 7	176 = -	177 = 7	178 = イ	179 = 9
180 = 工	181 = 4	182 = カ	183 = 4	184 = 7
185 = ケ	186 = コ	187 = 7	188 = シ	189 = ス
190 = セ	191 = ソ	192 = 7	193 = 4	194 = ツ
195 = テ	196 = ト	197 = 7	198 = ニ	199 = 8
200 = 7	201 = ノ	202 = ハ	203 = ヒ	204 = フ
205 = ハ	206 = ホ	207 = マ	208 = ミ	209 = 4
210 = メ	211 = モ	212 = 7	213 = 1	214 = 3
215 = ラ	216 = リ	217 = ル	218 = レ	219 = ロ
220 = ワ	221 = ン	222 = 7	223 = *	224 =

Figure 9-2

AH, SO! (What does it say?)

テウアイトリエ

Figure 9-3

Sayonara /s/ David Lien (in Katakana)

Unplug the Printer again and return switch 1-7 to OFF. (Your left as seen thru the hatch.)

## FRENCH

With the machine still OFF, locate switch 2-2 and turn it OFF. (Don't get the switches mixed up!) Plug it back in.

Type in this NEW program:

```

10 LPRINT TAB(15);"SPECIAL FRENCH CHARACTERS"
20 LPRINT
30 LPRINT TAB(23);"35 =",CHR$(35)
40 LPRINT TAB(23);"64 =",CHR$(64)
50 FOR N = 91 TO 93
60 LPRINT TAB(22);N;" = ",CHR$(N) : NEXT N
70 FOR N = 123 TO 126
80 LPRINT TAB(21);N;" = ",CHR$(N) : NEXT N

```

and RUN.

### SPECIAL FRENCH CHARACTERS

35	=	#
64	=	à
91	=	•
92	=	ç
93	=	§
123	=	é
124	=	ù
125	=	è
126	=	..

Figure 9-4

MERCI! Special characters. Le General would be proud.

The rest of the characters are the same (including the TRS-80 graphics). Those who need to prove that point can rerun the first program in this chapter.

**ENGLISH** (*The Queen's, that is. . .*)

Unplug the Printer. Locate switch 2-1 and flip it OFF. Leave all other switches alone.

Plug the Printer back in, modify the resident program so line 10 reads:

```
10 LPRINT TAB(15);"SPECIAL ENGLISH CHARACTERS"
```

and RUN.

**SPECIAL ENGLISH CHARACTERS**

35	=	£
64	=	ø
91	=	[
92	=	\
93	=	]
123	=	{
124	=	!
125	=	}
126	=	~

Figure 9-5

By Jove! God save the Queen. (And the pound.)

**GERMAN**

Unplug the Printer. Relocate switch 2-2 and flip it ON. Leave all other switches alone.

Plug the Printer back in and change line 10 to read:

```
10 LPRINT TAB(15);"SPECIAL GERMAN CHARACTERS"
```

and RUN.

**SPECIAL GERMAN CHARACTERS**

35	=	#
64	=	§
91	=	
92	=	ö
93	=	ü
123	=	ä
124	=	ö
125	=	ü
126	=	ß

Figure 9-6

ACH DU LIEBER, DAS IST GUT!



## AMERICAN

One more time. Unplug the Printer. Locate and return switch 2-1 to the ON position.

Plug the Printer in and modify line 10 to read:

```
10 LPRINT TAB(15); "SPECIAL AMERICAN CHARACTERS"
      and RUN.
```

### SPECIAL AMERICAN CHARACTERS

35	=	#
64	=	@
91	=	[
92	=	\
93	=	]
123	=	{
124	=	!
125	=	}
126	=	~

Figure 9-7

*OHH Ho-say Can You See?*

## Putting It To Work

Having exercised these capabilities and knowing that they work, now what good are they?

Well, to users who need the special versions for their own purposes, the answer is obvious. Also, unlike what often happens when extra features are added, there are few DISadvantages.

Since each character is assigned an ASCII number, everything we've learned about computer programming is still valid. In actual practice, the special characters will be used primarily for report headings, and in word processor generated correspondence.

We can however **compare** any special character against any other character, or against itself. Since BASIC is an ENGLISH-language programming language, they will probably rarely be used in that way, but each character can even be used as a variable. It's just like the character it replaced with the same ASCII number.

APPENDIX C contains a summary of the switch settings we just exercised.

# Appendix A

## Part I

Graphics — See Appendix G Part I

Blank

NUL		SP	ø	@	P		p	NUL									
0	16	32	48	64	80	96	112	128	144	160	176	192	208	224	240	256	272
DC1	!	1	A	Q	a	q		DC1									
1	17	33	49	65	81	97	113	129	145	161	177	193	209	225	241	257	273
	·	2	B	R	b	r											
2	18	34	50	66	82	98	114	130	146	162	178	194	210	226	242	258	274
DC3	#	3	C	S	c	s		DC3									
3	19	35	51	67	83	99	115	131	147	163	179	195	211	227	243	259	275
DC4	\$	4	D	T	d	t		DC4									
4	20	36	52	68	84	100	116	132	148	164	180	196	212	228	244	260	276
	%	5	E	U	e	u											
5	21	37	53	69	85	101	117	133	149	165	181	197	213	229	245	261	277
	&	6	F	V	f	v											
6	22	38	54	70	86	102	118	134	150	166	182	198	214	230	246	262	278
BEL	'	7	G	W	g	w		BEL									
7	23	39	55	71	87	103	119	135	151	167	183	199	215	231	247	263	279
CAN	(	8	H	X	h	x		CAN									
8	24	40	56	72	88	104	120	136	152	168	184	200	216	232	248	264	280
HT	)	9	I	Y	i	y		HT									
9	25	41	57	73	89	105	121	137	153	169	185	201	217	233	249	265	281
LF	*	:	J	Z	j	z		LF									
10	26	42	58	74	90	106	122	138	154	170	186	202	218	234	250	266	282
VT	ESC	+	:	K	[	k	]	VT	ESC								
11	27	43	59	75	91	107	123	139	155	171	187	203	219	235	251	267	283
FF	'	<	L	\	l	;		FF									
12	28	44	60	76	92	108	124	140	156	172	188	204	220	236	252	268	284
CR	-	=	M	)	m	!		CR									
13	29	45	61	77	93	109	125	141	157	173	189	205	221	237	253	269	285
SO	..	>	N	^	n	~		SO									
14	30	46	62	78	94	110	126	142	158	174	190	206	222	238	254	270	286
SI	/	?	O	_	o	DEL		SI									
15	31	47	63	79	95	111	127	143	159	175	191	207	223	239	255	271	287

