

CENTRONICS®

SALES & SERVICE CORPORATION

August 29, 1984

Enclosed please find copies of the programming sections for Models 358 and Horizon. Because manuals are difficult to obtain, I made copies for your use. If you need further information, please contact me.

Sincerely,

CENTRONICS DATA COMPUTER CORP.



Steven J. Hall
District Sales Manager

SJH:amc
Enclosures

SECTION 3

PROGRAMMER'S REFERENCE

3.1 PURPOSE OF THIS SECTION

This section provides samples of actual BASIC programs that exercise the functions of the printer. There are control codes and sequences that change character styles, spacing, tabbing, etc. These commands normally originate at the keyboard and are sent from the computer to the printer.

IMPORTANT

In order to control all of the functions of this powerful printer, your computer or word processor must be able to generate and send the 8th bit (128 decimal). However, most important functions do NOT require bit 8. For example, all character styles can be programmed with any computer that operates in BASIC.

3.2 COMPUTERS USED TO GENERATE THE PROGRAMS

The programs in this section were generated using an Apple® IIe, IBM PC and a COMPAQ™ Portable computer. Those programs with PRINT statements were generated on the Apple® IIe. Those with LPRINT statements (and those requiring bit 8) were generated on the IBM PC and COMPAQ™ Portable. Refer to your computer's reference documentation to determine whether your computer sends bit 8 to the printer, or try the program below.

SIMPLE CHECK FOR BIT 8—Run this program to see if bit 8 is being sent to the printer. Note that DIP switch SW1-7 must be ON to get the italic, accented character i.

```
10 REM CHECK FOR BIT 8
20 LPRINT CHR$(27);"I";CHR$(1);
30 LPRINT CHR$(132);:LPRINT:END
```

The printer should respond with an *italic*, accented letter i (*ı*). If the i is not *italic*, the printer did not receive the 8th bit.

3.3 DIP SWITCH SETTINGS

Some printer functions are affected by the setting of DIP switches on the logic board. Look for the NOTE included with the programs for any DIP switch settings required. The functions of the DIP switches are described in Section 2.

3.4 DECIMAL AND HEXADECIMAL NUMBERS

The programs listed in this section are written in BASIC (Beginner's All-purpose Symbolic Instruction Code), and the number system is the familiar decimal (0-9). For the programmer's convenience, the control sequence is also expressed in hexadecimal (0-9 + ABCDEF) for each function.

3.5 ASCII CODES AND CHARACTERS

ASCII (American Standard Code for Information Interchange) control codes and characters are assigned numbers 0 (NUL) thru 127 (DEL) decimal. Printable characters are codes 33 thru 126. A complete ASCII table (including numerical cross-reference) is included in the Appendix.

3.6 CONTROL CODES AND ESCAPE SEQUENCES

The control codes and ESC sequences are summarized on the next two pages. The summary is listed in ascending ASCII code sequence as a general rule. Following the summary, programs are listed for each function where appropriate and the result is included for comparison.

CONTROL CODE SUMMARY

ASCII	FUNCTION	PAGE
N/A	Self-Test Function: (hold LF button, turn power ON)	3-4
N/A	Hex Code Dump Function: (hold FF + LF buttons, turn power ON)	3-4
BEL	Sound Printer Alarm (beep!)—DIP Switch SW2-2 OFF	3-5
BS	Backspace Width of One Character	3-5
HT	Horizontal Tab Execution (used with ESC D)	3-6
LF	Line Feed Execution (value set by ESC 0, 1, 2, 3, A)	3-6
VT	Vertical Tab Execution (positions set by ESC B, ESC b)	3-7
FF	Form Feed Execution (values set by ESC C, ESC C 0)	3-8
CR	Carriage Return (automatic LF if DIP Switch SW3-4 is ON)	3-8
SO	Select Enlarged Characters (one line; cancelled by LF)	3-9
SI	Select Condensed Characters (effective until cancelled by DC2)	3-9
DC1	Remote Printer Select (DIP Switch SW3-1 OFF)	3-10
DC2	Cancel Condensed Characters	3-10
DC3	Remote Printer Deselect (DIP Switch SW3-1 OFF)	3-11
DC4	Cancel Enlarged Characters	3-11
CAN	Cancel Printer Buffer Data (on same line as CAN code)	3-12
DEL	Delete Last Character in Printer Buffer	3-12
ESC SO	Select Enlarged Characters (cancelled by LF)	3-13
ESC SI	Select Condensed Characters (cancelled by DC2)	3-13
ESC !	Select Print Mode (Modes 0-63)	3-14
ESC #	Cancel MSB Sequence (cancel bit 8 being forced to 1 or 0)	3-16
ESC %0	Select Internal CG (Character Generator) ROM	3-17
ESC %1	Select Download Character RAM	3-17
ESC &	Define Download Character (ASCII code, attribute)	3-18
ESC (Select NLQ (Near Letter Quality) Characters	3-21
ESC *	Selectable Density Graphics Mode	3-22
ESC -	Print Underline Function (1 = ON, 0 = OFF)	3-24
ESC /	Select VFU (Vertical Format Unit) Channel 0 to 7	3-25
ESC 0	1/8 Inch Line Spacing (8 lines per inch)	3-27
ESC 1	8 Pin Graphics Line Spacing	3-27
ESC 2	1/6 Inch Line Spacing (6 lines per inch)	3-28
ESC 3	1/3 Dot Diameter Line Spacing (n = 1 to 180)	3-28
ESC 4	Select Alternate Character Set (italics)	3-29
ESC 5	Cancel Alternate Character Set (italics)	3-29
ESC 6	Printable Area Code Expansion (International Characters)	3-30
ESC 7	Cancel Printable Area Code Expansion	3-30
ESC 8	Remote Paper End Deselect	3-31
ESC 9	Remote Paper End Select	3-31

