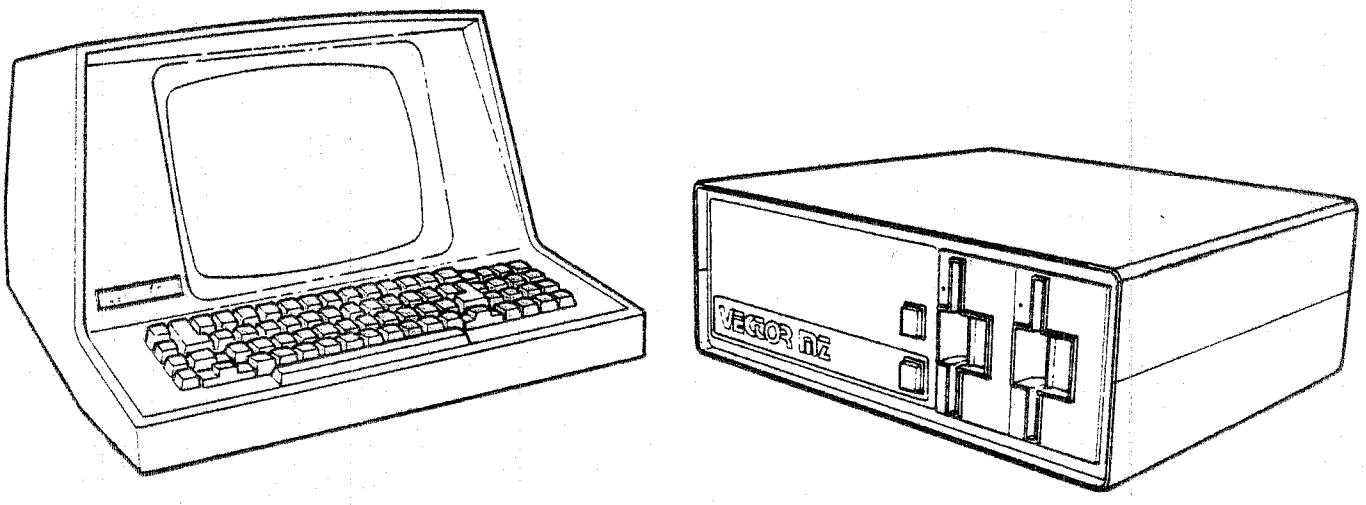
MATRIX PRINTER



VECTOR

VECTOR GRAPHIC, INC.

MP DOT MATRIX PRINTER

Revision 1

USER'S MANUAL

Revision A

March 10, 1980

Disclaimer

Vector Graphic makes no representations or warranties with respect to the contents of this manual itself, whether or not the product it describes is covered by a warranty or repair agreement. Further, Vector Graphic reserves the right to revise this publication and to make changes from time to time in the content hereof without obligation of Vector Graphic to notify any person of such revision or changes, except when an agreement to the contrary exists.

Revisions

The date and revision of each page herein appears at the bottom of each page. The revision letter such as A or B changes if the MANUAL has been improved but the PRODUCT itself has not been significantly modified. The date and revision on the Title Page corresponds to that of the page most recently revised. When the product itself is modified significantly, the product will get a new revision number, as shown on the manual's title page, and the manual will revert to revision A, as if it were treating a brand new product. EACH MANUAL SHOULD ONLY BE USED WITH THE PRODUCT IDENTIFIED ON THE TITLE PAGE.

Rev. 1-A 3/10/80

REPAIR AGREEMENT

The MP Dot Matrix Printer sold hereunder is sold "as is", with all faults and without any warranty, either expressed or implied, including any implied warranty of fitness for intended use or merchantability. However, the above notwithstanding, VECTOR GRAPHIC, INC., will, for a period of ninety (90) days following delivery to customer, repair or replace any MP Dot Matrix Printer that is found to contain defects in materials or workmanship, provided:

1. Such defect in material or workmanship existed at the time the MP Dot Matrix Printer left the VECTOR GRAPHIC, INC., factory;
2. VECTOR GRAPHIC, INC., is given notice of the precise defect claimed within ten (10) days after its discovery;
3. The MP Dot Matrix Printer is promptly returned to VECTOR GRAPHIC, INC., at customer's expense, for examination by VECTOR GRAPHIC, INC., to confirm the alleged defect, and for subsequent repair or replacement if found to be in order.

Repair, replacement or correction of any defects in material or workmanship which are discovered after expiration of the period set forth above will be performed by VECTOR GRAPHIC, INC., at Buyer's expense, provided the MP Dot Matrix Printer is returned, also at Buyer's expense, to VECTOR GRAPHIC, INC., for such repair, replacement or correction. In performing any repair, replacement or correction after expiration of the period set forth above, Buyer will be charged in addition to the cost of parts the then-current VECTOR GRAPHIC, INC., repair rate. At the present time the applicable rate is $35.00 for the first hour, and $18.00 per hour for every hour of work required thereafter. Prior to commencing any repair, replacement or correction of defects in material or workmanship discovered after expiration of the period for no-cost-to-Buyer repairs, VECTOR GRAPHIC, INC., will submit to Buyer a written estimate of the expected charges, and VECTOR GRAPHIC, INC., will not commence repair until such time as the written estimate of charges has been returned by Buyer to VECTOR GRAPHIC, INC., signed by duly authorized representative authorizing VECTOR GRAPHIC, INC., to commence with the repair work involved. VECTOR GRAPHIC, INC., shall have no obligation to repair, replace or correct any MP Dot Matrix Printer until the written estimate has been returned with approval to proceed, and VECTOR GRAPHIC, INC., may at its option also require prepayment of the estimated repair charges prior to commencing work.

Repair Agreement void if the enclosed card is not returned to VECTOR GRAPHIC, INC. within ten (10) days of end consumer purchase.

FOREWORD

Audience

This manual is intended for computer distributors, or others with at least a moderate technical knowledge of small computers.

Scope

It will describe what the Vector Graphic MP Dot Matrix Printer does in the context of a computer system, how to use the printer both in Vector Graphic and in other S-100 systems, and how the printer works.

Organization

Each section is written at a uniform level of technical depth. "Perspective" describes WHAT the printer does and requires only a moderate knowledge of computer design. "User's Guide" describes HOW to install the MP and make it work and assumes the same level of knowledge plus the ability to use a few simple tools. "Theory of Operation" discusses WHY the board works and assumes a knowledge of digital electronics and software principles.

Vector Graphic MP Dot Matrix Printer

SPECIFICATIONS

Interface	TTL level: 2 parallel output ports and 1 parallel input port.
Compatibility:	Designed for Vector Graphic systems though may be used with most Z-80 S-100 bus systems which can supply an additional 2-1/2 amps at +8VDC and -16 VDC.
I/O Ports Used	56K systems: Ports 8 and 9 (I/O II standard addressing) 48K systems: Ports 4 and 5 (I/O I standard addressing)
Software Driver	On 2708 PROM at CC00H-CFFFH for 48K systems, at EC00H-EFFFH for 56K systems.
RAM Used	Approximately 128 bytes at FC00H-FC63H
Throughput speed	70 lines per minute. 150 characters per second. Has 1 line input buffer.
Printing Method	Unidirectional 7-wire x 5 column dot matrix.
Line Spacing	6 lines per inch
Column Spacing	80 columns, 1/10 inch wide, software modifiable. 40/80 characters per line using software driver supplied with MP.
Character Size	0.122 inch by 0.083 inch.
Resolution	1/60 inch, horizontally and vertically. 60 dot positions per inch.
Copies	Original and 1 copy. Maximum paper thickness, 0.2 mm.
Form Width	4 to 10 inches.
Sprocket Pins	5/32" diameter, spaced 1/2" apart.
Paper Feed	Pin wheel actuated by pulse motor drive.
Paper Feed Speed	10 lines per second.
Paper Loading	Through rear of cabinet.
Print Head Life	100 million character expectancy at 14 dots per character.
Drive Life	5 million lines expectancy.

Vector Graphic MP Dot Matrix Printer

Ribbon	Ø.5 inch x 36 feet; standard unicolor matrix printer ribbon, black or purple. Underwood type spools.
Power Required	+8VDC @ 2-1/2 A, -16VDC @ 2-1/2 A.
Environmental	68°F to 1Ø4°F operating temperature range. 1Ø% to 8Ø% non-condensing humidity.
Dimensions	7 inches high by 18 inches wide by 13 inches deep.
Weight	21 lbs.

I. PERSPECTIVE

1.1 Compatibility

The Vector Graphic MP is a low cost, tractor feed, 80 column dot matrix printer. Though the MP was designed specifically for use with any Vector Graphic microcomputer system, it can be used with any S-100 Z-80 based system which has at least one parallel TTL input port, two parallel TTL output ports and can supply 2-1/2 amps at +8VDC and -16VDC to the printer.

1.2 Modifications to mainframe may be required.

If your printer was ordered separately and not part of a complete system, modifications may be necessary on your Vector Graphic mainframe, particularly if it was manufactured prior to 1/1/80.

Modifications which may be required include wiring a printer power outlet into the power supply, installing a cable between the interface board and the back panel and installing a PROM on the PROM/RAM board.

All parts necessary for the modifications are included with the printer.

1.3 Software Driver

The Vector Graphic MP dot matrix printer's features are controlled by a software driver resident in PROM. In addition, a small printer driver access routine is necessary. Vector Graphic CP/M 2.0 diskettes include this routine as part of the CONFIG module. Vector Graphic MDOS 8.6 diskettes will include this routine as an overlay file called SYSP. Besides printing 96 standard ASCII characters, an almost endless array of graphics characters are possible via direct assembly language software control. The PROM supplied with the system provides character generation and other necessary programming. A listing of the printer driver routine is included in the appendix of this manual. An optional MP Graphics Driver PROM provides software support for special graphics characters.

The printer driver supplied with the MP dot matrix printer will work, without modification, on Vector Graphic MZ, System B, 3030 and Memorite II microcomputers operating with the following software: MDOS, MZOS and CP/M. Simply follow the appropriate print instructions with the software to provide printed output.

While it is certainly possible to modify or rewrite the printer driver routine to operate with other systems, this is a job which should only be undertaken by someone familiar with Z-80 assembly language programming.

Note: You must have the 4.0 monitor to run the MP if using MDOS (version 8.6 or later. It will work with the 3.1 monitor if using CP/M.

1.4 Reliability and cost effectiveness of the printer.

The Vector Graphic MP Dot Matrix Printer is clearly a breakthrough in cost effectiveness. This was accomplished by driving the printer from the host computer and deriving power from the mainframe. This permits Vector Graphic to offer a compact printer which is eminently compatible with your Vector Graphic microcomputer at a truly advantageous price.

Considerable attention was given during design of the printer to the elimination of noise. It features an enclosure specially treated with sound absorbant material to reduce machine vibration and resultant noise.

In addition to the above features, the Vector Graphic MP Dot Matrix Printer has proven to be remarkably reliable. The life expectancy of the printer drive, for example, is estimated to be 5 million lines. Print head life expectancy is estimated to be 100 million characters.

The MP Dot Matrix Printer comes complete with all necessary cables, firmware, installation instructions and documentation including a program which prints high-resolution memory mapped images. The unit is completely factory assembled and tested.

II. USER'S GUIDE

In order to understand the User's Guide, you must be aware that the MP dot matrix printer was designed to work as part of a Z-8Ø based system, it is not intended to be a free standing unit. It cannot handle RS-232 or current loop signals directly, for example. By taking this approach and exploiting the inherent capabilities of the Z-8Ø, however, Vector Graphic, Inc. is able to offer a dot matrix printer that can deliver exceptional performance to your system at low cost.

2.1 Modifications to the Vector Graphic Microcomputer

This section describes the modifications which must be made to the mainframe's power supply to accomodate the printer's power needs. If you are going to use the printer in an existing system that was not shipped with an MP printer, THIS SECTION IS VERY IMPORTANT.

However, if the printer was shipped as a part of a complete Vector Graphic computer system, aside from the section on plugging in the interconnecting cables, you may skip directly to the test section.

There are four cable assemblies, a small adapter printed circuit board and and a ROM supplied with the MP dot matrix printer. Two of the cables are used to connect from sources inside the computer to the rear panel of the computer. Two are used to connect from the back panel of the computer to the MP dot matrix printer. The small adaptor board is supplied for use with the Bitstreamer II I/O board. The ROM is installed on the PROM/RAM board.