ASCII Chart for MX-80 Mode Switch Settings

Part II

Graphics — See Appendix G Part II

0	16	SP	0	@	P	,	p	128	144	160	176	192	208	224	240		
1	17	!	1	A	Q	a	q	129	145	161	177	193	209	225	241		
2	18	•	2	B	R	b	r	114	130	146	162	178	194	210	226	242	
3	19	#	3	C	S	c	s	115	131	147	163	179	195	211	227	243	
4	20	\$	4	D	T	d	t	116	132	148	164	180	196	212	228	244	
5	21	%	5	E	U	e	u	117	133	149	165	181	197	213	229	245	
6	22	&	6	F	V	f	v	118	134	150	166	182	198	214	230	246	
7	23	'	7	G	W	g	w	119	135	151	167	183	199	215	231	247	
8	24	(	8	H	X	h	x	120	136	152	168	184	200	216	232	248	
9	25	)	9	I	Y	i	y	121	137	153	169	185	201	217	233	249	
LF	10	*	:	J	Z	j	z	122	138	154	170	186	202	218	234	250	
11	27	ESC	+	;	K	(	k	123	139	155	171	187	203	219	235	251	
12	28	'	<	L	\	l	l	124	140	156	172	188	204	220	236	252	
CR	GS	-	=	M	)	m	;	125	141	157	173	189	205	221	237	253	
14	30	RS	>	N	^	n	~	126	142	158	174	190	206	222	238	254	
15	31	US	/	?	O	_	o	DEL	127	143	159	175	191	207	223	239	255

ASCII Chart when Switch #2-4 Set for Special TRS-80 Mode

## Appendix B: Control Codes

### MX-80 Mode

#### CONTROL CODES Require LPRINT CHR\$(#)

ASCII #	FUNCTION
% 0	NULL. Follows ESC“B” & ESC“D” as terminator for TABS.
* 0	Sets line spacing to 1/8". Follows ESC“A”.
* 1	Sets line spacing to 7/72". Follows ESC“A”.
* 2	Returns line spacing to wherever previously set with ESC“A”. (Defaults to 1/6" spacing)
7	Sounds buzzer for 3 1/2 seconds.
10	Line Feed (LF)
% 11	Vertical Tab (VT)
% 12	Advances paper to TF(top of next form).
13	Carriage Return (CR)
14	Turns on DOUBLE WIDTH mode. (SO)
15	Turns on COMPRESSED CHARACTER mode. (SI)
% 17	“Selects” printer. Lets it accept data. (DC1)
% 18	Turns off COMPRESSED CHARACTER mode. (DC2)
% 19	“Deselects” printer. Can’t accept data. (DC3)
% 20	Turns off DOUBLE WIDTH mode. (DC4)
% 24	CANCELS unprinted DATA. Resets SI & SO to normal width.
27	ASCII code for ESCAPE.
* 65	ASCII “A”. Sets spacing of LF(line feed) with 0,1 or 2
* 66	ASCII “B”. Sets VT (vertical tabs) up to 64.
* 67	ASCII “C”. Sets FL (form length). Default is 66.
* 68	ASCII “D”. Sets HT (horiz tabs) up to 80 (or 132).
* 69	ASCII “E”. Turns on EMPHASIZED mode.
* 70	ASCII “F”. Turns off EMPHASIZED mode.
* 71	ASCII “G”. Turns on DOUBLE STRIKE mode.
* 72	ASCII “H”. Turns off DOUBLE STRIKE mode.
127	DELETES unprinted DATA. Resets SI & SO to normal width.

NOTE: Numbers flagged with a % MAY require the addition of 128 to make them work reliably. When in doubt, add 128.

Numbers flagged with a \* DO need to be preceded by the Escape Code, CHR\$(27).

## TRS-80 Mode

### CONTROL CODES Require LPRINT CHR\$(#)

ASCII #	FUNCTION
* 6	Returns spacing to 1/6" (Default)
* 8	Sets line spacing to 1/8".
10	Line Feed (LF)
11	Form feed of 61(?) spaces
12	Form Feed of exactly 66 spaces. (Not to top of form!)
13	Carriage Return (CR)
27	ASCII code for ESCAPE.
29	Condensed Character Mode (132 CPL) (GS)
30	Normal Character Width (80 CPL) (RS)
31	Double Character Width (40 CPL) (US)
* 65	ASCII "A". Resets width to "normal" from Escape "B"
* 66	ASCII "B". Reduces line width to about 6 1/2".

NOTE: We must use the ASCII number indicated here while in this mode. If we added 128, we would run into the regular TRS-80 graphics set. No other commands available in this mode.

Numbers flagged with a \* DO need to be preceded by the Escape Code, (CHR\$(27)).

## Appendix C

### The Internal Switches

NEVER set the 12 internal DIP (Dual In-line Pin) switches when the power is on. Turn BOTH the printer and computer OFF.

\*\*\*\*\*

Switch #	Normal Position	Function
1-8	ON	Allows ASCII 17 to "select" and ASCII 19 to "deselect" the printer. That means, if this switch is OFF, we can remotely control whether the printer is allowed to print what is sent from the computer. When the switch is ON, printer is permanently in the "selected" mode and no external command can "deselect" it.

This option is not available if a Radio Shack cable is used.

1-7	OFF	Selects between TRS-80 graphics characters and Japanese Katakana characters. OFF is TRS-80 and ON is Katakana.
-----	-----	--

Both these character sets begin at ASCII 160. In the event that Switch #2-4 is OFF, both graphics sets are shifted down to begin at ASCII 128.

1-6	ON	Allows internal buzzer to sound for about 3 seconds if an "error" occurs. An example would be when the Printer runs out of paper. If the switch is OFF, errors cannot activate the buzzer.
-----	----	--

This switch has no effect on allowing the computer to activate the buzzer by sending an ASCII 7; the buzzer will still sound.

1-5	ON	Delete Code Switch. When the switch is ON, features below are overridden.
-----	----	---

When the switch is OFF, certain data waiting in the print buffer can be deleted. It allows the operator to change his mind at the last minute.

String data, and Condensed and Double Width control codes can be deleted. Escapes A-H and 0-2 cannot be deleted. See switch 1-4, below.

1-4	OFF	Cancel Code Switch. Similar to switch 1-5 above. Both switches are needed since some computers can send only delete OR cancel, but not both codes. Having 2 switches provides for both contingencies. Activated when switch is OFF.
-----	-----	---

**1-3**                    **ON**                    **Auto LF Switch.** This switch controls whether or not a line feed (LF) is automatically added each time the print buffer fills with data. Activated when switch is OFF.

Switch #2-3 is the "master" switch. If it is ON, a LF is added to every CR when the print buffer is full. If it is OFF, each LF must be deliberately sent by software, or added automatically by setting switch 1-2 or 1-3 to OFF.

**EXAMPLE:** Assume 2-3 is OFF: IF 1-3 & 1-2 are OFF, when the print buffer is full, a LF is automatically added. If 1-3 is ON, when the print buffer is full, only the CR is recognized and the print head overprints what is already on paper.