(Continued on next page)

CONTROL CODE SUMMARY (CONT.)

ASCII	FUNCTION	PAGE
ESC :	Copy ROM CG to Download CG	3-32
ESC <	Unidirectional Print, 1 Line (cancelled by LF)	3-33
ESC =	Set MSB to 0 (force bit 8 to 0, cancelled by ESC #)	3-34
ESC >	Set MSB to 1 (force bit 8 to 1, cancelled by ESC #)	3-34
ESC ?	Multiple Function Graphics (Mode)	3-35
ESC @	Remote Printer Initialization Command	3-37
ESC A	1 Dot Line Spacing (n = 1 to 60)	3-37
ESC B	Vertical Tab Setting, 16 Positions	3-38
ESC C	Set Form Length, # of Lines (n = 1 to 127)	3-39
ESC C 0	Set Form Length, # of Inches (n = 1 to 22)	3-39
ESC D	Horizontal Tab Setting, 32 Positions	3-40
ESC E	Select Emphasized Print	3-41
ESC F	Cancel Emphasized Print	3-41
ESC G	Select Double Strike Print Mode (1 = ON, 0 = OFF)	3-42
ESC H	Cancel Double Strike Print Mode	3-42
ESC I	Enable Printable Control Codes (1 = ON, 0 = OFF)	3-43
ESC J	Paper Feed, 1/3 Dot Increment (n = 1 to 180)	3-44
ESC K	Normal Density Graphic Mode	3-45
ESC L	Double Density Graphic Mode	3-47
ESC M	Select Elite Characters (12 characters per inch)	3-48
ESC N	Enable Skip-Over Perforation (DIP switch 3-3 OFF)	3-49
ESC O	Disable Skip-Over Perforation	3-49
ESC P	Select Pica Characters (10 characters per inch)	3-48
ESC Q	Right Margin Setting (n = 1 to 255)	3-50
ESC R	Select International Character Set Mode (n = 0 to 8)	3-51
ESC S 0	Select Subscript Character Set (cancelled by ESC T)	3-53
ESC S 1	Select Superscript Character Set (cancelled by ESC T)	3-53
ESC T	Cancel Subscript, Superscript Character Sets	3-54
ESC U	Select Unidirectional Print Mode (1 = ON, 0 = OFF)	3-33
ESC W	Select Enlarged Print Mode (1 = ON, 0 = OFF)	3-54
ESC Y	Double Speed, Dual Density Graphics Mode	3-55
ESC Z	Quadruple Density Graphics Mode	3-56
ESC ^	Select 9-Bit & 16-Bit Graphics Mode	3-57
ESC b	VFU Tab Position Setting (8 channels, 16 positions)	3-61
ESC i	Incremental Typewriter Mode	3-62
ESC j	Reverse Paper Feed, 1/3 Dot Increment (n = 1 to 180)	3-44
ESC l	Left Margin Setting (n = 0 to 255)	3-50
ESC p	Proportional Spacing Mode (1 = ON, 0 = OFF)	3-63
ESC s	Half-Speed (quiet) Print Mode (1 = ON, 0 = OFF)	3-63

table4 (horizon1) 1-19-84

BEL**(Sound Printer Alarm)****BEL**

BASIC: CHR\$(7);**HEX:** 07**NAME:** BEL (Bell)**FUNCTION:** Sound Printer Alarm (beep)**PROGRAM:** LPRINT CHR\$(7);**RESULT:** ◀beep▶ (sounds for approximately 0.5 second)**NOTE**

DIP switch SW2-2 must be OFF for this function.

BS**(Backspace one character width)****BS**

BASIC: CHR\$(8);**HEX:** 08**NAME:** BS (Backspace)

FUNCTION: When BS is received, all codes in the printer buffer are printed, then the next print position is moved to the left (in the same line) by the width of one character. Multiple BS commands move the position one character width for each BS command. Subsequent characters will then be printed starting at the new position. In enlarged print mode, the new position is two character widths for each BS.