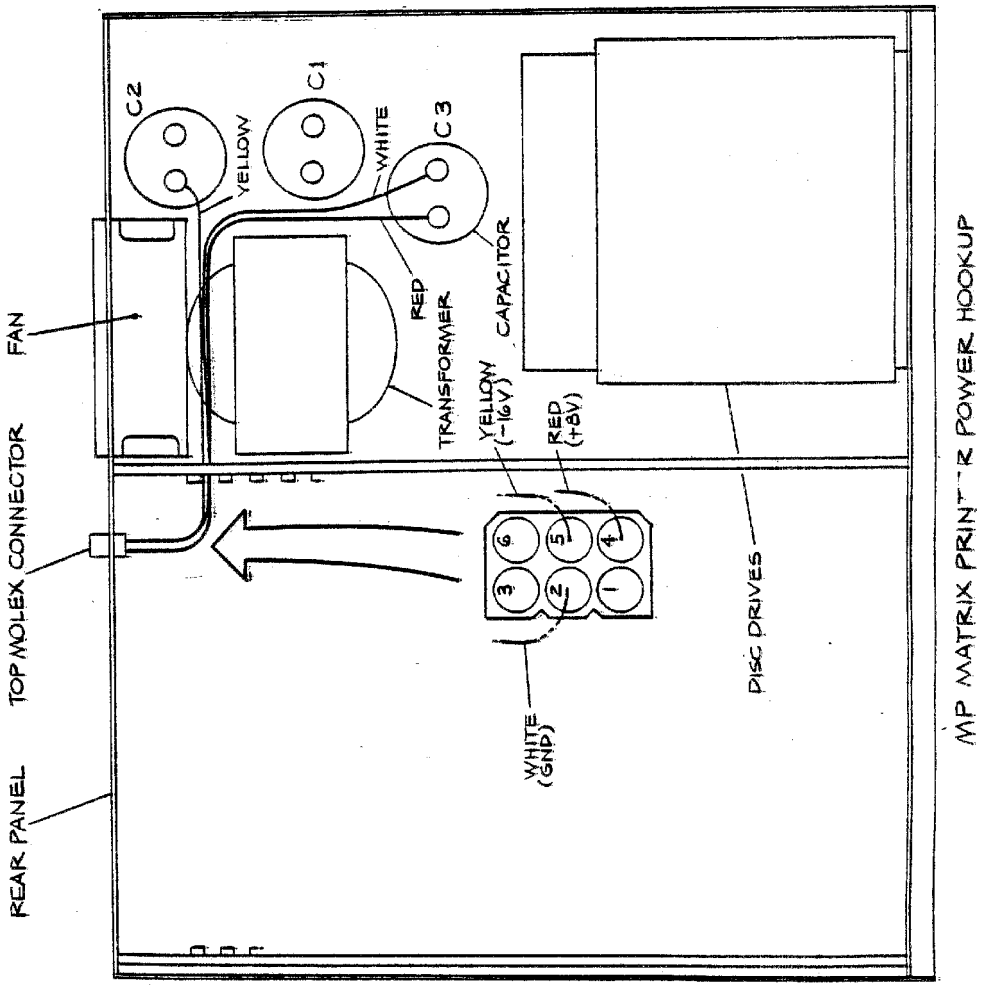
1) The three wire cable with a Molex-type plug on one end and spade lugs on the other ends is used to make power available from the internal power supply in the mainframe to the rear panel of the computer.

2) A 25 conductor flat cable which terminates in a DB-25S female connector on one end and a 24 pin DIP connector on the other is used to connect the Bitstreamer I/O board to the back panel of the computer.

3) The cable with a female Molex-type connectors on one end and a male on the other is used to connect the power supply in the mainframe to the MP.

4) The flat cable which terminates in a male DB-25S connectors on one end and a female DB-25S connector on the other is used to provide signals from the mainframe to the MP.

5) The adaptor board with two 34 pin sockets on one side and a 24 pin DIP socket on the other is used to connect the Bitstreamer II I/O board with the second cable listed above.



MP MATRIX PRINTER POWER HOOKUP

2.1.1 Power Supply Modifications

In order to run the MP dot matrix printer, a power supply cable must be wired into the mainframe power supply and then fastened to the rear of the computer. A separate cable then connects the mainframe to the printer.

1) Unplug the microcomputer from the outlet and disconnect the power supply cord from the back of the machine.

2) Wait at least one minute.

3) Unscrew the 4 screws which hold the cover to the frame using a Phillip's screwdriver. Remove the cover. The power supply components are on the right side of the computer and consist of a transformer, three electrolytic capacitors, two stud-type diodes and a bridge rectifier.

4) Make sure at least one minute has elapsed between the time you have disconnected the power and have taken the cover off.

5) Thread the three wires of the power cable, lug ends first, through a cut out in the back of the machine. If you are interfacing the MP to an S-100 computer not manufactured by Vector Graphic, Inc. see the note at the end of this section.

6) Identify capacitor C2. It is one of the two 28,000 mfd. capacitors and has white wires going to one terminal and yellow wires going to the other. Loosen the minus terminal on this capacitor (the one with the yellow wires attached to it) and fasten the yellow wire of the MP power cable to it using the spade lug attached. Tighten the terminal.

7) Identify capacitor C3. It is the 60,000 mfd. capacitor. The plus terminal of C3 has red wires attached to it, the minus terminal of C3 has white wires going to it. Loosen both these terminals and fasten the remaining two wires of the MP power cable to them: red to red, white to white. Tighten the terminal. This completes the modifications to the power supply.

8) You may test the voltages at the connector. Pin 2 is ground. Pin 1 should be approximately +8VDC. Pin 3 should be approximately -16VDC.

9) If you have an appropriate cut out in the rear of your mainframe, you may permanently fasten the connector to the rear panel. If you do not already have such a hole in the back panel of your computer, you have two options: you may leave the connector hanging loose out of the back or you may cut a hole in the back panel yourself. The dimensions of the hole required are .600 inch by .725 inch. Label this connector "PRINTER POWER CONNECTOR."

Note:

If you are interfacing the Vector Graphic MP dot matrix printer to an S-100 computer which was not manufactured by Vector Graphic, Inc., connect the yellow wire of the power cable assembly to your -16V source, the red wire to your +8V source and the white wire to ground.

2.1.2 Bitstreamer I Board Modifications

In order to interface a Vector Graphic MP dot matrix printer to an S-100 system via the Vector Graphic Bitstreamer I board, two printed circuit traces must be cut and two jumpers must be added as per the following. The Bitstreamer I board may be identified by the single DIP switch on the upper left hand side of the board. The Bitstreamer II board has three DIP switches in the same position. If you are interfacing the MP dot matrix printer to a Vector Graphic microcomputer which is equipped with a Bitstreamer II I/O board, go to the next section.

1) On the component side of the board, there is a trace which begins at pin 10 of U17, runs underneath and exits from beneath the chip between pins 6 and 7. Cut this trace.

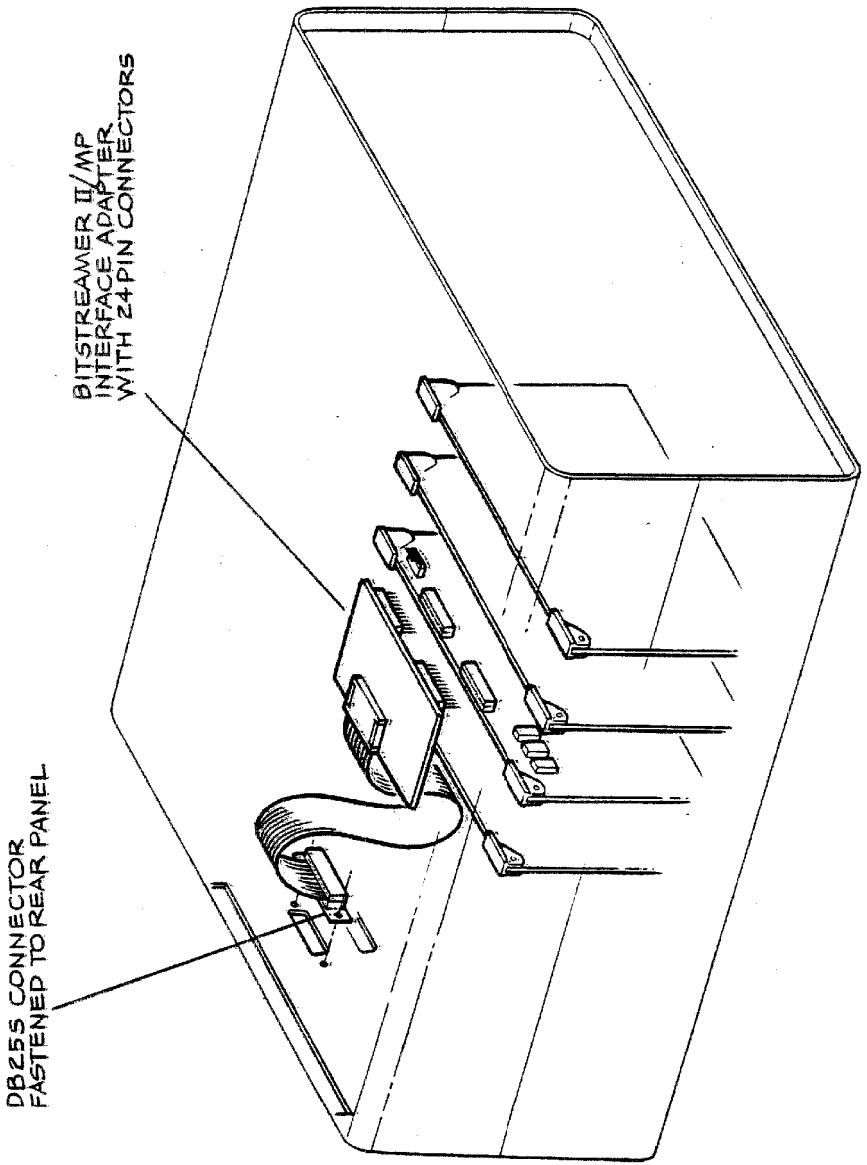
2) On the circuit side of the board, there is a trace which runs to pad 8 of J3. Cut this trace at the pad.

3) On the circuit side of the board, solder a jumper from pad 8 of J3 to pad 20 of J2.

4) On the circuit side of the board, solder a jumper from pad 17 of J3 to pad 19 of J2.

5) Plug in the cable with the 24 pin DIP connector to J3 of the Bitstreamer I board. Fasten the other end of the cable assembly (terminating in an DB-25S connector) to the back panel through one of the cut outs provided for this purpose. Label this connector "PRINTER SIGNAL CONNECTOR."

Note: Make sure that the original port addresses have not been changed. On a 48K system using an I/O I board, the parallel port addresses should be set at 4 and 5 (factory standard.)



BITSTREAMER II/MP INTERFACE

2.1.3 Bitstreamer II Board Modifications

If you are interfacing the Vector Graphic MP dot matrix printer with the Bitstreamer II board, use the following procedure.

1) An adapter board has been supplied with the MP. Connect the adapter board to the two 34 pin connectors on top of the Bitstreamer II board, orienting the board so that the 24 pin DIP socket is towards the rear panel.

2) Take the cable which has a 24 pin DIP plug on one end and an DB-25S female connector attached to the other end and plug the DIP plug into the DIP socket on the adapter board. Make sure the notches on the socket and the plug are on the same end.

3) Attach the connector on the other end of the cable to one of the blank cut outs at the back panel which has been provided for this purpose. Label this connector "PRINTER SIGNAL CONNECTOR."

SIGNAL CABLE PIN OUT DIAGRAM

BITSTREAMER II SOCKET	DB-25 CONNECTOR	24 PIN DIP SOCKET	PORT	DESCRIPTION
J4- 2*	18	5	AØØ	Step enable
J4- 3	19	6	AO1	Step phase A
J4- 4	22	9	AO2	Step phase B
J4- 5	23	1Ø	AO3	Motor on
J4-11	8	17	AIØ	Timing
J4-12	21	8	AI1	Home
J5- 2	17	4	BOØ	Wire 1 (top)
J5- 3	16	3	BO1	Wire 2
J5- 4	15	2	BO2	Wire 3
J5- 5	14	1	BO3	Wire 4
J5- 6	24	11	BO4	Wire 5
J5- 7	25	12	BO5	Wire 6
J5- 8	12	13	BO6	Wire 7 (bottom)
J5- 1	11	14	BO7	Strobe

* J4 is the left hand socket on top of the I/O II board.

Note: Make sure the original port addresses have not been changed. On a 56K system using the I/O II board, the parallel port addresses should be set at 8 and 9 (factory standard.)

2.1.4 Installing the PROM on the PROM-RAM board

The following instructions tell you how to install the Printer Driver PROM on the 12K Vector Graphic PROM-RAM board.

1) You may identify the PROM-RAM board in your computer by the row of 8 24-pin DIP sockets at the top of the board. With the power off, remove this card from your machine. Block B must be addressed at E000H for 56K systems; C000H for 48K systems. Check your PROM/RAM board manual for details.

2) Install the printer driver PROM in socket 11. Make sure the notch on the PROM matches the indicated notch on the board.

3) Plug the PROM-RAM back into its socket on the motherboard.

4) The PROM installation is now complete.

2.2 Cable Hook-Up

To connect the cables between the computer and the printer, do the following:

1) With the power at the mainframe turned off, connect the printer power cable (Molex connectors at both ends) to the printer power connector on the back panel of the mainframe and the matching connector at the back of the MP.

2) With the power at the mainframe still turned off, connect the printer signal cable (the flat wide cable with a 25 pin connector at each end) from the printer signal connector on the back panel of the mainframe to the matching connector at the back of the MP. The cable hook-up procedure is now complete.

2.3 Loading Paper

To load paper into the Vector Graphic MP dot matrix printer:

1) Remove the clear plastic top from the printer.

2) Tilt tractor feed mechanism forward until it stops.

3) Place box of fan folded paper behind printer.

4) Feed paper carefully into paper guide inlet at bottom of printer until it feeds up between platen and print head. Open tractor guides. Pull paper up through tractor assemblies, aligning guide holes in paper with tractor feed pins. It would be convenient, at this time, to position the top of the next form just above the print line. Make sure that both left and right sides are aligned correctly, otherwise the paper will bind. One easy way to do this is to count the number of holes from the top of form to the first (top) pins on either tractor. They should be the same. Now close the tractor

guides.

5) Replace the plastic cover on the printer. The paper feeding process is now complete.