If using a Radio Shack cable, switches 1-3, 1-2 & 2-3 are *ALL* overridden and LF is automatically added to every CR. There is no way to separate them.

**1-2**                    **ON**                    **Automatic CR Switch.** Similar to 1-3, except, if 1-2 is ON, when CR is sent no LF is added. It prints over what is already on the paper.

**1-1**                    **ON**                    **Blank.** Not hooked to anything.

**2-4**                    **OFF**                    When OFF, gives us access to all MX-80 capabilities. None are locked out. Shifts TRS-80 graphic codes up by +32 ASCII positions.

When ON, Printer responds to standard TRS-80 ASCII graphics numbers, but severely restricts Printer's response to control codes.

**2-3**                    **ON**                    Forces automatic LF with each CR. When OFF, LF must be provided via software as needed, or via setting of switches 1-2 & 1-3.

**2-2**                    **ON**                    Used with 2-1 to select foreign character sets. See table below.

**2-1**                    **ON**                    Used with 2-2 to select foreign character sets. See table below.

**Switch Settings For Foreign Character Sets**

2-1	2-2	COUNTRY
ON	ON	U.S.A.
ON	OFF	FRANCE
OFF	OFF	ENGLAND
OFF	ON	GERMANY
See Switch #1-7		JAPAN



## Appendix D

### Using Your MX-80 With Other Than A TRS-80

This Appendix covers special software considerations when using the MX-80 with these computers/connections:

- A. Apple
- B. Pet
- C. RS232C Serial Interface

It specifically *EXCLUDES* hardware considerations, such as how to set the 12 internal switches. That information is supplied in the instruction sets which accompany the special matching cables and hardware designed to interface with your computer.

This section includes such things as how the other computers send ASCII codes down the line if they don't support LPRINT.

#### APPLE

Install the interfacing card in the Apple per the directions that accompany it. Since the Apple interface card sends a Line Feed at the end of every line, we don't need to force one. Accordingly, switches 1-2, 1-3 in the MX-80 should be ON and switch 2-3 OFF. Double spacing will otherwise result.

Apple computers do not separate the PRINT commands to the screen and to an external port, such as a printer. The use of either PRINT or LIST will result in output to only the screen, unless otherwise instructed.

#### Apple Test and Demo Program

Use this program in place of the one in Chapter 2:

```
10 PRINT TAB(23) "GREETINGS FROM THE VERSATILE"  
20 PRINT TAB(32) CHR$(14) "MX-80"  
30 PRINT TAB(57) CHR$(15) "BY"CHR$(14)CHR$(27) "G" "EPSON"  
50 PRINT TAB(30) CHR$(18) "YOU'LL LIKE ME!"  
60 PRINT CHR$(27) "H"
```

but DO NOT RUN IT, YET!

We need to activate the printer port. Assuming you placed the interface card in the Apple slot #1, type:

```
PR#1 <cr>
```

The computer output will now be routed to the printer port, AND to the screen. (If you used a different slot number, use that same number above and in all printer related commands.)

To see if everything is hooked up and properly initialized, type:

```
LIST <cr>
```

and the program should be LISTed on the screen and Printer.

### *However*

The Apple screen is limited to 40 characters, and it imposes the same limitation on the Printer. This severely restricts the MX-80.

To overcome this limitation, we type:

```
<CTRL I>80N <cr>
```

The Control I followed by 80N sets computer output to 80 columns – a good starting match for the MX-80. The “80N” can be any number from 40 to 255 as far as the computer is concerned. The MX-80 might have some other ideas, however.

Having exceeded the 40 character screen limitation, the computer output no longer goes to the screen – only the Printer. *Everything* goes to the printer even **input** statements, and *Nothing* goes to the screen. In this mode, questions requiring keyboard response should be asked by PRINT statements – the answers picked up by separate INPUT statements.

Since all these commands can be sent at either the command or BASIC program levels, it is possible to use software to switch back and forth, using the screen and Printer for alternate outputs, or the same, formatted differently.

To default back to 40 column screen AND printer, type:

```
<CTRL I>I <cr>
```

To switch back to the screen only, type:

```
PR#0 <cr>
```

## Sample Apple Program

To demonstrate alternate use of the screen and Printer in dialogue situations, try this sample program:

```
10 PR#1 : <CTRL I>80N
20 PRINT "THE MX-80 WILL PRINT MORE THAN 40 COLUMNS";
30 PRINT "IN THIS DEMONSTRATION PROGRAM.  JUST WATCH!
40 PR#0 : INPUT"ENTER YOUR NAME";A$
50 PR#1 : PRINT "THANK YOU ";A$;" --- I'M SURE";
60 PRINT" YOU'LL GET USED TO THIS STYLE OF PROGRAMMING."
```

## Other Apple Considerations

The Apple interface card does not allow ASCII codes greater than 127. Codes 128-255 are simply duplicates of 0-127. This means no graphics characters can be printed.

The Apple BASIC interpreter tested does not support the STRING\$ statement. It can be simulated by the technique demonstrated in Chapter 2. For additional information see *The BASIC Handbook*, by Lien.

## Appendix E

### Using the MX-80 With a Standard Radio Shack Cable

There is a difference between the 'official' EPSON printer cable and the ones supplied by Radio Shack. They are wired slightly differently.

The EPSON cable allows separation of the CR(carriage return) and the LF(line feed) commands. This in turn allows such things as underlining, overstrikes to slash zeros, sevens, the letter Z, to "black out" material, etc.

If you do not need to separate CR and LF and already have a Radio Shack printer cable, it should work fine. If purchasing a new cable, buy an EPSON to keep all your options open.

## Appendix F

### Using the MX-80 With the Model II TRS-80

EPSON does not supply a special cable for use with the Model II TRS-80. A standard Radio Shack Catalog number 26-4402 cable may be obtained thru any Radio Shack store. One end connects to the MX-80. The other end plugs into the PARALLEL PRINTER CHANNEL on the back of the TRS-80. Do NOT connect the computer end until directed to do so by the Manual.

The Model II is in many ways similar to the Model I. Its Disk BASIC interpreter uses LPRINT in the same way as the Model I, and the control functions respond generally the same, except as follows:

#### *INITIALIZATION*

The Model II must be initialized with its own FORMS command in order to properly feed a printer. Initialization can be performed either at the DOS or BASIC levels.

1. From DOS: Type:

```
FORMS
```

The computer will respond with:

```
PRINTER READY (Y/N)
```

Assuming the Printer is properly connected and turned ON, type:

```
Y <en> and
```

```
Q <en>
```

Initialization is complete and we are returned back to DOS.

2. From BASIC, type:

```
SYSTEM "FORMS" <en>
```

After finishing the questions and typing:

```
Q <en>
```

we are automatically returned to BASIC.