NOTE

BS function is most accurate in unidirectional print mode (ESC "U"). Also, BS is active only on the same line, thus characters following BS must be entered on the same line as previous characters.

```
PROGRAM:      10 REM BACKSPACE 10
                20 PRINT CHR$(27);"U"; CHR$(1);
                30 PRINT "12345678901234567890";
                40 FOR X = 1 TO 10
                50 PRINT CHR$(8);
                60 NEXT X
                70 PRINT "123456789/BACKSPACE"
                80 END
```

```
RESULT:      JRUN
                12345678901234567890BACKSPACE
```

HT**(Horizontal Tab Execution)****HT**

BASIC: CHR\$(9);

HEX: 09

NAME: HT (Horizontal Tab)

FUNCTION: Executes Horizontal Tab function. Values of horizontal tab position are determined by ESC D (N₁) (N₂) . . . (N_K); otherwise, tabs are automatically set every 8 characters when power is turned on. In enlarged mode, HT executes the tabs in twice the normal width value.

NOTE

Some Microsoft BASIC programs cannot use CHR\$(9). In this case, use CHR\$(137). This is established by 9 + 128 = 137.

```
PROGRAM:      5  REM HTAB POWER UP (8)
                10  PRINT "0123456789012345678901
                  234567890"
                20  FOR X = 1 TO 3
                25  PRINT CHR$ (9); "TAB";
                30  NEXT
```

```
RESULT:      0123456789012345678901234567890
                  TAB      TAB      TAB
```

LF**(Line Feed Execution)****LF**

BASIC: CHR\$(10);

HEX: 0A

NAME: LF (Line Feed)

FUNCTION: Receipt of LF causes the printing of all data in printer buffer, then paper is moved the effective line feed value. At power ON, line feed value is 1/6" (six lines per inch). Other values are established by ESC "A", ESC 0, ESC 1, ESC 3.

NOTE

When in OFF LINE mode, the LF button executes paper movement by the value established at power ON (six lines per inch) or by the above codes.

```
PROGRAM:      10  REM LINE FEED 3
                12  PRINT "*****";
                15  FOR X = 1 TO 3
                20  PRINT CHR$ (10);: REM LF
                25  NEXT
                30  PRINT "*****"
```

```
RESULT:      *****
```

```
*****
```

BASIC: CHR\$(11);**HEX:** 0B**NAME:** Vertical Tab Execution

FUNCTION: Receipt of VT causes the printing of all data in the print buffer, then a line feed to the position established by ESC B or ESC b. VT cancels enlarged characters set by SO. If no vertical tab has been set, VT simply performs a line feed.

PROGRAM:

```
10 REM ESC B VERT TAB
20 FOR X= 0 TO 11
30 LPRINT X: NEXT
40 LPRINT CHR$(27); "B"; CHR$(2);
45 LPRINT CHR$(6); CHR$(10); CHR$(0);
50 FOR X = 1 TO 2
60 LPRINT CHR$(27); "j"; CHR$(180);
70 NEXT
80 FOR Z = 1 TO 3
90 LPRINT CHR$(11); " VTAB"; : NEXT
95 LPRINT : END
```

RESULT:

```
0
1
2 VTAB
3
4
5
6 VTAB
7
8
9
10 VTAB
11
```

FF**(Form Feed Execution)****FF**

BASIC: CHR\$(12);

HEX: 0C

NAME: FF (Form Feed)

FUNCTION: When the FF code is received, all data in the printer buffer is printed, then paper is fed the number of lines or inches in effect, established by ESC C, ESC C 0 or by SW3-2. The normal Form Feed is 11 inches (SW3-2 OFF).

NOTE

Top of form is established by the position of the paper prior to execution of any line feeds. Whenever FF is received, (or when FF button is pressed when OFF LINE) the paper is advanced the appropriate amount relative to the top of form position.

```
PROGRAM:      10 REM FF 3 LINES=1/2"
               20 PRINT "*****";
               30 PRINT CHR$ (27);"C"; CHR$ (3);
               40 PRINT CHR$ (12);: REM FF
               50 PRINT "*****"
```

RESULT: *****

CR**(Carriage Return)****CR**

BASIC: CHR\$(13);

HEX: 0D

NAME: CR (Carriage Return)

FUNCTION: When the CR code is received, all data in the printer buffer is printed. If Auto Linefeed is active (SW3-4 ON) paper will be fed one line automatically; otherwise the next print position is at the first character on the same line.

NOTE

Some word processing programs and interfaces force a linefeed after a CR. In this case CR without a linefeed is not functional unless the program/interface can be modified to not issue the linefeed.

```
PROGRAM:      10 REM CR OVERPRINT
               15 PRINT CHR$ (27);"U"; CHR$ (1);
               20 PRINT "0000000000";
               30 PRINT CHR$ (13);: REM CR
               40 PRINT "//////////"
```

RESULT: 0000000000

SO**(Select Enlarged Characters)****SO**

BASIC: CHR\$(14);

HEX: 0E

NAME: SO (Shift Out)**FUNCTION:** When SO code is received, all subsequent characters and spaces are printed double-width on the same line. This function is cancelled by a linefeed, DC4, ESC ! or ESC W.