TABLE OF CONTENTS

Section — Page

IV. Appendix

2.4 Initial Testing

After the cables have been hooked up and paper has been installed in the MP, some initial testing can be done to assure that everything is working properly. Two test procedures are provided, one for Vector Graphic systems under MDOS and one for Vector Graphic systems under CP/M.

NOTE:
When used with the printer driver furnished, the printer will not print a line until a Carriage Return has been received unless the automatic CRLF function has previously been selected. The characters received are stored in a buffer until a Carriage Return causes the program to send an entire line to the printer.

2.4.1 Testing the MP and Printer Driver under MDOS

To test the MP and printer driver software in a Vector Graphic system using MDOS:

1) Boot up MDOS using the normal procedure.

2) Load the printer driver call routine from the MDOS (8.6 version or later) system diskette by typing SYSP (return).

3) Assign the port used by the printer by typing ASSIGN 2,3 (return).

4) Type FILES (return). A list of the files present on the disk should now be printed on the MP dot matrix printer.

2.4.2 Testing the MP and Printer Driver under CP/M

To test the MP and printer driver software in a Vector Graphic system using CP/M.

1) Boot up CP/M using the normal procedure.

2) Type CONFIG (return). The program will prompt you on various system choices. In response to the question about printers, type D. By typing YES to the question about making the selection permanent, the system will configure itself to work with the MP printer every time it is booted up.

3) Type ^P. This causes the MP to print all data input from the keyboard. Type a few words of your choice to confirm that the printer is operating properly. By typing DIR (return), a list of the files on your diskette will be printed. Typing ^P will toggle the print function on and off.

2.5 Printer Control Commands

The following commands cause the MP to perform the listed functions provided the system has been initialized as explained in the following two sections. These commands will work from the keyboard or they can be sent under program control to perform various print functions.

1) Tab: type ^I or the (tab) key.

2) Line feed: type ^J or (lf) key.

3) Form feed: type ^L.

4) Carriage return: type ^M or (return) key.

5) Toggle auto. CRLF flag: type (ESC) and A.

6) Toggle character/graphic flag: (ESC) and G.
(Works only if MP Graphic PROM is present.)

7) Toggle 80/40 flag: (ESC) and N.

8) Set top of form: (ESC) and T.

9) Set form length: (ESC) and Fxx. xx designates form length in 1/6 inch increments. For values above 99 lines use A, B and C to represent 10, 11 and 12. These values are valid only for the tens place, they are not valid for the units position. To set a form length of 11", xx=66; 14", xx=84.

2.5.1 Printer Control Commands-MDOS

To use the MP under MDOS, initialize the system as follows.

1) Insert the diskette (MDOS 8.6 or later) and boot up the system by typing B.

2) Type SYSP (return).

3) To turn printer on type ASSIGN 2,3 (return).

4) To turn printer off type ASSIGN 2,2 (return)

NOTE:
Form length may also be set under MDOS (versions 8.6 and later) by typing SETFORM "N" (return) while in the operating system. N may be any number from .5 to 21.5 in .5 (inch) increments.

2.5.2 Printer Control Commands-CP/M

1) The first time the printer is used with a particular CP/M diskette, the

CONFIG routine must be run. After this has been done once, it does not have to be repeated provided that the same diskette is used each time.

2) To turn the printer on under CP/M type ^P.

3) To turn the printer off type ^P.

Note: To use any of the commands which use the (ESC) key, touch the key lightly, key in the next letter(s) and depress (return).

2.5.3 Printer Control Commands-Basic

The printer can be controlled from Micropolis Basic by doing the following:

In the immediate mode, the command LISTP can be used to output a program listing to the printer.

During program execution, output may be printed by writing to a print file. For example:

```
10 OPEN 1"*P"
20 PUT 1,X;
30 CLOSE 1
```

The printer can be controlled from Microsoft Basic by doing the following:

In the immediate mode, the command LLIST will output a program listing to the printer.

During program execution, output may be printed by simply using the LPRINT or the LPRINT USING commands.

2.6 Maintenance and Repair of the printer

In order to assure satisfactory printer performance, it is recommended that repairs and overhauls should be done by the Vector Graphic Dealer. Normal maintenance, however, such as changing ribbons and periodic lubrication, may be done by a person with average mechanical skill by following the instructions which follow.

2.6.1 Changing Ribbons

Note:

Replace ribbon only with a type intended for use with dot matrix printers. DO NOT use an ordinary typewriter ribbon, even for "emergency use". To do so will result in poor print quality and shortened print head life.

1) Remove both spools and the ribbon in place if there is one. Pay attention to how it is threaded.

2) Remove the new ribbon and spools from their package.

3) Unwind approximately 24 inches of ribbon from the feed spool.

4) Place the feed spool on the left spindle and the take-up spool on the right spindle.

5) Thread the ribbon from the feed spool around the rollers, reverse control levers, and frame sides.

6) Tighten the ribbon by manually turning one of the spools. The ribbon installation is now complete.

2.6.2 Periodic Lubrication

In order to insure proper operation, certain points of the MP must be lubricated at specific intervals. Three different lubricants are required:

Code	Description
O2	Light Machine Oil
G2	Light Grease
G11	Light Moly Grease

NOTE:
It is strongly suggested that in order to maximize printer life only lubricants purchased from Vector Graphic be used. Any substitution will result in shortened printer life.

First interval lubrication

The points illustrated in the appendix under "Lubrication Points-Printer" must be lubricated using the lubricants specified on the diagram at least once every 6 months or 1 million lines of use, whichever comes first.

Second interval lubrication

The points illustrated in the appendix under "Lubrication Points-Ribbon Mechanism" MP ribbon mechanism must be lubricated with G2 lubricant every 2-1/2 million lines or every six months, whichever comes first.

Third interval lubrication

It is suggested that your MP dot matrix printer be overhauled by your Vector Graphic Dealer every 5 million lines of use to assure dependable long life.

2.7 Special Graphics Characters

The optional MP Graphics Driver Prom is available from Vector Graphic for those who desire to create special graphics characters. In order to understand how to create special graphics characters, it is necessary to know how "normal" characters are generated and printed. The same principle is used whether the output is a printed letter on a page or a number on a video display. What happens is this: in response to a command to print an ASCII character, the operating system or high level language consults a look up table already in computer memory. It obtains from this table the column codes of the letter it is going to print. There are as many column codes as the character matrix is wide. For example, for the MP the number of column codes required for each character is 5. The column codes are eight bit binary numbers, generally expressed as two hexadecimal digits. For example, in response to a command to print the ASCII letter "R", the system would find in the look up table the following five column codes: 11111110, 00010010, 00110010, 01010010, 10001100. (In hex, FE, 12, 32, 52 and 8C.) You will notice that since the matrix is 5X7, the least significant bit is not used and is always 0. Thus, the numbers are always even. These binary numbers are plotted on a 5X7 grid, starting with the first (most significant) bit of the first number which is plotted as the first dot or non-dot at the bottom-most point of the first column. You then plot up the column, converting each one to a dot and each 0 to a space (non-dot). When you reach the eighth (least significant) bit, you ignore it and start on the leftmost bit of the next column code continuing in this manner until the fifth column code is completed.

Example

Column	Number	1	2	3	4	5	
	Code	0	0	0	0	0	Least significant bit
		1	1	1	1	0	
		1	0	0	0	1	
		1	0	0	0	1	
		1	1	1	1	0	
		1	0	1	0	0	
		1	0	0	1	0	
		1	0	0	0	1	Most significant bit

ASCII "R"

	Binary	Hex
Column 1	11111110	FE
Column 2	00010010	12
Column 3	00110010	32
Column 4	01010010	52
Column 5	10001100	8C

To create special graphics characters, first sketch the desired character on a 5x7 grid (graph paper would be useful for this.) Rotate the paper 90 degrees to the right. From left to right, change each dot to a 1, each non dot to a Ø. Add an extra Ø at the end of each 7 bit number to change it into an eight bit number. Convert that eight bit number into two hexadecimal digits. Do this for each of the five columns needed for each character. Store the 5 pairs of hex digits created in this manner in a table at a convenient area in memory. The table thus created can consist of up to 95 characters, each composed of 5 column codes. Make sure that the space used for the table does not conflict with the operating system or other higher level software you may be using in conjunction with the printer. If you wish to save the graphic character data you have just created, it is suggested that you do so at this point. The printer driver must be told where to find the table of graphics characters, so store the address of the first hex pair at FC5Ø/1H.

The printer hardware strobes the character code to the print wires when the most significant bit goes low. This is taken care of in the printer driver furnished by a 'RRC' instruction. If you are writing your own printer driver, be sure to include an 'RRC' instruction on the hex code before it is transmitted to the printer.

III. THEORY OF OPERATION

3.1 Print wire firing

The firing of the 7 print wires are controlled by the 8 bit output of port B. The signal from output bits 1-7 (the character column bits) are each presented to an input on a 7426 NAND gate (U3 and U8) and held there for a predetermined time by the software. Simultaneously, the 8th (most significant) bit of Port B provides the strobe pulse needed to activate the pulse width timer.

The pulse width timer (V1) is used to regulate the timing of the print wire solenoid actuation. This is particularly important since print quality is dependent upon a precise amount of energy being applied to the solenoids. The strobe pulse from the 8th bit triggers the 555 timer to begin timing, holding the output high. Capacitor C1 begins charging. If the voltage present is slightly higher than normal, it will charge quickly and then turn off the timer. Conversely, if the voltage is lower than normal, C1 will take longer to charge and the output of the timer will be held up longer. Thus this timer is able to compensate for differences in supply voltage, ensuring that the energy supplied to the print wire solenoids remains constant no matter what the variation. Transistor Q1 is used to convert the signal output of analog IC V1 to digital IC voltage levels while inverting it at the same time. This signal is again inverted through inverter on U5 and is presented to one of the inputs of the AND gate U2.

Due to mechanical considerations, two conditions have to be met before the print wires can fire. The print head has to be off the home position and dot (column) timing has to be received.

This is taken care of by IC U4. A signal indicating that the print head is off the home position is received from the printer mechanism via J2-13. This clears the 7474 flip-flop. The dot timing from the printer mechanism clocks the Q output of the flip flop. The Q signal is ANDed at U2 with the pulse from the pulse width timing circuit discussed above. This resultant signal is logically NANDed with the character column bits. If both inputs are high, the NAND gate pulls the print wire solenoid transistor base low, permitting current to flow through the solenoid thereby firing print wire. Diode, resistor and capacitor wired in series/parallel to the solenoid are for arc suppression and current limitation.

3.2 Line feed control

The tractor feed mechanism is driven by a stepper motor which is under software control via the MP interface. A stepper motor functions when its coils are energized in a predetermined sequence. This pulls the armature around in a very exact and precise manner.

The stepper motor in the MP's tractor feed mechanism functions by the printer driver software sending 2 sequences of signals on two output lines of Port A. As the signal is received from the the output line it is split between a buffer and an inverter. If the first bit received is a Ø, this goes through the buffer portion of the circuit, causing the base of the transistor to go low, switching it on and causing current to flow through the stepper motor coil. The same signal, going through the inverter causes the base of the transistor to go high, blocking current flow through the emitter-collector circuit. The diodes in the circuit prevent current surge when the coil is turned off.

The sequence of bits sent to the first line of Port A is Ø11Ø. The sequence sent to the second line is ØØ11. The coils are energized in the following order: 1 & 4, 1 & 3, 2 & 3, 2 & 4, 1 & 4, etc. The printer driver program determines how many cycles are needed and sends the appropriate number of cycles. Twenty four cycles or steps are needed to complete one line feed.

IV. APPENDIX

4.1 Function of Optional Graphics Driver PROM

The MP Graphics Driver is a program which allows you to use the Vector MP Dot Matrix Printer to print images created on the high resolution graphics board. It can also be used to print patterns specified by 1 byte hex codes.

The MP Graphics Driver program is supplied on a 2708 EPROM which fits into slot 10 on the Vector Graphic PROM/RAM board and is addressed at C800 for 48K systems and at E800 for 56K systems. See section 2.1.4 in this manual for EPROM installation instructions.

4.2 Graphics Driver Commands

Though the Graphics Driver is independent of the MP printer driver, it shares several commands with it. In addition, there is a command (ESC) G which allows the user to toggle back and forth between the printer driver and the graphics driver. When going from the printer driver to the graphics driver, (ESC) G must be followed by the hex address of a 480 character buffer in user memory. The address must be expressed in Intel format, that is, the first and second pairs of hex address digits must be reversed. For example, A000H would be entered as 00A0H.

The control commands shared between the printer and graphics driver are:

^I or (TAB)	=	Tab
^J or (LF)	=	Linefeed
^L	=	Formfeed
^M or (RETURN)	=	Carriage Return

4.3 Printing from the Monitor

There are three commands which will cause the MP to print what is displayed on the high resolution graphics board monitor. In the following list, ADDR stands for the address of the High Resolution Graphics board expressed in the Intel format, as explained above.

(ESC) D ADDR - Will cause the MP to print the screen in digital mode, bit for bit.

(ESC) R ADDR - Causes the MP to print the screen in reversed digital mode, that is, all 0's are printed as 1's and vice versa.

(ESC) V ADDR - Prints the grey scale image of what is on the screen, if the High Resolution Graphics Board is in the grey scale mode. If the board is set for the digital mode, the program will attempt to combine the digital bits together as if they were hex bytes and produce unpredictable results.