---

## Glitches

Under certain circumstances Escape“G”, the Double-Strike mode seems to cause “de-initialization”, so should be avoided. Escape“E”, the Emphasized mode seems to work alright, but of course we cannot use it with compressed characters.

A sign of Printer problems is the “I/O ERROR” message. This might mean the Printer has been “de-initialized” and the above procedure will have to be repeated.

## GRAPHICS

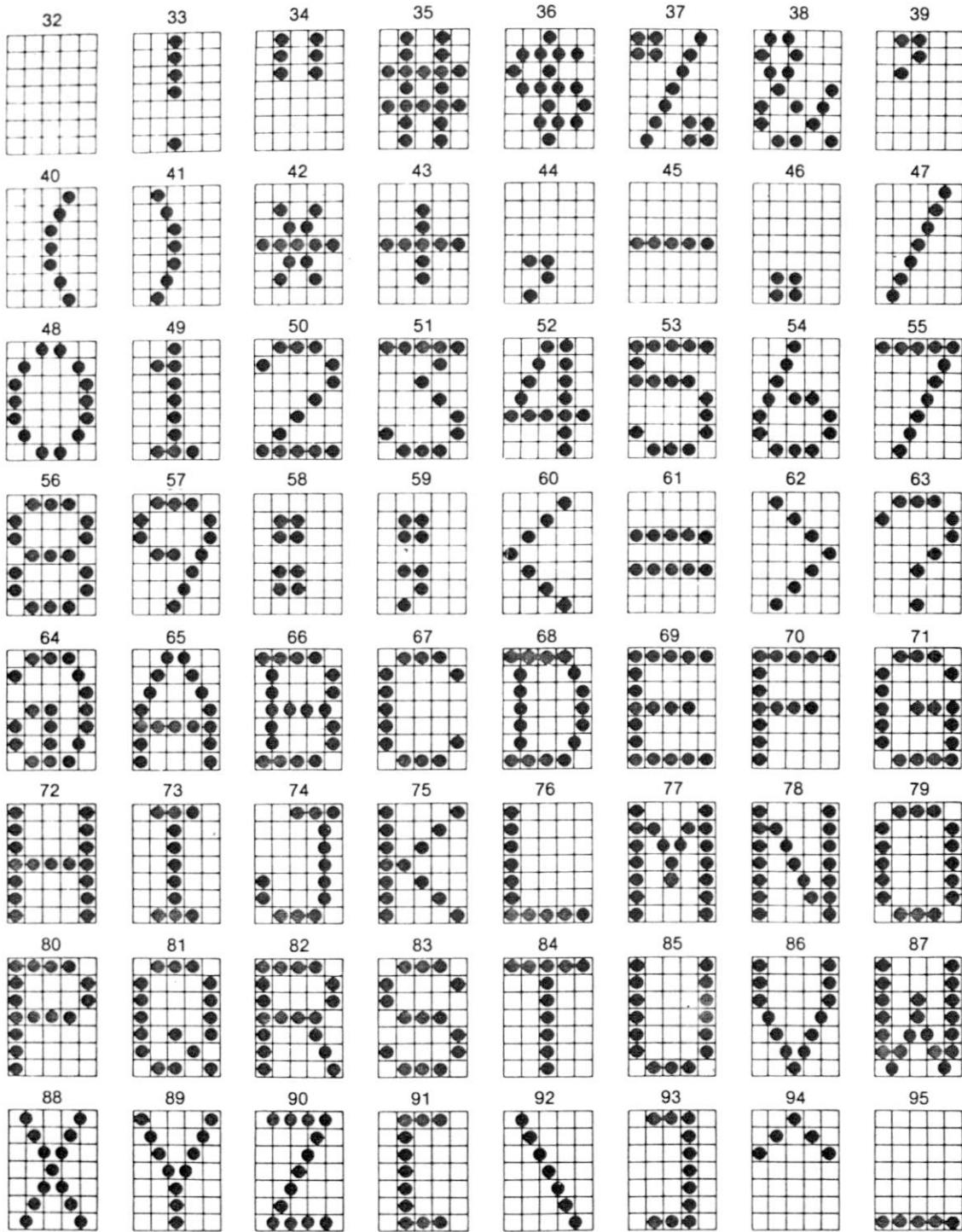
1. The graphics characters are not identical. The Model II has only 32 graphic characters, compared with the Model I's 64. Model II graphics are not a cut-down version of Model I, but include many new characters. This eliminates the possibility of using a “screen dump”.

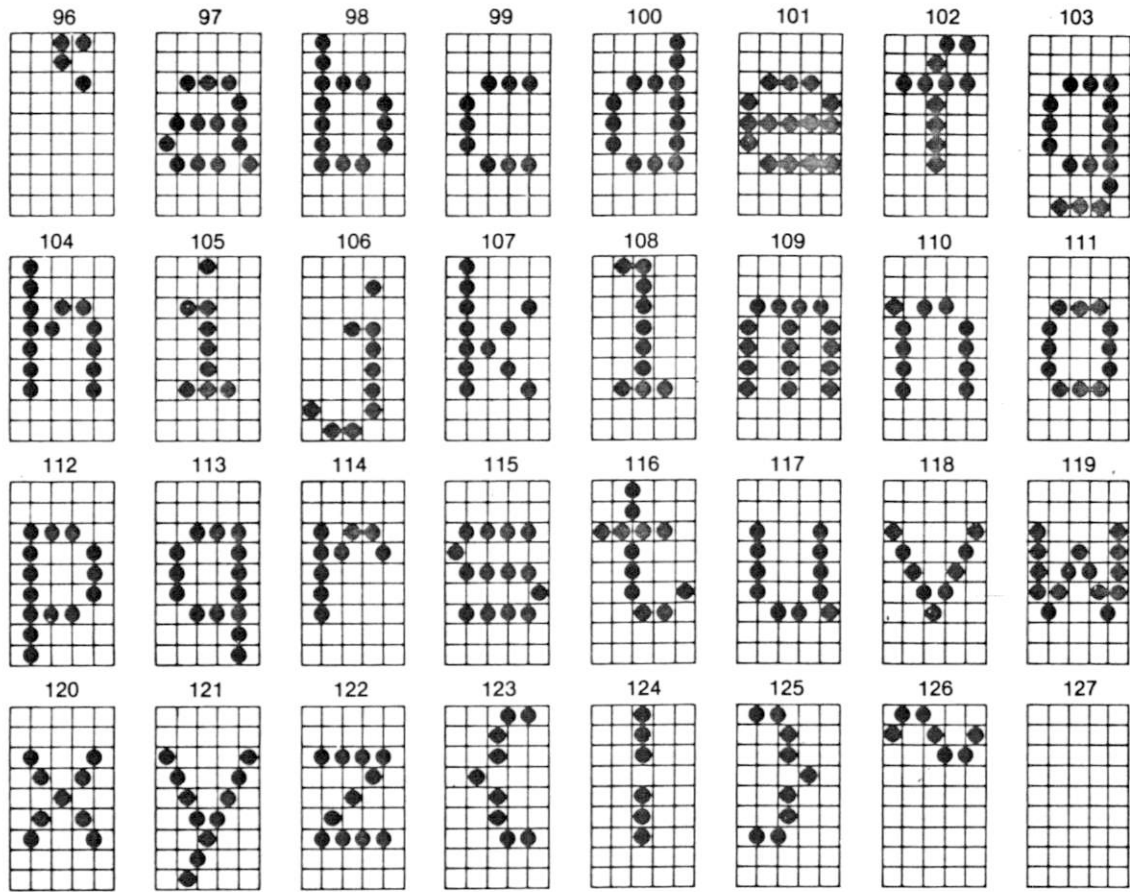
Remember, the Printer uses the graphics set built into the **Printer**, not the computer. A graphics program written for Model I will work just fine on the Model II as long as all graphics characters are called up by ASCII number.