NOTE

All character styles can be enlarged, and this function can be mixed within a line. ESC SO is functionally identical to SO. However, ESC W 1 selects enlarged characters until cancelled by ESC W 0.

```
PROGRAM:      10  REM  ENLARGED
                20  PRINT "JACK SPRAT COULD EAT NO ";
                30  PRINT  CHR$ (14);"FAT"
```

RESULT: JACK SPRAT COULD EAT NO **FAT**

SI**(Select Condensed Characters)****SI**

BASIC: CHR\$(15);

HEX: 0F

NAME: SI (Shift In)**FUNCTION:** When SI code is received, all data in the printer buffer is printed, then all subsequent characters will be condensed until cancelled by the DC2 code.

NOTE

The Elite and NLQ character styles cannot be condensed. Also, ESC SI is functionally identical to SI.

```
PROGRAM:      10  REM  CONDENSED
                20  PRINT "HIS WIFE COULD EAT NO";
                30  PRINT  CHR$ (15);"LEAN"
                40  PRINT  CHR$ (18);: REM  DC2
                50  PRINT "AND SO BETWEEN THE TWO
                        OF THEM"
                60  PRINT "THEY LICKED THE PLATTER CLEAN"
```

RESULT: HIS WIFE COULD EAT NO LEAN
AND SO BETWEEN THE TWO OF THEM
THEY LICKED THE PLATTER CLEAN

DC1

(Remote Printer Select)

DC1

BASIC: CHR\$(17);

HEX: 11

NAME: DC1 (Device Control 1)

FUNCTION: The DC1 code selects the printer, enabling data reception. DC1 is normally used to cancel DC3, which de-selects the printer.

NOTE

In order for DC1 to function as described here, the printer must be ON LINE and DIP switch SW3-1 must be OFF. If the DC1 code is received along with data, all data stored in the printer buffer before DC1 will be ignored (not printed).

```
PROGRAM:      10 REM PRINTER SELECT
               20 PRINT "$$$$$$$$$$$$$$$$$"
               30 PRINT CHR$ (19);
               40 PRINT "#####"
               50 PRINT CHR$ (17);
               60 PRINT "$$$$$$$$$$$$$$$$$"
```

```
RESULT:      $$$$$$$$$$$$$$$$$
             $$$$$$$$$$$$$$$$$
```

DC2

(Cancel Condensed Mode)

DC2

BASIC: CHR\$(18);

HEX: 12

NAME: DC2 (Device Control 2)

FUNCTION: The DC2 code is used to cancel condensed characters as selected by SI or ESC SI.

NOTE

The condensed character mode is not affected by linefeed as with SO and ESC SO (enlarged). Elite and NLQ characters cannot be condensed.

```
PROGRAM:      10 REM DC2 CANCEL CONDENSED
               20 PRINT "NORMAL "; CHR$ (15);
               30 PRINT "CONDENSED";
               40 PRINT CHR$ (18); " NORMAL"
```

```
RESULT:      NORMAL CONDENSED NORMAL
```

DC3

(Remote Printer Deselect)

DC3

BASIC: CHR\$(19);

HEX: 13

NAME: DC3 (Device Control 3)

FUNCTION: The DC3 code de-selects the printer, inhibiting data reception. DC3 is used to cancel DC1, which selects the printer.

NOTE

In order for DC3 to function as described here, the printer must be ON LINE and DIP switch SW3-1 must be OFF.

PROGRAM:

```
10 REM PRINTER DE-SELECT
20 PRINT "PRINT THIS LINE"
30 PRINT CHR$ (19);
40 PRINT "IGNORE THIS LINE"
50 PRINT CHR$ (17);
60 PRINT "SELECTED ONCE AGAIN"
```

RESULT: PRINT THIS LINE
SELECTED ONCE AGAIN

DC4

(Cancel Enlarged Characters)

DC4

BASIC: CHR\$(20);

HEX: 14

NAME: DC4 (Device Control 4)

FUNCTION: The DC4 code is used to cancel enlarged characters selected by SO and ESC SO.

NOTE

Enlarged characters selected by ESC W1 or ESC! are not affected by the DC4 code.

PROGRAM:

```
10 REM CANCEL ENLARGED
20 PRINT "ACROSS THE "; CHR$ (14);
30 PRINT "WIDE"; CHR$ (20);: REM
   DC4
40 PRINT " MISSOURI"
```

RESULT: ACROSS THE **WIDE** MISSOURI

CAN

(Cancel Printer Buffer Data)

CAN

BASIC: CHR\$(24);

HEX: 18

NAME: CAN (Cancel)

FUNCTION: When the CAN code is received, all prior data stored in the printer buffer on the same line is cancelled (not printed).