4.4 Other Print Functions

You can set the graphics driver to generate an automatic carriage return-line feed (CRLF) every time it has received 480 hex bytes sent to it. To do this, type or have the program send an (ESC) A. To toggle the function off, type or have the program send an (ESC) A again.

The graphics driver will also permit the user to print images from hex bytes stored in a buffer area in user memory. The hex bytes must be between the values of 80 and FF. The driver can be set in the 4 or 7 wire print mode by keying or having the program send an (ESC) M.

The user then sends the data to the program which then prints it when it receives a carriage return. If more than 480 bytes have been sent to the program without a carriage return, all data after the 480th byte will be lost.

To transmit a particular line, from program memory, for instance, the user would have to encode the 480 desired bytes and store it in program memory. He would further have to write a simple assembly language program which would fetch each of the successive bytes from program memory, move it to the 'C' register, push the contents of all registers onto the stack and call C803H for 48K systems and E803H for 56K systems. The program stores each of the bytes in its internal buffer and will print what has been received whenever it receives the hex code for a carriage return unless it is in the automatic CRLF mode in which case it will automatically print the line whenever 480 characters have been received.

The amount of space between the lines can also be specified using the (ESC) N command. This must be followed immediately by a 1 byte (2 hex digits) from 00-FFH. This controls the number of pulses sent to the stepping motor which pulls the paper through the tractors. Hex values of 0A and 11 are suggested as starting points for the 4 and 7 wire mode, respectively. Once this value has been set, the CRLF code will step the same number of pulses until it has been reset or a 4/7 wire switch is performed.

```
0000                        TITLE   DOT MATRIX PRINTER DRIVER   VS1.3
0000            *                                       2/6/80
0000            *
0000            ********************************************************
0000            *                                                      *
0000            *                  VECTOR  GRAPHIC                     *
0000            *              DOT MATRIX PRINTER DRIVER               *
0000            *                                                      *
0000            *                    VERSION 1.3                       *
0000            *                                                      *
0000            ********************************************************
0000            *
0000            *
0000            *                       USER NOTES
0000            *
0000            *         THE VECTOR GRAPHIC DOT MATRIX DRIVER PROGRAM
0000            *         SUPPORTS TWO MODES OF OPERATION.
0000            *         1.) 80 CHARACTERS/LINE (REGULAR PRINT)
0000            *         2.) 40 CHARACTERS/LINE (EXPANDED PRINT)
0000            *
0000            *         THESE MODES ARE SELECTABLE VIA ESCAPE SEQUENCES
0000            *       (SEE 'ESCAPE CONTROLS' BELOW). IN ORDER TO USE
0000            *       THESE PROPERLY, AND TO UNDERSTAND THE VARIOUS
0000            *       ALLOWABLE OPTIONS ALSO SUPPORTED, THE USER SHOULD
0000            *       FAMILIARIZE HIMSELF WITH THE RAM CONTROL AREA
0000            *       (FC00 TO FC5E) USED BY THE DRIVER.
0000            *
0000            *       CHARACTER BUFFER: THE AREA FROM FC00 TO FC4F IS
0000            *       RESERVED FOR HOLDING AN INPUT LINE UNTIL READY TO
0000            *       PRINT.
0000            *
0000            *       USER SELECTABLE FONTS: BY PROVIDING AN ADDRESS AT
0000            *       FC50/1, CUSTOM FONT TABLES MAY BE SELECTED. THE TABLE
0000            *       SHOULD CONSIST OF 95 CHARACTERS, EACH COMPOSED OF 5
0000            *       COLUMN CODES. AN EXAMPLE OF THE WAY COLUMN CODES MAY
0000            *       BE CALCULATED IS SHOWN HERE. (EXM. = 'A')
0000            *
0000            *             C1 C2 C3 C4 C5
0000            *              0  0  0  0  0
0000            *              0  0  1  0  0
0000            *              0  1  0  1  0
0000            *              1  0  0  0  1
0000            *              1  0  0  0  1
0000            *              1  1  1  1  1
0000            *              1  0  0  0  1
0000            *              1  0  0  0  1
0000            *
0000            *       CODE: F8 24 22 24 F8
0000            *
0000            *       THIS DRIVER PROGRAM USES THE FOLLOWING REGISTERS:
0000            *          A,B,C,D,E,H,L & IX.
0000            *       IF ANY VALUES STORED IN THESE REGISTERS ARE NECESSARY
0000            *       FOR CONTINUED OPERATION OF THE CALLING PROGRAM, THEN
0000            *       THEY MUST BE SAVED BY THE CALLING PROGRAM PRIOR TO
0000            *       CALLING THE DRIVER. THIS PROGRAM EXPECTS THE OUTPUT
0000            *       CHARACTER TO BE IN REGISTER 'C'.
0000            *
```

```
0000            *     AN EXAMPLE OF A TYPICAL CALLING SUBROUTINE IS:
0000            *     PRINT     PUSH IX    ; save registers
0000            *               PUSH H
0000            *               PUSH D
0000            *               PUSH B
0000            *               CALL ØECØ6 ; check printer status
0000            *               ORA  A
0000            *               CZ   ØECØØ ; initialize if not ready
0000            *               POP  B
0000            *               PUSH B
0000            *               MOV  C,B   ; put char. in reg. c
0000            *               CALL ØECØ3 ; send data to driver
0000            *               POP  B     ; restore registers
0000            *               POP  D
0000            *               POP  H
0000            *               POP  IX
0000            *               RET
0000            *
0000            *
0000            *
0000            *
0000            *             PROGRAM EQUATES
0000            *
0000  ECØØ =     BASE            REQ     'BASE ADDRESS =  '
0000  ØØØ4 =     STAT            EQU     4               ; status port is #4
0000  ØØØ5 =     DATA            EQU     5               ; data port is #5
0000  FCØØ =     BUFFER          EQU     ØFCØØH          ; 8Ø character buffer
0000  E8ØØ =     GRPHADD         EQU     ØE8ØØH          ; graph prom address
0000            *
0000            *         USER RAM AREA FOR DRIVER CONTROL
0000            *
0000  FC5Ø =     DCB             EQU     BUFFER+8Ø       ; device control block start
0000  FC5Ø =     FONTADD         EQU     BUFFER+8Ø       ; address of font table
0000  FC52 =     NLINE           EQU     BUFFER+82       ; lines per page
0000  FC53 =     LFPUL           EQU     BUFFER+83       ; step pulses per linefeed
0000  FC54 =     CHARCNT         EQU     BUFFER+84       ; character counter
0000  FC55 =     LNCNT           EQU     BUFFER+85       ; line counter
0000            *
0000            *    NON-USER RAM AREA FOR DRIVER CONTROL
0000            *
0000  FC56 =     PHASE           EQU     BUFFER+86       ; stepping motor control
0000  FC57 =     STATWRD         EQU     BUFFER+87       ; driver status word
0000  FC58 =     INITFLAG        EQU     BUFFER+88       ; initialization flag
0000  FC59 =     CHARNUM         EQU     BUFFER+89       ; number of chars/line
0000  FC5A =     GRSTAT          EQU     BUFFER+9Ø       ; graph status word
0000  FC5B =     BUFPTR          EQU     BUFFER+91       ; buffer pointer
0000  FC5D =     TABCNT          EQU     BUFFER+93       ; tab counter
0000  FC5E =     TEMP            EQU     BUFFER+94       ; temporary hold
0000            *
0000            *
0000            *                 CONTROL COMMANDS
0000            *
0000            *     ^I OR TAB ..................... TAB
0000            *     ^J OR LF ...................... LINEFEED
0000            *     ^L ............................ FORMFEED
0000            *     ^M OR CR ...................... CARRIAGE RETURN
0000            *     ESC A ......................... TOGGLE AUTO FLAG
```

```
0000            *     ESC F (2 BYTES) ..... SET FORM LENGTH
0000            *     ESC G ..... GRAPHICS/PRINT TOGGLE
0000            *     ESC N ..... TOGGLE 80/40 FLAG
0000            *     ESC T ..... SET TOP OF FORM
0000            *
0000            *                 ESCAPE CONTROLS
0000            *
0000            *     "A" FOR AUTO.
0000            *     WHEN SET, AUTO. FLAG CAUSES AUTOMATIC CARRIAGE
0000            *     RETURNS WHEN BUFFER IS FULL, FOLLOWED BY AN
0000            *     AUTOMATIC LINEFEED. IF WITHIN 6 LINES OF A NEW
0000            *     PAGE, AN AUTOMATIC FORMFEED IS ALSO PERFORMED.
0000            *     WHEN NOT SET, ALL CARRIAGE RETURNS, LINEFEEDS,
0000            *     AND FORMFEEDS MUST BE SENT BY THE PROGRAM THAT
0000            *     CALLS THE PRINTER DRIVER.
0000            *
0000            *     "F" FOR FORM LENGTH.
0000            *     THIS PRINTER CONTROL ALLOWS THE USER TO SET THE
0000            *     LENGTH OF THE FORM TO BE USED. A PAGE MAY RANGE
0000            *     IN SIZE FROM 1/6" (CODE 01) TO 21.5" (CODE C9).
0000            *     THE DEFAULT SIZE IS 11". THIS CONTROL IS USED
0000            *     BY SENDING AN ESC-F SEQUENCE FOLLOWED BY A TWO
0000            *     BYTE CODE. THIS VALUE IS THE NUMBER OF 1/6"
0000            *     INCREMENTS PER PAGE AND MUST BE CODED IN A
0000            *     SPECIAL WAY. THE FIRST BYTE SENT MUST BE 0-9,
0000            *     OR A-C. THIS BYTE IS THEN DECODED AS A MULTIPLE
0000            *     OF TEN. FOR EXAMPLE; A '9' WOULD SIGNIFY 90,
0000            *     WHILE A 'C' WOULD SIGNIFY 120. THE SECOND BYTE
0000            *     MUST BE A NUMBER FROM 0 TO 9, AND IS TAKEN AT
0000            *     FACE VALUE. SO, 'C9' = 129, AND 129/6 = 21.5,
0000            *     WHICH WOULD BE THE LENGTH OF THE FORM IN
0000            *     INCHES.
0000            *
0000            *     "G" FOR GRAPHIC/PRINT SWITCH. TOGGLES THE G/P
0000            *     FLAG CAUSING THE DRIVER TO TURN CONTROL OVER
0000            *     TO THE VECTOR MATRIX GRAPH PROM IF IT IS
0000            *     CURRENTLY IN PLACE AT 0E800H TO 0EBFFH.
0000            *
0000            *     "N" FOR NUMBER OF CHARACTERS PER LINE.
0000            *     TOGGLING THE 80/40 FLAG SWITCHES THE PRINTER
0000            *     FROM 80 CHARACTERS PER LINE (REGULAR PRINT)
0000            *     TO 40 CHARACTERS PER LINE (EXPANDED PRINT),
0000            *     OR VICE VERSA. THE LINECOUNTER IS UNCHANGED.
0000            *
0000            *     "T" FOR TOP OF FORM.
0000            *     THE ESCAPE-T SEQUENCE DEFINES THE CURRENT
0000            *     LINE AS THE TOP OF THE FORM.
0000            *
0000            *
0000            *           DRIVER STATUS WORD
0000            *
0000            *     BIT 7 ..... NOT USED, ALWAYS ZERO
0000            *     BIT 6 ..... FORM LENGTH SEQUENCE FLAG
0000            *     BIT 5 ..... FORM LENGTH SEQUENCE, BYTE COUNT FLAG
0000            *     BIT 4 ..... AUTOMATIC FUNCTION FLAG
0000            *     BIT 3 ..... ESCAPE SEQUENCE FLAG
0000            *     BIT 2 ..... EMPTY BUFFER FLAG
```

```
0000            *     BIT 1 ..... G/P FLAG, GRAPHIC/PRINT SELECT
0000            *     BIT 0 ..... MODE SELECT, 80/40 CHARACTER FLAG
0000            *
0000            *     ALL BITS ARE ACTIVE HIGH
0000            *     BIT 0 ..... 80 CHARS. = 0, 40 CHARS. = 1
0000            *     BIT 1 ..... PRINTING = 0, GRAPHICS = 1
0000            *     DRIVER STATUS WORD LOCATED AT FC57H
0000            *     STATWRD POINTED TO BY HL REGISTER PAIR
0000            *
0000            *
0000            *        JUMP TABLE ENTRY POINTS
0000            *
0000            *     EC00H ..... COLD START
0000            *     EC03H ..... DATA ENTRY (DATA IN REG. C)
0000            *     EC06H ..... STATUS CHECK (RETURNS STATUS IN REG. A)
0000            *     RETURNS "FFH" IN REG. A WHEN READY FOR INPUT
0000            *     RETURNS "00H" IN REG. A WHEN PRINTER IS NOT INITIALIZED
0000            *
0000            *
0000            *
0000            *
0000                                ORG     BASE
EC00 C300EE     CSTART              JMP     INIT            ; coldstart initialization
EC03            *
EC03 C30EEC     WSTART              JMP     CTRLCK          ; warmstart data entry
EC06            *
EC06 3A58FC     STATUS              LDA     INITFLAG        ; test initflag
EC09 FEFF                           CPI     0FFH
EC0B C8                             RZ                      ; Reg. A=FFH, ready
EC0C AF                             XRA     A               ; Reg. A=00H, not ready
EC0D C9                             RET
EC0E            *
EC0E            * CHECK FOR CONTROL CHARACTERS
EC0E            *
EC0E 2157FC     CTRLCK              LXI     H,STATWRD       ; point to driver status word
EC11 DD2150FC                       LXI     X,DCB           ; point to control block
EC15 3AFFEB                         LDA     GRPHADD+3FFH    ; check to see if graph prom
EC18 FE47                           CPI     'G'             ; is in place.
EC1A 2005                           JRNZ    CTRLCK0         ; jump if not there
EC1C CB4E                           BIT     1,M             ; else test the G/P flag
EC1E C200E8                         JNZ     GRPHADD         ; and do graphics if bit=1
EC21 79         CTRLCK0             MOV     A,C
EC22 E67F                           ANI     7FH             ; clear high bit
EC24 FE7F                           CPI     7FH             ; ignore delete chars.
EC26 C8                             RZ
EC27 4F                             MOV     C,A
EC28 CB76                           BIT     6,M             ; page length sequence ?
EC2A C20AED                         JNZ     PAGESET0        ; get two bytes for page length
EC2D CB5E                           BIT     3,M             ; test escseq flag
EC2F C2F8EC                         JNZ     ESCCTRL         ; escape sequence in progress
EC32 FE1B                           CPI     1BH             ; start of escape sequence ?
EC34 2003                           JRNZ    CTRLCK1         ; jump if not
EC36 CBDE                           SET     3,M             ; escape sequence started
EC38 C9                             RET
EC39 FE09       CTRLCK1             CPI     09H             ; see if ^I or tab
EC3B CACBED                         JZ      TAB
EC3E FE0A                           CPI     0AH             ; see if ^J or lf
```

```
EC40 2844                       JRZ     JPT0
EC42 FE0C                       CPI     0CH             ; see if ^L
EC44 CA62ED                     JZ      FORMFD
EC47 FE0D                       CPI     0DH             ; see if ^M or cr
EC49 2832                       JRZ     PRNDR
EC4B FE20                       CPI     20H             ; return if unrecognized
EC4D D8                         RC                      ; ctrl. char.
EC4E            *
EC4E            * STORE DATA IN BUFFER
EC4E            *
EC4E 2802       STBUF1          JRZ     STBUF2          ; jump if space
EC50 CBD6                       SET     2,M             ; set mtbuf flag
EC52 CDC6ED     STBUF2          CALL    CHARCHK         ; buffer full ?
EC55 2005                       JRNZ    STBUF4          ; jump if not
EC57 CB66       STBUF3          BIT     4,M             ; test auto. flag
EC59 C8                         RZ                      ; no auto, ignore data overflow
EC5A 1821                       JR      PRNDR           ; automatic print,lf,ff
EC5C E5         STBUF4          PUSH    H               ; save statwrd ptr.
EC5D DD3504                     DCR     4(X)            ; step the char. cntr.
EC60 2A5BFC                     LHLD    BUFPTR
EC63 71                         MOV     M,C             ; store the data
EC64 23                         INX     H               ; step bufptr
EC65 225BFC                     SHLD    BUFPTR
EC68 E1                         POP     H               ; restore statwrd ptr.
EC69 CDC6ED                     CALL    CHARCHK         ; buffer now full ?
EC6C 28E9                       JRZ     STBUF3          ; if yes, check auto. flag