2. The ASCII numbers assigned to graphics are not the same. Model II goes from 128 to 159. You will have to change Model I ASCII numbers to match similar characters in the Model II set. The graphics part of the program in Chapter 2 will not RUN, but the one in Chapter 7 will.

# Appendix G

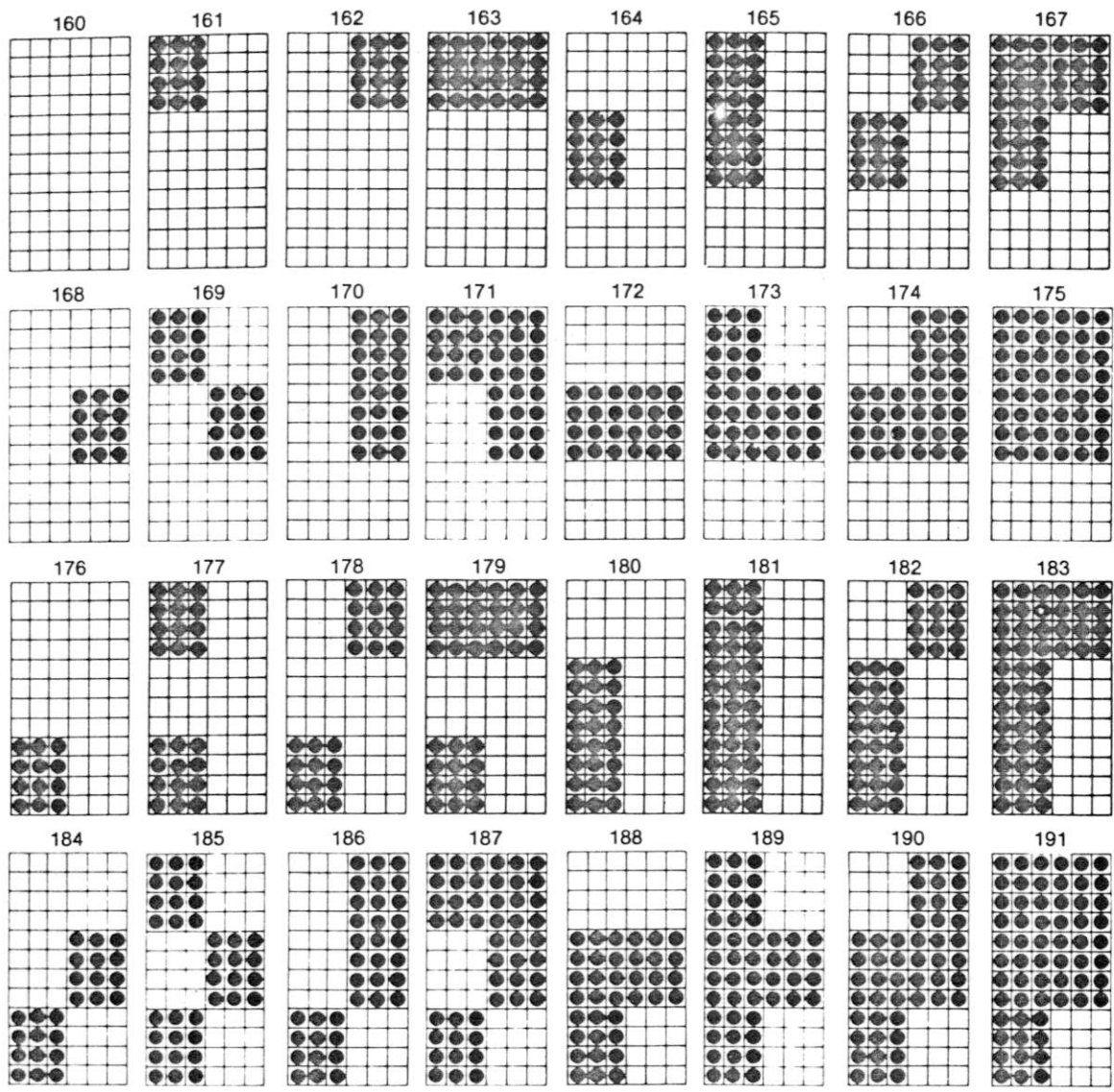
## Part I: Character Fonts

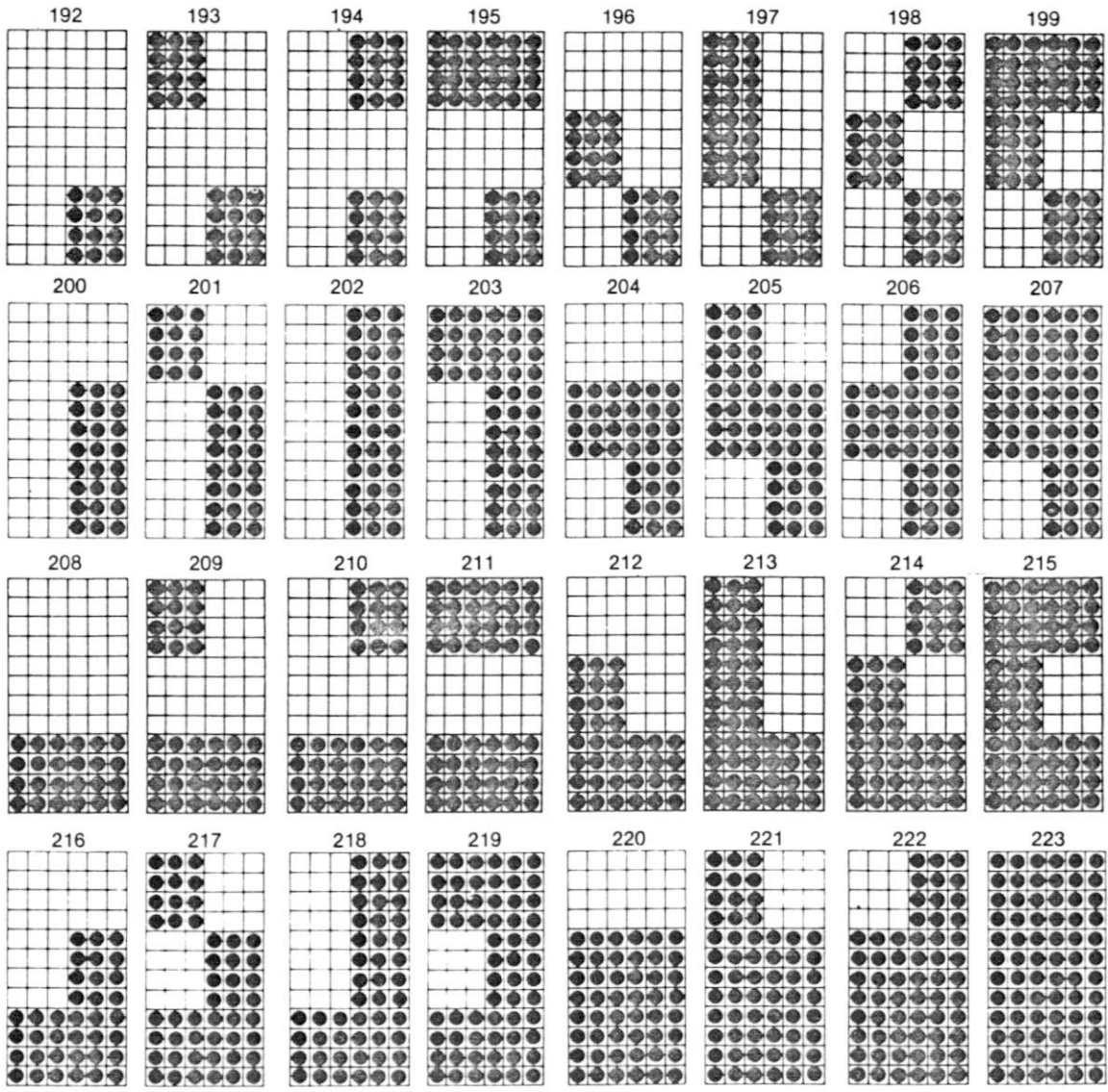






### Part II: Graphics Fonts





## Appendix H

### Specifications

#### Printing

Printing Method	Impact dot matrix printing 96 ASCII characters by 9×9 dot matrix and 64 graphic by 6×12
Speed	80 CPS
Form Width	101.6 mm (4'') to 254 mm (10'') pinfeed type
Line Spacing	6, 8, or n (programmable) lines per 25.4 mm (1'') (Operator Selectable)
Line Length	Up to 132 characters at 16.5 characters per inch.
Paper Drive	Pinfeed
Copies	Up to 3 sheets with 2 carbons (total thickness not exceed 0.3 mm (0.012'')).
Line Feed Time	200 ms
Diagnosis	Allows operator to check printer operation without another test instruments.

#### Interface

Parallel Interface (standard)      Compatible with CENTRONICS

#### Physical

Power	: 115V ± 10% 60 Hz
Power Requirement	: 100 VA max.
Temperature	: Operating 5°C to 35°C (41°F to 95°F) Storage -30°C to 70°C (-22°F to 158°F)
Humidity	: Operating 5% to 90% (no condensation) Storage 0% to 95% (no condensation)
Shock	: Operating 1 G (less than 1 msec.) Storage 2 G (less than 1 msec.)
Vibration	: Operating 0.25G, 55Hz (Max.) Storage 0.50G, 55Hz (Max.)
Insulation Resistance	: 10MΩ between AC Power line and chassis
Dielectric Strength	: Between AC power line and chassis, AC 1KV (R.M.S.) 50Hz or 60Hz is during 1 minute and no abnormal condition shall be observed.
Dimensions	: 374 mm (14.7'') W×305 mm (12.0'') D×107 mm (4.2'') H
Weight	: 5.5 kg (12.1 lbs)