NOTE

The DEL (Delete) code performs a similar function, but affects only the last character stored in the printer buffer.

```
PROGRAM:      10  REM  CANCEL CHR$(24)
                 20  PRINT "THAT"
                 30  PRINT "@#!*##!!"; CHR$ (24); "
                   CAT!"
```

RESULT: THAT
 CAT!

DEL

(Delete Last Printer Buffer Character)

DEL

BASIC: CHR\$(127);

HEX: 7F

NAME: DEL (Delete)

FUNCTION: When the DEL code is received, the last character in the printer buffer on the same line is deleted (not printed).

NOTE

The CAN (Cancel) code performs a similar function, but it affects all prior data on the same line.

```
PROGRAM:      10  REM  DEL LAST CHAR
                 20  PRINT "BEAUTY IS ONLY SKINY";
                 30  PRINT CHR$ (127); " DEEP"
```

RESULT: BEAUTY IS ONLY SKIN DEEP

ESC SO (Select Enlarged Characters)

ESC SO

BASIC: CHR\$(27);CHR\$(14);

HEX: 1B 0E

NAME: ESC SO (Escape Shift Out)

FUNCTION: Same as SO. When ESC SO code is received, all subsequent characters and spaces are printed double-width on the same line. This function is cancelled by a linefeed, DC4, ESC ! or ESC W.

NOTE

All character styles can be enlarged and mixed within a line. Note that ESC W 1 selects enlarged characters until cancelled by ESC W 0.

```
PROGRAM:      10  REM ENLARGED ESC SO
                20  PRINT "UNDER THE ";
                30  PRINT CHR$ (27); CHR$ (14);
                40  PRINT "BIG"; CHR$ (20);" TOP!"
```

RESULT: UNDER THE **B I G** TOP!

ESC SI (Select Condensed Characters)

ESC SI

BASIC: CHR\$(27);CHR\$(15);

HEX: 1B 0F

NAME: ESC SI (Escape Shift In)

FUNCTION: Same as SI. When ESC SI code is received, all subsequent characters will be condensed until cancelled by the DC2 code.

NOTE

The Elite and NLQ character styles cannot be condensed.

```
PROGRAM:      10  REM CONDENSED ESC SI
                20  PRINT "FOR ONLY ONE ";
                30  PRINT CHR$ (27); CHR$ (15);"
                   THIN"; CHR$ (18);" DIME!"
```

RESULT: FOR ONLY ONE **THIN** DIME!

ESC !**(Print Mode 0-63 Select)****ESC !**

BASIC: CHR\$(27);"!";CHR\$(n);**HEX:** 1B 21 (n)**NAME:** Print Mode (n = 0 to 63)**FUNCTION:** Print Mode selects one of 16 different print styles/combinations. The ESC !(n) sequence takes precedence over any prior print style command such as SO, SI etc.**NOTE**

Whenever leaving Print Mode to select NLQ (near letter quality) characters, always send ESC ! 0 to cancel Print Mode styles in effect, otherwise, NLQ could be enlarged or other styles could be affected.

```
PROGRAM:          10 REM SELECT MODE 24  
(TYPICAL)         20 PRINT "MODE 24 ";  
                   30 PRINT CHR$ (27);"!"; CHR$ (24);  
                   40 PRINT "ABCabc123"
```

```
RESULT:          MODE 0  ABCabc123  
(TYPICAL)         MODE 1  ABCabc123  
  
                   MODE 4  ABCabc123  
  
                   MODE 8  ABCabc123  
  
                   MODE 16 ABCabc123  
  
                   MODE 17 ABCabc123  
  
                   MODE 20 ABCabc123  
  
                   MODE 24 ABCabc123  
  
                   MODE 32 ABCabc123  
  
                   MODE 33 ABCabc123  
  
                   MODE 36 ABCabc123  
  
                   MODE 40 ABCabc123  
  
                   MODE 48 ABCabc123  
  
                   MODE 49 ABCabc123  
  
                   MODE 52 ABCabc123  
  
                   MODE 56 ABCabc123
```

NOTE: Refer to table for Print Modes 0 to 63

(ESC ! Print Mode 0-63 continued)

PRINT MODE BYTE (n=0-63)

n	ENLG	DBLS	EMPH	COND	ELIT
0*					
1					●
2*					
3					●
4				●	
5					●
6				●	
7					●
8			●		
9					●
10			●		
11					●
12			●		
13					●
14			●		
15					●
16		●			
17		●			●
18		●			
19		●			●
20		●		●	
21		●			●
22		●		●	
23		●			●
24		●	●		
25		●			●
26		●	●		
27		●			●
28		●	●		
29		●			●
30		●	●		
31		●			●

n	ENLG	DBLS	EMPH	COND	ELIT
32	●				
33	●				●
34	●				
35	●				●
36	●			●	
37	●				●
38	●			●	
39	●				●
40	●		●		
41	●				●
42	●		●		
43	●				●
44	●		●		
45	●				●
46	●		●		
47	●				●
48	●	●			
49	●	●			●
50	●	●			
51	●	●			●
52	●	●		●	
53	●	●			●
54	●	●		●	
55	●	●			●
56	●	●	●		
57	●	●			●
58	●	●	●		
59	●	●			●
60	●	●	●		
61	●	●			●
62	●	●	●		
63	●	●			●