EC6E 3A5DFC     TABSTEP         LDA     TABCNT          ; get the tab counter
EC71 3D                         DCR     A               ; and step it.
EC72 325DFC                     STA     TABCNT
EC75 C0                         RNZ
EC76 3E08       TABSET          MVI     A,8             ; reset to 8
EC78 325DFC                     STA     TABCNT
EC7B AF         ZRET            XRA     A               ; return with zero set
EC7C C9                         RET
EC7D            *
EC7D            * PRINT BUFFER
EC7D            *
EC7D 1100FC     PRNDR           LXI     D,BUFFER        ; set buffer pointer
EC80 ED535BFC                   SDED    BUFPTR          ; to front of buffer
EC84 CB56                       BIT     2,M             ; test mtbuf flag
EC86 2860       JPT0            JRZ     LINECK          ; if empty, just do a lf
EC88 DD7E09                     MOV     A,9(X)          ; reset the char. cntr.
EC8B DD7704                     MOV     4(X),A
EC8E 0E00                       MVI     C,0             ; clear char. column cntr.
EC90            *
EC90            * PRE-PRINT TIMING ALIGNMENT
EC90            *
EC90 DB04       RESET           IN      STAT            ; ready printer
EC92 E602                       ANI     02H
EC94 20FA                       JRNZ    RESET           ; wait till home
EC96 CD75ED                     CALL    RUN             ; send carriage
EC99 DB04       LHFL            IN      STAT
EC9B E602                       ANI     02H
EC9D 28FA                       JRZ     LHFL            ; wait while home
EC9F            *
EC9F            * LOOK UP CODE FOR CHARACTER IN FONT TABLE
EC9F            * ADDRESS = (CHAR.*5)+(COLUMN)+(FONT TBL. BASE)-(160)
```

```
EC9F            *
EC9F E5         LOOKUP      PUSH    H           ; save statwrd pointer
ECA0 D5                     PUSH    D           ; save character count
ECA1 2A5BFC                 LHLD    BUFPTR
ECA4 5E                     MOV     E,M         ; get char. from buffer
ECA5 6B                     MOV     L,E
ECA6 2600                   MVI     H,0
ECA8 54                     MOV     D,H
ECA9 29                     DAD     H           ; char. * 2
ECAA 29                     DAD     H           ; char. * 4
ECAB 19                     DAD     D           ; char. * 5
ECAC 59                     MOV     E,C
ECAD 19                     DAD     D           ; + column #
ECAE EB                     XCHG
ECAF 2A50FC                 LHLD    FONTADD
ECB2 19                     DAD     D           ; + font tbl. base
ECB3 1160FF                 LXI     D,0FF60H
ECB6 19                     DAD     D           ; - 160
ECB7 7E                     MOV     A,M         ; get code
ECB8 0F                     RRC                 ; fix code
ECB9 D1                     POP     D
ECBA E1                     POP     H
ECBB            *
ECBB 47                     MOV     B,A
ECBC CD4CED     PRNLP0      CALL    TIMEPLS     ; print a column
ECBF CB46                   BIT     0,M         ; test 80/40 flag
ECC1 2803                   JRZ     PRNLP1      ; jump if 80
ECC3 CD4CED                 CALL    TIMEPLS     ; 40, so print again
ECC6 0C         PRNLP1      INR     C           ; step char. col. cntr.
ECC7 79                     MOV     A,C
ECC8 FE05                   CPI     5
ECCA 38D3                   JRC     LOOKUP      ; print letter's next column
ECCC 0600                   MVI     B,0
ECCE FE06                   CPI     6
ECD0 20EA                   JRNZ    PRNLP0      ; send a space
ECD2 48                     MOV     C,B         ; clear char. col. cntr.
ECD3 E5                     PUSH    H           ; save statwrd ptr.
ECD4 2A5BFC                 LHLD    BUFPTR
ECD7 23                     INX     H           ; step the buffer pointer
ECD8 225BFC                 SHLD    BUFPTR
ECDB DD3504                 DCR     4(X)        ; step the char. cntr.
ECDE E1                     POP     H
ECDF CDC6ED                 CALL    CHARCHK     ; check the char. cntr.
ECE2 20BB                   JRNZ    LOOKUP      ; print next until done
ECE4            *
ECE4 CB66       AUTOLF      BIT     4,M         ; test auto. flag
ECE6 285D                   JRZ     JPT1        ; clear buffer if no auto.
ECE8 CD9EED     LINECK      CALL    LFCTRL      ; do one linefeed
ECEB CB66                   BIT     4,M         ; test auto. flag
ECED 2856                   JRZ     JPT1        ; clear buffer if no auto.
ECEF CDD8ED                 CALL    EOP         ; automatic formfeed needed ?
ECF2 FE07                   CPI     7           ; within 6 lines ?
ECF4 304F                   JRNC    JPT1        ; if no, go clear buffer
ECF6 186A                   JR      FORMFD      ; else do a formfeed.
ECF8            *
ECF8            * ESCAPE SEQUENCE CONTROL CHARACTER CHECK
ECF8            *
```

```
ECF8 CB9E      ESCCTRL         RES     3,M             ; reset escseq flag
ECFA FE41                      CPI     'A'             ; A ?
ECFC 2005                      JRNZ    ESC1
ECFE 7E                        MOV     A,M             ; test auto. flag
ECFF EE10                      XRI     10H             ; toggle auto. flag
ED01 77                        MOV     M,A
ED02 C9                        RET
ED03 FE46      ESC1            CPI     'F'             ; F ?
ED05 202C                      JRNZ    ESC2
ED07 CBF6                      SET     6,M             ; set form length seq.flag
ED09 C9                        RET
ED0A D630      PAGESET0        SUI     30H             ; adjust for numeral
ED0C CB6E                      BIT     5,M             ; first or second byte ?
ED0E 2011                      JRNZ    PAGESET2        ; jump if second byte
ED10 FE0A                      CPI     10              ; need alpha adjust too ?
ED12 3802                      JRC     PAGESET1        ; jump if not, else
ED14 D607                      SUI     7               ; do an alpha adjust.
ED16                                                   ; multiply high byte by 10
ED16 87        PAGESET1        ADD     A               ; A = (#)*2
ED17 4F                        MOV     C,A             ; C = (#)*2
ED18 87                        ADD     A               ; A = (#)*4
ED19 87                        ADD     A               ; A = (#)*8
ED1A 81                        ADD     C               ; A = (#)*10
ED1B DD770E                    MOV     14(X),A         ; hold in temp
ED1E CBEE                      SET     5,M             ; low byte next
ED20 C9                        RET
ED21 DD4E0E    PAGESET2        MOV     C,14(X)         ; get temp value
ED24 81                        ADD     C               ; A = lines/page
ED25 CBB6                      RES     6,M             ; reset form length flags
ED27 CBAE                      RES     5,M
ED29 FE01                      CPI     1               ; see if page size is
ED2B D8                        RC                      ; within allowable
ED2C FE82                      CPI     130             ; limits.
ED2E D0                        RNC                     ; (0<PS<130)
ED2F DD7702                    MOV     2(X),A          ; store good value at
ED32 C9                        RET                     ; NLINE and return.
ED33 FE47      ESC2            CPI     'G'             ; G ?
ED35 2003                      JRNZ    ESC3
ED37 CBCE                      SET     1,M             ; select the graph prom
ED39 C9                        RET
ED3A FE4E      ESC3            CPI     'N'             ; N ?
ED3C 2009                      JRNZ    ESC4
ED3E 7E                        MOV     A,M             ; test 80/40 flag
ED3F EE01                      XRI     01H             ; toggle 80/40 flag
ED41 77                        MOV     M,A
ED42 CDF5ED    MDCHK           CALL    MDSEL           ; select proper CHARNUM
ED45 183C      JPT1            JR      CLRBUF
ED47 FE54      ESC4            CPI     'T'             ; T ?
ED49 281F                      JRZ     CLEAR           ; set top of form
ED4B C9                        RET                     ; unrecognized escape sequence
ED4C           *
ED4C           * PRINT ONE COLUMN
ED4C           *
ED4C DB04      TIMEPLS         IN      STAT
ED4E E601                      ANI     01H             ; wait for dot timer
ED50 20FA                      JRNZ    TIMEPLS         ; to go low
ED52 DB04      TPLS            IN      STAT            ; now wait for
```

```
ED54 E601                       ANI     01H             ; rising edge then
ED56 28FA                       JRZ     TPLS            ; print a column by
ED58 78                         MOV     A,B             ; sending the data
ED59 F680                       ORI     80H             ; out while pulsing
ED5B D305                       OUT     DATA            ; strobe bit (B7).
ED5D E67F                       ANI     7FH
ED5F D305                       OUT     DATA
ED61 C9                         RET
ED62                *
ED62                * FORMFEED MODULE
ED62                *
ED62 CDD8ED         FORMFD      CALL    EOP             ; calculate # of lf's
ED65 CD9EED         TOF         CALL    LFCTRL          ; do one linefeed
ED68 20FB                       JRNZ    TOF             ; jump if not done, else ...
ED6A                *
ED6A                * CLEAR BUFFER AND FLAGS
ED6A                *
ED6A CD83ED         CLEAR       CALL    CLRBUF          ; clear buffer
ED6D 3255FC                     STA     LNCNT           ; clear linecntr.
ED70                *
ED70                * CARRIAGE HOMED ?
ED70                *
ED70 DB04           CARHOME     IN      STAT
ED72 E602                       ANI     02H
ED74 C8                         RZ                      ; return if homed, else ...
ED75                *
ED75                * SEND CARRIAGE ACROSS PAPER
ED75                *
ED75 3A56FC         RUN         LDA     PHASE           ; get phase word and
ED78 E606                       ANI     06H             ; turn on motor by setting
ED7A F609                       ORI     09H             ; motor flip-flop. save
ED7C D304                       OUT     STAT            ; phase information.
ED7E E607                       ANI     07H             ; home signal resets f.f.
ED80 D304                       OUT     STAT            ; whichs shuts off motor
ED82 C9                         RET
ED83                *
ED83                * CLEAR BUFFER&SPACES, RESET CHARCNT. AND BUFFPTR
ED83                *
ED83 E5             CLRBUF      PUSH    H               ; save statwrd ptr.
ED84 DD7E09                     MOV     A,9(X)
ED87 DD7704                     MOV     4(X),A          ; reset char.cntr.
ED8A 2100FC                     LXI     H,BUFFER
ED8D 225BFC                     SHLD    BUFFPTR         ; reset buffer pointer
ED90 0650                       MVI     B,80            ; clear 80 spaces
ED92 0E20                       MVI     C,20H
ED94 71             CLRB0       MOV     M,C             ; put into buffer
ED95 23                         INX     H
ED96 10FC                       DJNZ    CLRB0
ED98 E1                         POP     H               ; restore statwrd ptr.
ED99 CB96                       RES     2,M             ; reset mtbuf flag (empty buffer
ED9B C376EC                     JMP     TABSET          ; reset tab counter
ED9E                *
ED9E                * LINEFEED MODULE
ED9E                *
ED9E F5             LFCTRL      PUSH    PSW             ; save the linefeed cntr.
ED9F DD4E03         LINEFD      MOV     C,3(X)          ; get # of pulses
EDA2 2B                         DCX     H               ; point to phase word
```

```
EDA3 7E                     MOV     A,M             ; get phase info.
EDA4 0F                     RRC
EDA5 47                     MOV     B,A
EDA6 78         STPLP       MOV     A,B
EDA7 07                     RLC
EDA8 47                     MOV     B,A
EDA9 E606                   ANI     06H
EDAB D304                   OUT     STAT
EDAD            *
EDAD            * DELAY REQUIRED BETWEEN PULSES
EDAD            *
EDAD 118001     DELAY       LXI     D,0180H
EDB0 1B         DLAY        DCX     D
EDB1 7A                     MOV     A,D
EDB2 B3                     ORA     E
EDB3 20FB                   JRNZ    DLAY
EDB5 0D                     DCR     C
EDB6 20EE                   JRNZ    STPLP           ; all pulses sent ?
EDB8 70                     MOV     M,B             ; save current phase word
EDB9 3C                     INR     A
EDBA D304                   OUT     STAT            ; hold stepping motor
EDBC 1D         DLP         DCR     E               ; between linefeed delay
EDBD 20FD                   JRNZ    DLP             ; to allow cap. to charge
EDBF            *
EDBF            * STEP LINE COUNTER
EDBF            *
EDBF 2B                     DCX     H               ; point to line cntr.
EDC0 34                     INR     M               ; now step it.
EDC1 23                     INX     H               ; restore ptr. to statwrd
EDC2 23                     INX     H
EDC3 F1                     POP     PSW             ; restore the linefeed cntr.
EDC4 3D                     DCR     A               ; and step it.
EDC5 C9                     RET
EDC6            *
EDC6            * FULL BUFFER CHECK
EDC6            *
EDC6 DD7E04     CHARCHK     MOV     A,4(X)          ; get the char. cntr.
EDC9 B7                     ORA     A               ; zero flag set if buffer
EDCA C9                     RET                     ; is full
EDCB            *
EDCB            * TAB MODULE
EDCB            *
EDCB 3A5DFC     TAB         LDA     TABCNT          ; get tab counter
EDCE 47                     MOV     B,A
EDCF 0E20                   MVI     C,' '           ; spaces
EDD1 CD52EC     TAB1        CALL    STBUF2          ; store in buffer
EDD4 C8                     RZ                      ; return if end of line or page
EDD5 10FA                   DJNZ    TAB1
EDD7 C9                     RET
EDD8            *
EDD8            * END OF PAGE MODULE
EDD8            *
EDD8 DD7E02     EOP         MOV     A,2(X)          ; get lines/page (NLINE)
EDDB 47                     MOV     B,A
EDDC 3A55FC     EOP1        LDA     LNCNT           ; get line counter
EDDF B8                     CMP     B               ; is LNCNT < NLINE ?
EDE0 3808                   JRC     EOP2            ; jump if so.
```

```
EDE2 90                     SUB     B               ; LNCNT - NLINE
EDE3 280C                   JRZ     EOP3            ; at page top ?
EDE5 3255FC                 STA     LNCNT
EDE8 18F2                   JR      EOP1            ; check again.
EDEA 3A55FC     EOP2        LDA     LNCNT           ; get corrected line cntr.
EDED 4F                     MOV     C,A
EDEE 78                     MOV     A,B
EDEF 91                     SUB     C               ; NLINE - LNCNT
EDF0 C0                     RNZ
EDF1 E1         EOP3        POP     H               ; at top of page, destroy
EDF2 C36AED     JPT2        JMP     CLEAR           ; ret. add., clr. buffer
EDF5            *
EDF5            *    CHOOSE PROPER COLDEC (80 CHARS. OR 40 CHARS.)
EDF5            *
EDF5 3E50       MDSEL       MVI     A,80            ; set for 80
EDF7 CB46                   BIT     0,M             ; now test STATWRD
EDF9 2801                   JRZ     MDSEL10         ; jump if 80
EDFB 0F                     RRC                     ; set for 40
EDFC DD7709     MDSEL10     MOV     9(X),A
EDFF C9                     RET
EE00            *
EE00            * INITIALIZE ... SET: POINTERS, FLAGS, COUNTERS
EE00            * SELECT: ON-CHIP FONT TABLE, 80 CHAR. MODE
EE00            * NO AUTO FUNCTION, CLEAR BUFFER
EE00            *
EE00 AF         INIT        XRA     A
EE01 D305                   OUT     DATA            ; clear data latches
EE03 3C                     INR     A
EE04 D304                   OUT     STAT            ; holding current
EE06 211AEE                 LXI     H,INITTBL
EE09 1150FC                 LXI     D,DCB           ; load init. data
EE0C 010B00                 LXI     B,11            ; into ram (11 bytes)
EE0F EDB0                   LDIR
EE11 DD2150FC               LXI     X,DCB
EE15 2157FC                 LXI     H,STATWRD       ; driver status word
EE18 18D8                   JR      JPT2
EE1A            *
EE1A            * DEFAULT VALUES FOR INITIALIZATION