## Appendix I

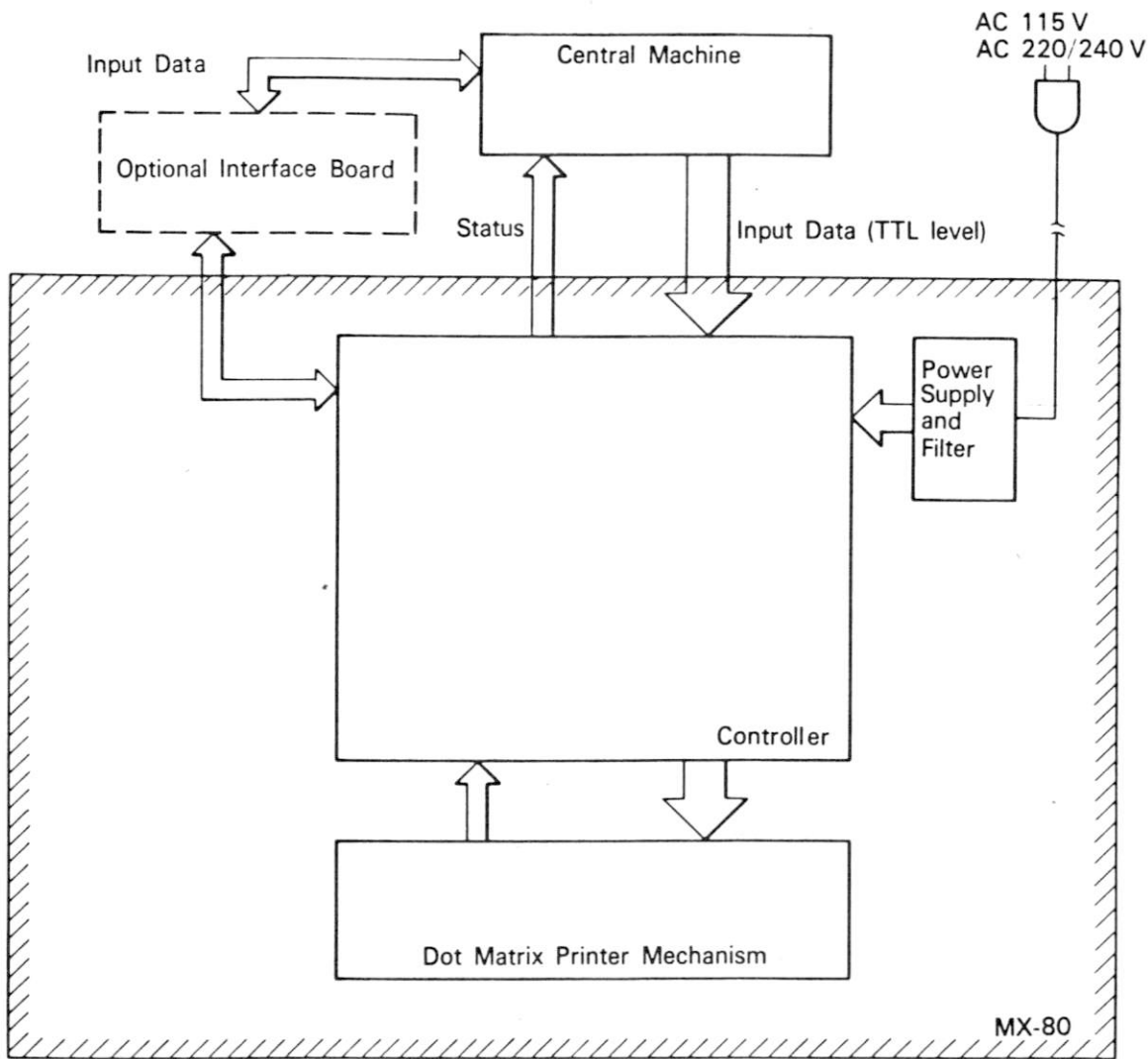
### Function Codes

MX-80 has various function codes described below :

- CR Carriage Return
  - LF Line Feed
  - VT Vertical Tabulation
  - FF Form Feed
  - SO Shift Out ; sets the enlarged printing.
  - SI Shift In ; sets the condensed printing.
  - DC 4 The function of SO is cancelled.
  - HT Horizontal Tabulation
  - CAN Cancel ; all data preceding CAN on a same line are cleared.
  - DEL Delete ; same as CAN
  - DC 1 Selects the printer, enable it to receive data.
  - DC 3 Deselects the printer, preventing it from receiving data.
  - ESC 0 Sets line spacing to 8 lines per inch.
  - ESC 1 Sets line spacing to 7/72" per line.
  - ESC 2 Sets line spacing to 6 lines per inch.
  - ESC 8 Enables data transfer even in paper out.
  - ESC 9 Resets ESC 8 function.
  - ESC SO Same as SO
  - ESC SI Same as SI
  - NUL NULL ; terminates the tab setting sequence
  - BEL BELL ; rings about 3 seconds
  - ESC X Control
- By ESC code and X (alphabets (A~Z)) code, following functions are available.  
 In the following descriptions each of  $n, n_1, n_2, \dots$ , is 7 bit binary numbers.  
 ("+" is for legibility only.)
- ESC A+n  
 Determines a various line spacing. Paper is fed in the rate of  $n/72$ .  
 $1 \leq n \leq 85$
  - ESC B+n<sub>1</sub>+n<sub>2</sub>+...+n<sub>k</sub>+NUL  
 Sets vertical tabs ; Clears all previous vertical tabs and sets new tabs at the lines ( $n_1, n_2, \dots, n_k$ ). ( $1 \leq n \leq 255, 1 \leq k \leq 64$ )
  - ESC C+n  
 Sets the form length.  $1 \leq n \leq 66$
  - ESC D+n<sub>1</sub>+n<sub>2</sub>+...+n<sub>k</sub>+NUL  
 Sets horizontal tabs ; clears all previous horizontal tabs and sets new tabs at locations.  
 ( $1 \leq n \leq 127, k \leq 112$ )
  - ESC E Sets the emphasized printing.