*Modes 0 and 2 are Pica 10 cpi

ENLG=Enlarged; DBLS=Double Strike; EMPH=Emphasized; COND=Condensed; ELIT=Elite (12 cpi)

ESC #**(Cancel MSB Sequence)****ESC#**

BASIC: CHR\$(27)"#";**HEX:** 1B 23**NAME:** Cancel MSB Sequence (MSB = 0, MSB = 1)**FUNCTION:** ESC # cancels control of MSB (8th bit) as set by ESC = (MSB=0) and ESC > (MSB = 1). Thus, ESC # releases the 8th bit from forced condition and allows it to change with data.

PROGRAM:

```
10 REM ESC # CANCEL MSB SEQUENCE
15 LPRINT "          (NORMAL)"
20 LPRINT "If you wish to go nowhere..."
25 LPRINT "Any road will take you there!"
30 LPRINT:LPRINT CHR$(27);">";
35 LPRINT "          (ITALIC)"
40 LPRINT "When in Rome do as the Italics do"
45 LPRINT:LPRINT CHR$(27);"#";:REM CANCEL MSB
50 LPRINT "          (NORMAL)"
55 LPRINT "Doe... a deer, a female deer..."
```

RESULT:

```
          (NORMAL)
If you wish to go nowhere...
Any road will take you there!

          (ITALIC)
When in Rome do as the Italics do

          (NORMAL)
Doe... a deer, a female deer...
```

ESC % 0**(Select Internal CG ROM)****ESC % 0**

BASIC: CHR\$(27)“%”;CHR\$(0);CHR\$(0);**HEX:** 1B 25 00 00**NAME:** ROM Character Generator selection**FUNCTION:** This code sequence selects the character generator in the Internal ROM and is normally used to cancel the Download CG as selected by ESC % 1.**PROGRAM:** (See example in ESC &)

ESC % 1**(Select Download CG ROM)****ESC % 1**

BASIC: CHR\$(27)“%”;CHR\$(1);CHR\$(0);**HEX:** 1B 25 01 00**NAME:** Download Character Generator Selection**FUNCTION:** This code sequence selects the Download Character set previously defined by ESC & code.**NOTE**

DIP switch SW2-3 must be OFF to enable Download Buffer for this function.

PROGRAM: (See example in ESC &)

ESC &

(Define Download Character)

ESC &

BASIC: CHR\$(27);"&";CHR\$(0);CHR\$(n);CHR\$(n);CHR\$(a);CHR\$(p)...CHR\$(p,); **HEX:** 1B 26

NAME: Define Download Character

FUNCTION: ESC & (etc.) defines the dot pattern of a download character, assigns an ASCII code to it so it can be accessed (CHR\$(n)) and defines an attribute (CHR\$(a)) that contains starting and ending column data, plus descender information (if MSB = 1, no descender). In the example below, ESC & defines the Greek letter omega and assigns the ASCII character "#" to it.

NOTE
DIP switch SW2-3 must be OFF in order for this function to work properly.

```

PROGRAM: 10 REM # = OMEGA
          20 LPRINT CHR$(27); "&"; CHR$(0); "##";
          30 LPRINT CHR$(139); :REM ATTRIBUTE
          32 LPRINT CHR$(50); CHR$(72); CHR$(6); CHR$(64);
          33 LPRINT CHR$(0); CHR$(64); CHR$(6); CHR$(72);
          34 LPRINT CHR$(50); CHR$(0); CHR$(0);
          45 LPRINT "#####"
          47 LPRINT "The resistance = 7";
          50 LPRINT CHR$(27); "%"; CHR$(1); CHR$(0);
          55 LPRINT "#"
          70 LPRINT "#####"
          85 LPRINT CHR$(27); "%"; CHR$(0); CHR$(0);
          90 LPRINT "#####"
          95 LPRINT:END

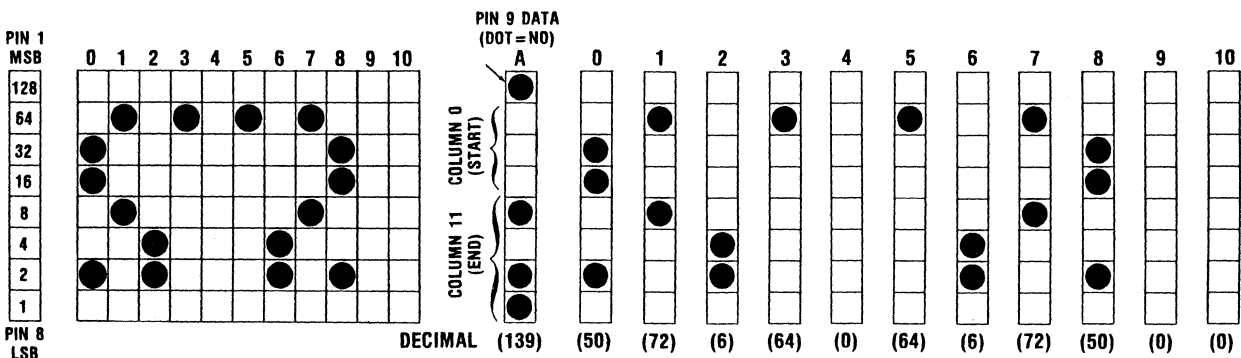
```