EE1A            *
EE1A 25EE       INITTBL     DW      FONTTBL         ; on-chip font table
EE1C 42                     DB      66              ; lines per page
EE1D 19                     DB      25              ; pulses per linefeed
EE1E 50                     DB      80              ; 80 char. mode
EE1F 00                     DB      00H             ; clear line cntr.
EE20 99                     DB      99H             ; preset phase word
EE21 00                     DB      00H             ; driver status word
EE22 FF                     DB      0FFH            ; initialization flag
EE23 50                     DB      80              ; 80 char. mode
EE24 00                     DB      00H             ; graph status word
EE25            *
EE25            * DOT LOOK UP TABLE
EE25            *
EE25 00         FONTTBL     DB      0               ; SPACE
EE26 00                     DB      0
EE27 00                     DB      0
EE28 00                     DB      0
EE29 00                     DB      0
```

```
EE2A 00                 DB      0               ; !
EE2B 00                 DB      0
EE2C BE                 DB      0BEH
EE2D 00                 DB      0
EE2E 00                 DB      0
EE2F 00                 DB      0               ; "
EE30 06                 DB      06H
EE31 00                 DB      0
EE32 06                 DB      06H
EE33 00                 DB      0
EE34 28                 DB      28H             ; #
EE35 FE                 DB      0FEH
EE36 28                 DB      28H
EE37 FE                 DB      0FEH
EE38 28                 DB      28H
EE39 48                 DB      48H             ; $
EE3A 54                 DB      54H
EE3B FE                 DB      0FEH
EE3C 54                 DB      54H
EE3D 24                 DB      24H
EE3E 46                 DB      46H             ; %
EE3F 26                 DB      26H
EE40 10                 DB      10H
EE41 C8                 DB      0C8H
EE42 C4                 DB      0C4H
EE43 6C                 DB      6CH             ; &
EE44 92                 DB      92H
EE45 AC                 DB      0ACH
EE46 40                 DB      40H
EE47 A0                 DB      0A0H
EE48 00                 DB      0               ; '
EE49 00                 DB      0
EE4A 0E                 DB      0EH
EE4B 00                 DB      0
EE4C 00                 DB      0
EE4D 00                 DB      0               ; (
EE4E 38                 DB      38H
EE4F 44                 DB      44H
EE50 82                 DB      82H
EE51 00                 DB      0
EE52 00                 DB      0               ; )
EE53 82                 DB      82H
EE54 44                 DB      44H
EE55 38                 DB      38H
EE56 00                 DB      0
EE57 54                 DB      54H             ; *
EE58 38                 DB      38H
EE59 7C                 DB      7CH
EE5A 38                 DB      38H
EE5B 54                 DB      54H
EE5C 10                 DB      10H             ; +
EE5D 10                 DB      10H
EE5E 7C                 DB      7CH
EE5F 10                 DB      10H
EE60 10                 DB      10H
EE61 00                 DB      0               ; ,
EE62 80                 DB      80H
```

```
EE63 60                         DB      60H
EE64 00                         DB      0
EE65 00                         DB      0
EE66 10                         DB      10H             ; -
EE67 10                         DB      10H
EE68 10                         DB      10H
EE69 10                         DB      10H
EE6A 10                         DB      10H
EE6B 00                         DB      0               ; .
EE6C 00                         DB      0
EE6D 80                         DB      80H
EE6E 00                         DB      0
EE6F 00                         DB      0
EE70 C0                         DB      0C0H            ; /
EE71 20                         DB      20H
EE72 10                         DB      10H
EE73 08                         DB      08H
EE74 06                         DB      06H
EE75 7C                         DB      7CH             ; 0
EE76 A2                         DB      0A2H
EE77 92                         DB      92H
EE78 8A                         DB      8AH
EE79 7C                         DB      7CH
EE7A 00                         DB      0               ; 1
EE7B 84                         DB      84H
EE7C FE                         DB      0FEH
EE7D 80                         DB      80H
EE7E 00                         DB      0
EE7F C4                         DB      0C4H            ; 2
EE80 A2                         DB      0A2H
EE81 92                         DB      92H
EE82 92                         DB      92H
EE83 8C                         DB      8CH
EE84 44                         DB      44H             ; 3
EE85 82                         DB      82H
EE86 92                         DB      92H
EE87 9A                         DB      9AH
EE88 66                         DB      66H
EE89 30                         DB      30H             ; 4
EE8A 28                         DB      28H
EE8B 24                         DB      24H
EE8C FE                         DB      0FEH
EE8D 20                         DB      20H
EE8E 4E                         DB      4EH             ; 5
EE8F 8A                         DB      8AH
EE90 8A                         DB      8AH
EE91 8A                         DB      8AH
EE92 72                         DB      72H
EE93 78                         DB      78H             ; 6
EE94 94                         DB      94H
EE95 92                         DB      92H
EE96 92                         DB      92H
EE97 62                         DB      62H
EE98 C2                         DB      0C2H            ; 7
EE99 22                         DB      22H
EE9A 12                         DB      12H
EE9B 0A                         DB      0AH
```

```
EE9C 06                 DB      06H
EE9D 6C                 DB      6CH             ; 8
EE9E 92                 DB      92H
EE9F 92                 DB      92H
EEA0 92                 DB      92H
EEA1 6C                 DB      6CH
EEA2 8C                 DB      8CH             ;9
EEA3 92                 DB      92H
EEA4 92                 DB      92H
EEA5 52                 DB      52H
EEA6 3C                 DB      3CH
EEA7 00                 DB      0               ; :
EEA8 00                 DB      0
EEA9 28                 DB      28H
EEAA 00                 DB      0
EEAB 00                 DB      0
EEAC 00                 DB      0               ; ;
EEAD 80                 DB      80H
EEAE 68                 DB      68H
EEAF 00                 DB      0
EEB0 00                 DB      0
EEB1 10                 DB      10H             ; <
EEB2 28                 DB      28H
EEB3 44                 DB      44H
EEB4 82                 DB      82H
EEB5 00                 DB      0
EEB6 28                 DB      28H             ; =
EEB7 28                 DB      28H
EEB8 28                 DB      28H
EEB9 28                 DB      28H
EEBA 28                 DB      28H
EEBB 00                 DB      0               ; >
EEBC 82                 DB      82H
EEBD 44                 DB      44H
EEBE 28                 DB      28H
EEBF 10                 DB      10H
EEC0 04                 DB      04H             ; ?
EEC1 02                 DB      02H
EEC2 B2                 DB      0B2H
EEC3 0A                 DB      0AH
EEC4 04                 DB      04H
EEC5 7C                 DB      7CH             ; @
EEC6 82                 DB      82H
EEC7 BA                 DB      0BAH
EEC8 92                 DB      92H
EEC9 8C                 DB      8CH
EECA F8                 DB      0F8H            ; A
EECB 24                 DB      24H
EECC 22                 DB      22H
EECD 24                 DB      24H
EECE F8                 DB      0F8H
EECF FE                 DB      0FEH            ; B
EED0 92                 DB      92H
EED1 92                 DB      92H
EED2 92                 DB      92H
EED3 6C                 DB      6CH
EED4 7C                 DB      7CH             ; C
```

```
EED5 82                         DB      82H
EED6 82                         DB      82H
EED7 82                         DB      82H
EED8 44                         DB      44H
EED9 FE                         DB      0FEH            ; D
EEDA 82                         DB      82H
EEDB 82                         DB      82H
EEDC 82                         DB      82H
EEDD 7C                         DB      7CH
EEDE FE                         DB      0FEH            ; E
EEDF 92                         DB      92H
EEE0 92                         DB      92H
EEE1 82                         DB      82H
EEE2 82                         DB      82H
EEE3 FE                         DB      0FEH            ; F
EEE4 12                         DB      12H
EEE5 12                         DB      12H
EEE6 02                         DB      02H
EEE7 02                         DB      02H
EEE8 7C                         DB      7CH             ; G
EEE9 82                         DB      82H
EEEA 82                         DB      82H
EEEB A2                         DB      0A2H
EEEC 64                         DB      64H
EEED FE                         DB      0FEH            ; H
EEEE 10                         DB      10H
EEEF 10                         DB      10H
EEF0 10                         DB      10H
EEF1 FE                         DB      0FEH
EEF2 00                         DB      0               ; I
EEF3 82                         DB      82H
EEF4 FE                         DB      0FEH
EEF5 82                         DB      82H
EEF6 00                         DB      0
EEF7 40                         DB      40H             ; J
EEF8 80                         DB      80H
EEF9 80                         DB      80H
EEFA 80                         DB      80H
EEFB 7E                         DB      7EH
EEFC FE                         DB      0FEH            ; K
EEFD 10                         DB      10H
EEFE 28                         DB      28H
EEFF 44                         DB      44H
EF00 82                         DB      82H
EF01 FE                         DB      0FEH            ; L
EF02 80                         DB      80H
EF03 80                         DB      80H
EF04 80                         DB      80H
EF05 80                         DB      80H
EF06 FE                         DB      0FEH            ; M
EF07 04                         DB      04H
EF08 18                         DB      18H
EF09 04                         DB      04H
EF0A FE                         DB      0FEH
EF0B FE                         DB      0FEH            ; N
EF0C 08                         DB      08H
EF0D 10                         DB      10H
```

```
EF0E 20                         DB      20H
EF0F FE                         DB      0FEH
EF10 7C                         DB      7CH             ; O
EF11 82                         DB      82H
EF12 82                         DB      82H
EF13 82                         DB      82H
EF14 7C                         DB      7CH
EF15 FE                         DB      0FEH            ; P
EF16 12                         DB      12H
EF17 12                         DB      12H
EF18 12                         DB      12H
EF19 0C                         DB      0CH
EF1A 7C                         DB      7CH             ; Q
EF1B 82                         DB      82H
EF1C A2                         DB      0A2H
EF1D 42                         DB      42H
EF1E BC                         DB      0BCH
EF1F FE                         DB      0FEH            ; R
EF20 12                         DB      12H
EF21 32                         DB      32H
EF22 52                         DB      52H
EF23 8C                         DB      8CH
EF24 44                         DB      44H             ; S
EF25 8A                         DB      8AH
EF26 92                         DB      92H
EF27 A2                         DB      0A2H
EF28 44                         DB      44H
EF29 02                         DB      02H             ; T
EF2A 02                         DB      02H
EF2B FE                         DB      0FEH
EF2C 02                         DB      02H
EF2D 02                         DB      02H
EF2E 7E                         DB      7EH             ; U
EF2F 80                         DB      80H
EF30 80                         DB      80H
EF31 80                         DB      80H
EF32 7E                         DB      7EH
EF33 0E                         DB      0EH             ; V
EF34 30                         DB      30H
EF35 C0                         DB      0C0H
EF36 30                         DB      30H
EF37 0E                         DB      0EH
EF38 FE                         DB      0FEH            ; W
EF39 40                         DB      40H
EF3A 30                         DB      30H
EF3B 40                         DB      40H
EF3C FE                         DB      0FEH
EF3D C6                         DB      0C6H            ; X
EF3E 28                         DB      28H
EF3F 10                         DB      10H
EF40 28                         DB      28H
EF41 C6                         DB      0C6H
EF42 06                         DB      06H             ; Y
EF43 08                         DB      08H
EF44 F0                         DB      0F0H
EF45 08                         DB      08H
EF46 06                         DB      06H
```

```
EF47 C2                         DB      0C2H            ; Z
EF48 A2                         DB      0A2H
EF49 92                         DB      92H
EF4A 8A                         DB      8AH
EF4B 86                         DB      86H
EF4C FE                         DB      0FEH            ; [
EF4D FE                         DB      0FEH
EF4E 82                         DB      82H
EF4F 82                         DB      82H
EF50 82                         DB      82H
EF51 06                         DB      06H             ; BACKSLASH
EF52 08                         DB      08H
EF53 10                         DB      10H
EF54 20                         DB      20H
EF55 C0                         DB      0C0H
EF56 82                         DB      82H             ; ]
EF57 82                         DB      82H
EF58 82                         DB      82H
EF59 FE                         DB      0FEH
EF5A FE                         DB      0FEH
EF5B 20                         DB      20H             ; ^
EF5C 10                         DB      10H
EF5D 08                         DB      08H
EF5E 10                         DB      10H
EF5F 20                         DB      20H
EF60 80                         DB      80H             ; UNDERLINE
EF61 80                         DB      80H
EF62 80                         DB      80H
EF63 80                         DB      80H
EF64 80                         DB      80H
EF65 00                         DB      0               ; '
EF66 02                         DB      02H
EF67 04                         DB      04H
EF68 08                         DB      08H
EF69 00                         DB      0
EF6A C0                         DB      0C0H            ; a
EF6B A8                         DB      0A8H
EF6C A8                         DB      0A8H
EF6D A8                         DB      0A8H
EF6E F0                         DB      0F0H
EF6F FE                         DB      0FEH            ; b
EF70 88                         DB      88H
EF71 88                         DB      88H
EF72 88                         DB      88H
EF73 F0                         DB      0F0H
EF74 70                         DB      70H             ; c
EF75 88                         DB      88H
EF76 88                         DB      88H
EF77 88                         DB      88H
EF78 88                         DB      88H
EF79 70                         DB      70H             ; d
EF7A 88                         DB      88H
EF7B 88                         DB      88H
EF7C 88                         DB      88H
EF7D FE                         DB      0FEH
EF7E 70                         DB      70H             ; e
EF7F A8                         DB      0A8H
```