  - ESC F Cancels ESC E function.
  - ESC G Sets the double printing.
  - ESC H Cancels ESC G function.

# Appendix J

## Logic Block Diagram



## Appendix K

## Pinout Signal Chart

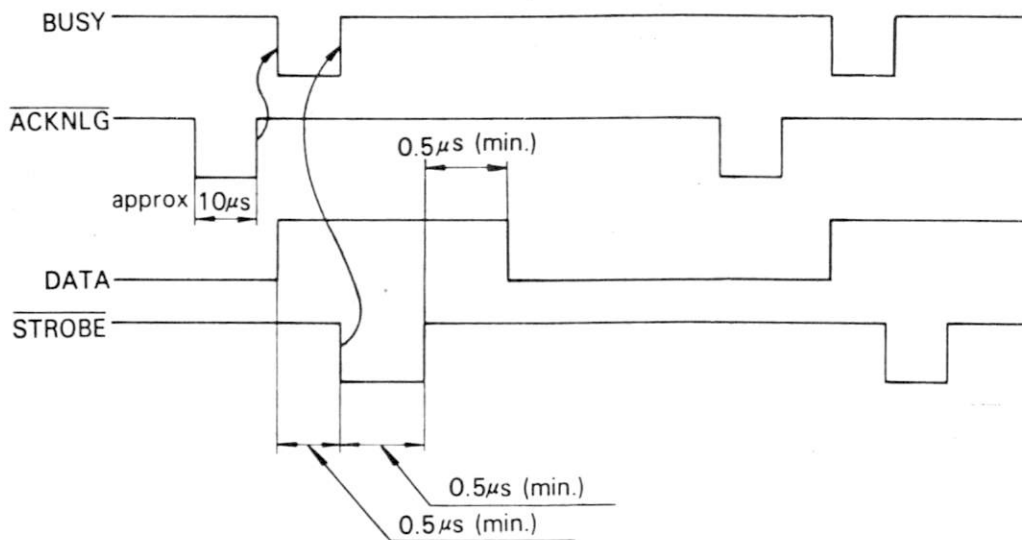
Parallel Interface Those signals listing a return pin require twisted pair lines with the return to logic ground. A 36-pin female mating connector is Amphenol type 57-30360.	PIN	RETURN PIN	SIGNAL NAME	SOURCE	FUNCTION
		1	19	STROBE	Input Device
	2	20	DATA 1	Input Device	Input data levels from 1 st bit to 8 th bit. A high represents a binary one, a low represents a binary zero.
	3	21	DATA 2	"	
	4	22	DATA 3	"	
	5	23	DATA 4	"	
	6	24	DATA 5	"	
	7	25	DATA 6	"	
	8	26	DATA 7	"	
	9	27	DATA 8	"	
	10	28	ACKNLG	Printer	A pulse low indicates that a character data has been received and the printer is ready to accept another character.
	11	29	BUSY	Printer	High level signal indicates that the printer can not receive data. Low level signal indicates that the printer can receive data. This signal becomes high in the following conditions. ① During data entry. ② Printing or printhead's moving time ③ Off Line state ④ Error state
	12	30	PE	Printer	High level signal indicates that the printer is out of paper.
	13		SLCT	Printer	High level signal indicates that the printer is selected.
	14		AUTOFEED XT	Input Device	Low level signal enables automatical line feed, after printing one line.
	15		NC		N/A
	16		0V		Logic ground
	17		CHASSIS GND		Chassis ground
	18		NC		N/A
	31		INIT	Input Device	A 50 $\mu$ s pulse (min.) used to initialize the printer
	32		ERROR	Printer	Low level signal indicates an error condition.
	33		GND		
	34		NC		N/A
	35			Printer	pulled up to +5 VDC
	36		SLCT IN	Input Device	Low level signal makes printer selected condition.