RESULT:

```

#####
The resistance = 7Ω
Ω Ω Ω Ω Ω Ω Ω Ω Ω Ω
#####

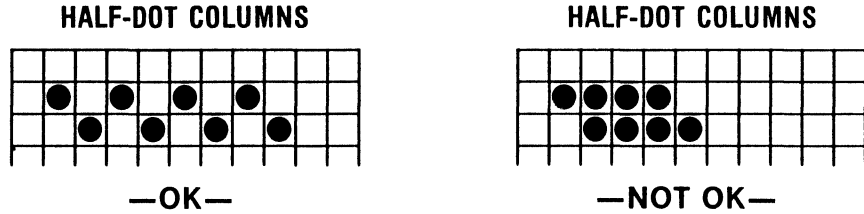
```



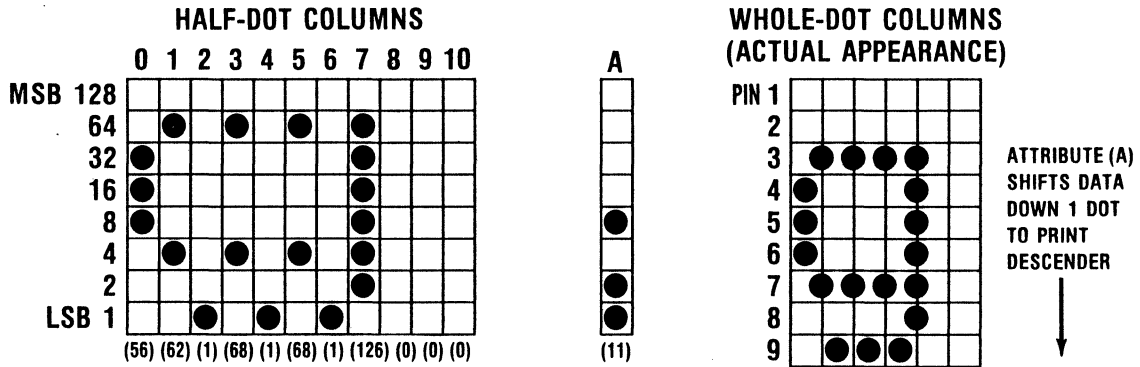
(Continued on next page)

(ESC & continued)

ADJACENT PINS—When defining a half-dot column pattern for a download character, be certain **NOT** to define adjacent dots; that is, if a pin is fired, it cannot be fired again in the next half-dot column. To do so could cause damage to the printhead or associated circuitry (See illustration).



DESCENDER—The attribute (A) defines descender data (for pin 9). If the MSB is zero (no dot), there is a descender (A = 11 decimal). Refer to the illustration of lower case letter “g”. The character is defined in the standard 11 × 8 pattern, attribute (A) shifts the pattern down one dot and it is printed with a descender.



```

PROGRAM: 10 REM ESC & (* = LOWER CASE g)
15 LPRINT CHR$(27);": ";CHR$(0);CHR$(0);CHR$(0);
20 LPRINT CHR$(27);"&";CHR$(0);"***";
30 LPRINT CHR$(11);:REM ATTRIBUTE FOR DESCENDER
35 REM HERE COMES LOWER CASE "G" DATA
40 LPRINT CHR$(56);CHR$(68);CHR$(1);CHR$(68);
45 LPRINT CHR$(1);CHR$(68);CHR$(1);CHR$(126);
50 LPRINT CHR$(0);CHR$(0);CHR$(0);
55 LPRINT "*****"
60 LPRINT CHR$(27);"%";CHR$(1);CHR$(0);
65 LPRINT "U*U*U*U*U*U*U*U*U*U"
70 LPRINT CHR$(27);"%";CHR$(0);CHR$(0);
75 LPRINT "*****"
80 END
    
```

RESULT:

```

*****
UgUgUgUgUgUgUgUgUgU
*****
    
```

(Continued on next page)

(ESC & continued)

PROPORTIONAL: Attribute (A) defines the start and end columns of the character. If the attribute is 11 decimal (descender) or 139 decimal (no descender) for all characters, the character set is monospaced at 10 cpi. However, if the end column is less than those values, the character has a proportional attribute and this will be evident in the Proportional Spacing Mode (ESC p). If ESC p is not active, the attribute defaults to monospace. A program for a simple proportional character is provided below.