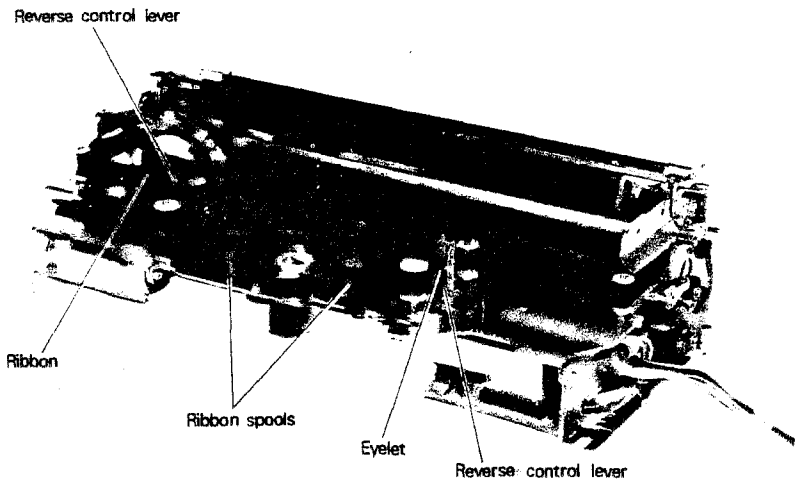
```
EF80 A8                         DB      0A8H
EF81 A8                         DB      0A8H
EF82 30                         DB      30H
EF83 00                         DB      0               ; f
EF84 10                         DB      10H
EF85 FC                         DB      0FCH
EF86 12                         DB      12H
EF87 00                         DB      0
EF88 10                         DB      10H             ; g
EF89 A8                         DB      0A8H
EF8A A8                         DB      0A8H
EF8B A8                         DB      0A8H
EF8C 78                         DB      78H
EF8D FE                         DB      0FEH            ; h
EF8E 08                         DB      08H
EF8F 08                         DB      08H
EF90 08                         DB      08H
EF91 F0                         DB      0F0H
EF92 00                         DB      0               ; i
EF93 88                         DB      88H
EF94 FA                         DB      0FAH
EF95 80                         DB      80H
EF96 00                         DB      0
EF97 40                         DB      40H             ; j
EF98 80                         DB      80H
EF99 80                         DB      80H
EF9A 7A                         DB      7AH
EF9B 00                         DB      0
EF9C 00                         DB      0               ; k
EF9D FE                         DB      0FEH
EF9E 20                         DB      20H
EF9F 50                         DB      50H
EFA0 88                         DB      88H
EFA1 00                         DB      0               ; l
EFA2 82                         DB      82H
EFA3 FE                         DB      0FEH
EFA4 80                         DB      80H
EFA5 00                         DB      0
EFA6 F8                         DB      0F8H            ; m
EFA7 08                         DB      08H
EFA8 F0                         DB      0F0H
EFA9 08                         DB      08H
EFAA F0                         DB      0F0H
EFAB F8                         DB      0F8H            ; n
EFAC 10                         DB      10H
EFAD 08                         DB      08H
EFAE 08                         DB      08H
EFAF F0                         DB      0F0H
EFB0 70                         DB      70H             ; o
EFB1 88                         DB      88H
EFB2 88                         DB      88H
EFB3 88                         DB      88H
EFB4 70                         DB      70H
EFB5 F8                         DB      0F8H            ; p
EFB6 28                         DB      28H
EFB7 28                         DB      28H
EFB8 28                         DB      28H
```

```
EFB9 10                 DB      10H
EFBA 10                 DB      10H             ; q
EFBB 28                 DB      28H
EFBC 28                 DB      28H
EFBD 28                 DB      28H
EFBE F8                 DB      0F8H
EFBF 00                 DB      0               ; r
EFC0 F8                 DB      0F8H
EFC1 10                 DB      10H
EFC2 08                 DB      08H
EFC3 08                 DB      08H
EFC4 90                 DB      90H             ; s
EFC5 A8                 DB      0A8H
EFC6 A8                 DB      0A8H
EFC7 A8                 DB      0A8H
EFC8 48                 DB      48H
EFC9 00                 DB      0               ; t
EFCA 08                 DB      08H
EFCB 7C                 DB      7CH
EFCC 88                 DB      88H
EFCD 00                 DB      0
EFCE 78                 DB      78H             ; u
EFCF 80                 DB      80H
EFD0 80                 DB      80H
EFD1 40                 DB      40H
EFD2 F8                 DB      0F8H
EFD3 38                 DB      38H             ; v
EFD4 40                 DB      40H
EFD5 80                 DB      80H
EFD6 40                 DB      40H
EFD7 38                 DB      38H
EFD8 78                 DB      78H             ; w
EFD9 80                 DB      80H
EFDA 60                 DB      60H
EFDB 80                 DB      80H
EFDC 78                 DB      78H
EFDD 88                 DB      88H             ; x
EFDE 50                 DB      50H
EFDF 20                 DB      20H
EFE0 50                 DB      50H
EFE1 88                 DB      88H
EFE2 98                 DB      98H             ; y
EFE3 A0                 DB      0A0H
EFE4 A0                 DB      0A0H
EFE5 A0                 DB      0A0H
EFE6 78                 DB      78H
EFE7 88                 DB      88H             ; z
EFE8 C8                 DB      0C8H
EFE9 A8                 DB      0A8H
EFEA 98                 DB      98H
EFEB 88                 DB      88H
EFEC 00                 DB      0               ; {
EFED 10                 DB      10H
EFEE 6C                 DB      6CH
EFEF 82                 DB      82H
EFF0 00                 DB      0
EFF1 00                 DB      0               ; LINE
```

```
EFF2 00                         DB      0
EFF3 FE                         DB      0FEH
EFF4 00                         DB      0
EFF5 00                         DB      0
EFF6 00                         DB      0               ; }
EFF7 82                         DB      82H
EFF8 6C                         DB      6CH
EFF9 10                         DB      10H
EFFA 00                         DB      0
EFFB 04                         DB      04H             ; TILDE
EFFC 02                         DB      02H
EFFD 04                         DB      04H
EFFE 08                         DB      08H
EFFF 04         FIN             DB      04H
F000   EFFF =   DIS             EQU     BASE+3FFH
F000   0000 =   OVER            EQU     DIS-FIN
F000                            PRT     'BYTES AVAILABLE = ',OVER
F000   00 FF                    FILL    OVER,0FFH
```