# Appendix K

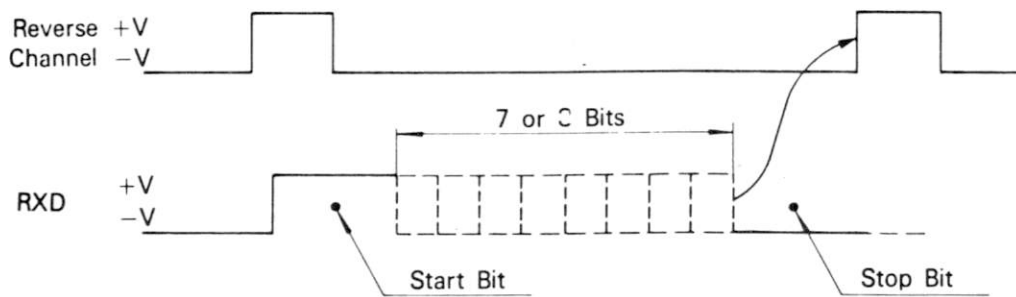
EIA RS-232C Serial Interface with a 25-pin female connector.	PIN	SIGNAL NAME	SOURCE	FUNCTION
	1	Protective Ground		Chassis ground
	2	Transmitted Data	Printer	This signal is at the negative EIA level in the normal mode.
	3	Received Data	Input Device	Received serial data.
	6	Data Set Ready	Input Device	This signal must be at the positive EIA level for the printer to receive data.
	7	Signal Ground		Return path for data and control signals.
	8	Carrier Detect	Input Device	This signal must be at the positive EIA level for the printer to receive data.
	11	Reverse Channel 2 <sup>nd</sup> Request to Send	Printer	This signal is at the positive EIA level when the printer is on line and at the negative EIA level when off line. Operator can change the polarity of this signal by changing the jumper wire of the printer.
	20	Data Terminal Ready	Printer	
DC Current Loop Serial Interface	17	TTY Transmitted Data	Printer	Low impedance (marking) between pins 17 and 24 when the TTY printer is ready to accept data ; high impedance (spacing) when the TTY printer is busy. Operator can change the polarity of this signal by changing the jumper wire of the printer.
	24	TTY Transmitted Data Return		
	25 23	TTY Received Data TTY Received Data Return	Input Device	Received serial data

# Appendix L

## Data Transmission Sequence



Parallel Interface Timing



Serial Interface Timing



# Index

Subject	Page
APPLE, use with	87
American character-set	75
ASCII	23, 79, 80
differences	42, 43, 79, 80
Aspect ratio	45
Bell code	35
Block diagram	102
Cable, Printer	15
CHRS	23, 34, 35
Compressed character printing	28, 29, 34, 36
Control Codes	26, 27, 34, 81, 82
Data Transmission Sequence	106
Descender	50
Dot matrix	47
Double strike printing	28, 34, 37
Double width printing	27, 29, 34, 36
Emphasized printing	38, 39, 40
English character-set	74
Escape codes	28, 34
Form Feed (FF) button	19, 20
Form Feed command	57, 59
French character-set	73
Function codes	101
German character-set	74
Graphics characters	31, 32, 41, 42, 48, 63, 97, 98
Horizontal Tabs	59, 60, 61
Japanese character-set	71, 72
Line Feed (LF) button	19, 20
Line Feed command	57, 58
Line spacing	50, 51, 52, 55
LLIST	26
Logo design, sample	64
Lower case modification	49
LPRINT	25, 26, 34
LPRINT TAB	26
Page length	55
Paper, feeding	16, 21
Paper thickness adjustment	18
Pinout Signals	104
Rack, paper	16
Ribbon installation	11
Screen dump	44, 45
Shipping screws	9
Specifications	99
STRING\$	30
Switches, graphics/character selection	14, 71, 83
Underlining	54
Vertical Tabs	62
Word processing	40

---

# NOTICE

Please send your comments, suggestions, and any errors you might find to:

The EDITOR  
MX-80 Manual  
Box 19669  
San Diego, CA 92119  
U.S.A.

*Thank You.*

## LIMITED WARRANTY

EPSON AMERICA, INC. (hereinafter referred to as "Epson"), warrants this product against defects in material and workmanship for a period of ninety (90) days from the original date of purchase to the original purchaser only.

During the warranty period, Epson will repair (or, at its option, replace), at no charge, components that prove to be defective, provided the product is returned (shipping prepaid and properly packed) to the nearest authorized Epson Service Center or Epson at the address printed below.

You may request information on how to obtain service by contacting the Epson dealer from whom the product was purchased, or by contacting Epson directly at the address printed below.

Your sales receipt is your warranty validation. Dated proof of purchase (such as a bill of sale or cancelled check) must be provided when requesting warranty work to be performed. This warranty does not apply if, in the opinion of Epson, the product has been damaged by accident, misuse, neglect or misapplication as a result of service or modification by other than an authorized Epson Service Center.

### WARRANTY NOTICE

THIS WARRANTY IS IN LIEU OF ALL OTHER EXPRESS WARRANTIES, STATEMENTS OR REPRESENTATIONS, AND UNLESS STATED HEREIN, ALL SUCH WARRANTIES, STATEMENTS OR REPRESENTATIONS MADE BY ANY OTHER PERSON OR FIRM ARE VOID. ALL IMPLIED WARRANTIES IN CONNECTION WITH THIS SALE, INCLUDING THE WARRANTY OF MERCHANTABILITY, SHALL BE OF THE SAME DURATION AS THE WARRANTY PERIOD STATED ABOVE. SOME STATES DO NOT ALLOW LIMITATIONS ON HOW LONG AN IMPLIED WARRANTY LASTS, SO THE ABOVE LIMITATION MAY NOT APPLY TO YOU.

IN THE EVENT THAT THIS PRODUCT SHALL PROVE DEFECTIVE IN WORKMANSHIP OR MATERIALS, YOUR SOLE REMEDY SHALL BE THE REPAIR OR REMEDY AS STATED IN THIS WARRANTY, AND UNDER NO CIRCUMSTANCES SHALL EPSON BE LIABLE FOR ANY LOSS OR DAMAGE, DIRECT, INCIDENTAL OR CONSEQUENTIAL, ARISING OUT OF THE USE OF, OR INABILITY TO USE, THIS PRODUCT. SOME STATES DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THE ABOVE LIMITATION OR EXCLUSION MAY NOT APPLY TO YOU.

Epson America, Inc.  
Service Center  
23844 Hawthorne Blvd.  
Torrance, CA 90505  
Phone: (213) 378-2220