Setting of the ribbon:

1) Set the ribbon along the ribbon setting course shown in Photo 3.16.



CAUTION: See to it that the eyelet of the ribbon be situated between the ribbon detector lever and the ribbon spool.

Photo 3.16 Ribbon Setting Course

2) Check if the ribbon spools have been properly placed on the spool shafts.

Fig. 3.15 Setting of Ribbon Spool

3) Check if the ribbon is correctly engaged with the ribbon detector lever and the ribbon guide.

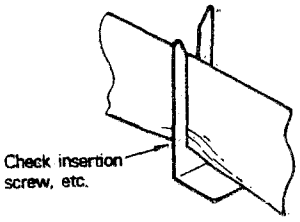


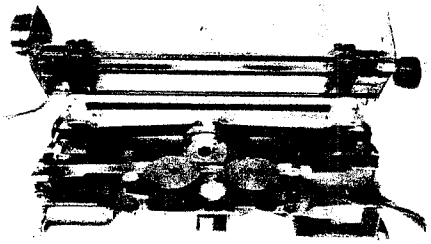
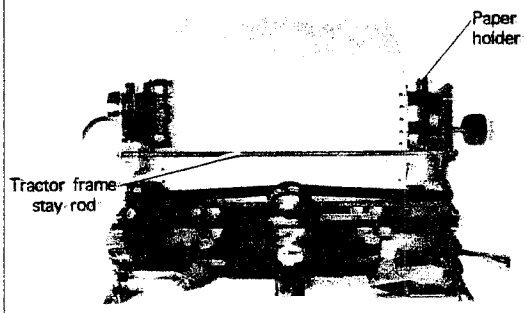
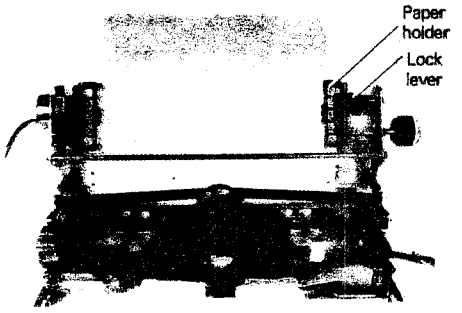
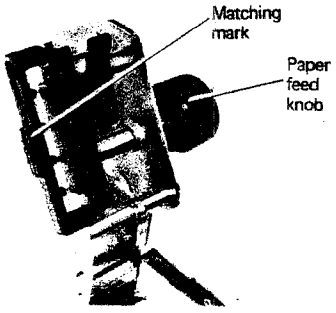
Fig. 3.16 Setting of Inked Ribbon

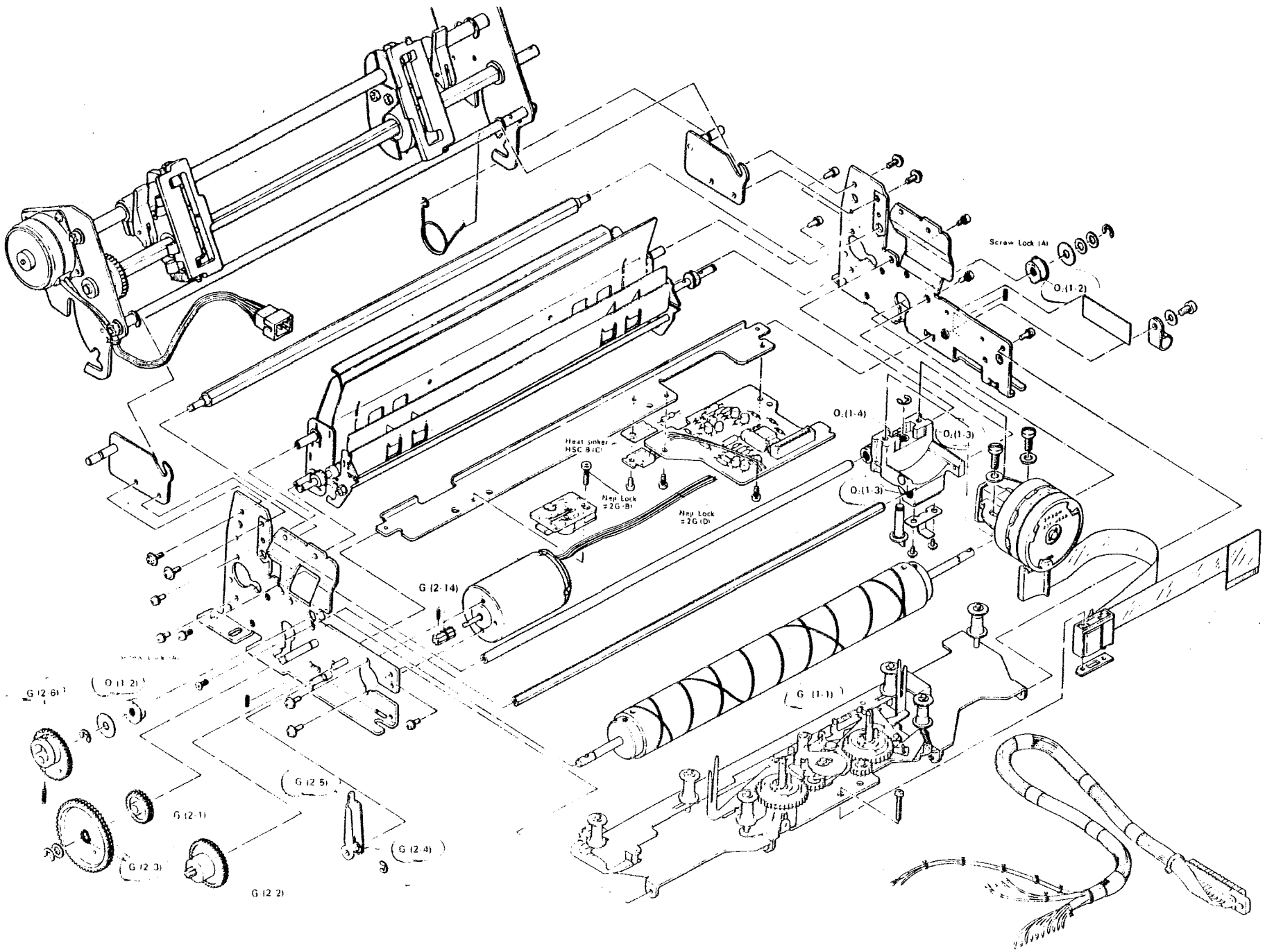
4) After setting the ribbon and conducting check 3) above, give two or three turns to the ribbon spools to see if the ribbon has been properly set or not.

CAUTION: In setting or removing the ribbon, be careful that printer parts be not stained with the ink contained in the ribbon.

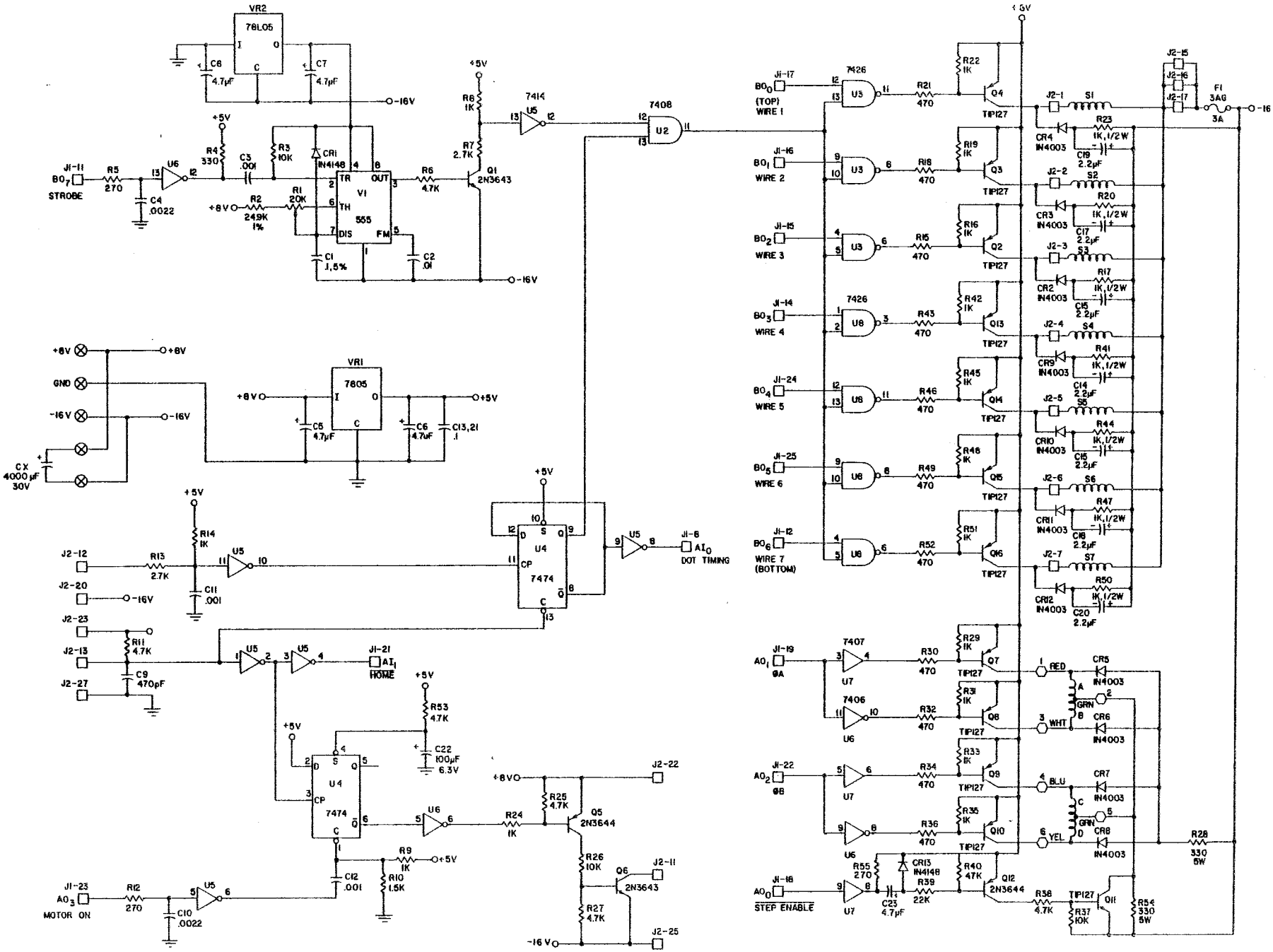
(b) Type T:

Continuous business form can be easily set in the printer by following the steps below.

SETTING STEPS		ILLUSTRATION
1.	Raise the tractor unit up (turn it toward the printer front side).	
2.	Insert the paper into the printer via the paper inlet located on the rear side of the printer.	 Photo 3.11
3.	After the leading end of the paper has come out of the printer, put the tractor unit back to its initial position (turn it toward the rear side of the printer). Then, raise the paper holders up, and engage paper feed holes on tractor pins. NOTE: The paper should be set on the tractor unit from the front side of the printer after its leading end has passed behind the tractor stay rod.	 Photo 3.12
4.	After putting the paper holders back into position, loosen the lock lever and adjust the tension of paper in the direction of width. (Turning the lock lever toward the front of the printer causes it to be loosened, and toward the rear, tightened.)	
5.	Referring to the matching mark, position the paper by operating the paper feed knob.	 Photo 3.13  Photo 3.14

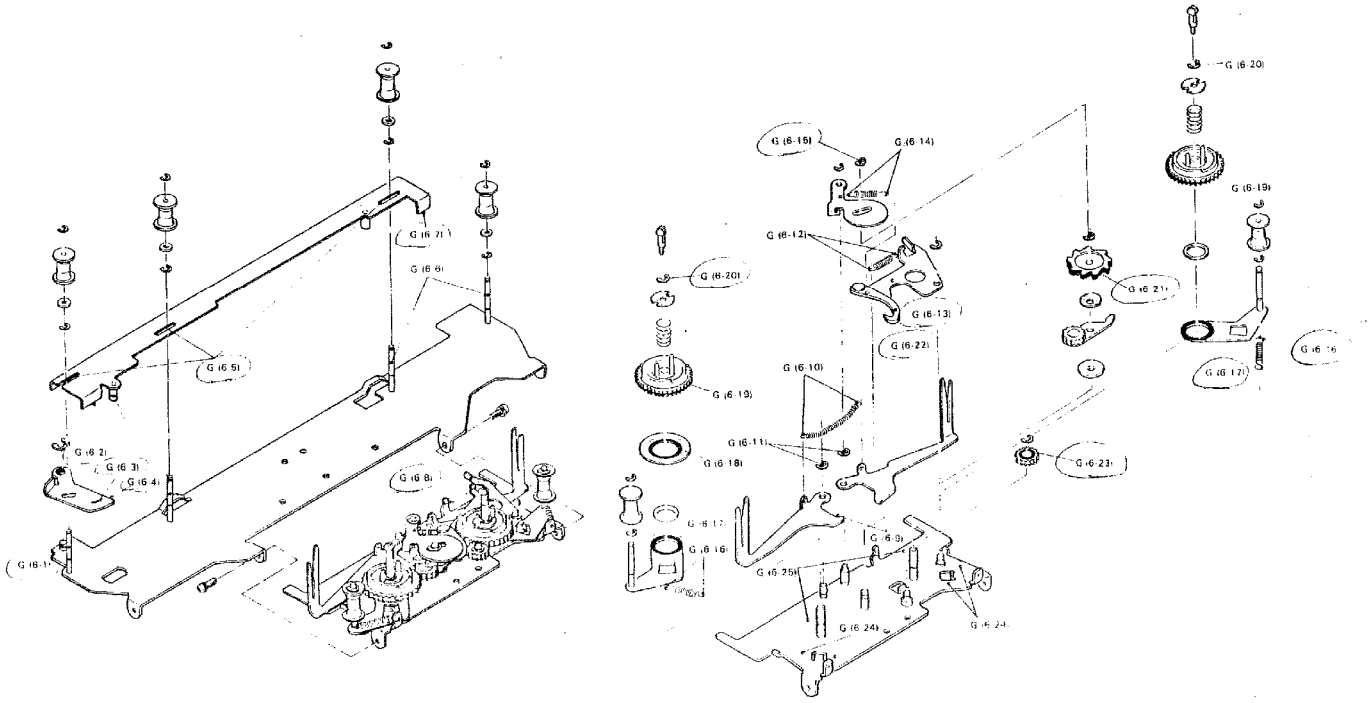


Lubrica ints



VECTOR GRAPHIC, INC.
WES VILLAGE, CALIFORNIA

VECTOR MP INTERFACE



Lubrication Points
(Ribbon Unit)