NOTE

Proportional characters are printed in Emphasized Mode. In this mode, all dots are fired twice in the horizontal plane. If space is desired between characters, the end column should be defined as at least two half-dot columns greater than the last data column (dots) of the character. Trailing non-data columns must be filled with zeroes so that eleven (11) columns are defined.

```
PROGRAM: 10 REM ESC & PROPORTIONAL SPACING
          15 LPRINT CHR$(27);":":CHR$(0);CHR$(0);CHR$(0);
          20 LPRINT CHR$(27);"&":CHR$(0);"***";
          25 LPRINT CHR$(132);:REM PROPORTIONAL ATTRIBUTE
          30 REM HERE DEFINE SIMPLE DOWNLOAD CHARACTER
          35 LPRINT CHR$(254);CHR$(0);CHR$(0);CHR$(0);
          40 LPRINT CHR$(0);CHR$(0);CHR$(0);CHR$(0);
          45 LPRINT CHR$(0);CHR$(0);CHR$(0);
          50 GOSUB 70
          55 LPRINT CHR$(27);"p";CHR$(1);:REM PROPORTIONAL
          60 GOSUB 70
          65 LPRINT CHR$(27);"p";CHR$(0);:END
          70 LPRINT "* * * * * * * * * * *"
          75 LPRINT CHR$(27);"%":CHR$(1);CHR$(0);
          80 LPRINT "U*U*U*U*U*U*U*U*U*U"
          85 LPRINT CHR$(27);"%":CHR$(0);CHR$(0);
          90 RETURN
```

```
RESULT: * * * * * * * * * * *
         UI UI UI UI UI UI UI UI UI U
         * * * * * * * * * * *
         UIUIUIUIUIUIUIUI
```

ESC ((Select NLQ Characters) ESC (

BASIC: CHR\$(27);"(";

HEX: IB 28

NAME: NLQ (Near Letter Quality)

FUNCTION: When ESC "(" is received, all subsequent characters will be printed in near-letter quality. NLQ is produced by two passes of the print head.

NOTE

NLQ is automatically selected by holding the FF button as power is turned ON. Also, NLQ is selected at power ON using DIP switch SW2 settings. Refer to power-up Print Mode chart below.

Also, NLQ characters can be set to proportional spacing by ESC p and enlarged by SO and ESC W codes. However, NLQ cannot be condensed or italicized.

```
PROGRAM: 10 REM SELECT NLQ
          20 PRINT CHR$ (27);"(";
          30 PRINT "THIS IS NLQ (Near Lett
            er Quality)"
          40 PRINT "ABCDEFGHJKLMNOPQRSTUVWXYZ
            WXYZ"
          50 PRINT "abcdefghijklmnopqrstuv
            wxyz"
          60 PRINT "1234567890!@#%&^*()_+
            {}<>"
          70 PRINT CHR$ (27);"P";: REM F
            ICA
```

RESULT: THIS IS NLQ (Near Letter Quality)
ABCDEFGHIJKLMNPOQRSTUVWXYZ
abcdefghijklmnopqrstuvwxy
1234567890!@#%&^*()_+{}<>

DIP Switch SW2	7	6	5	4
NLQ (10 cpi)	OFF	OFF	ON	ON
Enlarged NLQ	ON	OFF	ON	ON

ESC ***(Selectable Density Graphics Mode)****ESC ***

BASIC: CHR\$(27);“*””;CHR\$(M);CHR\$(n1);CHR\$(n2);**HEX:** 1B 2A**NAME:** Selectable Density Graphics Mode
(M = 0 to 6) (n1 = remainder) (n2 = # ÷ 256)**FUNCTION:** This code sequence selects 1 of 7 graphics densities based on the value of M (M = 0 to 6) and is defined below. Values n1 and n2 specify the number of graphic characters to be printed.

n1 and n2 are calculated as follows:

- n2 is an integer (whole number) = # of characters to print ÷ 256
- n1 is the remainder of the division in n2

Example: # of characters to print = 680
n2 = 680 ÷ 256 = 2
n1 = 680 - (2 × 256) = 168

Thus, to print 680 graphic characters, the BASIC code sequence is:

CHR\$(27);“*””;CHR\$(M);CHR\$(168);CHR\$(2);

MODE SELECTIONS

MODE (M)	DENSITY	SAME AS	DESCRIPTION
0	SINGLE	ESC K	60 DOTS/INCH; 480 CHARACTERS/LINE
1	LO-SPEED DOUBLE	ESC L	120 DOTS/INCH; 960 CHARACTER/LINE.
2	HI-SPEED DOUBLE	ESC Y	SAME AS 1, BUT FASTER. CONSECUTIVE DOTS IN SAME ROW NOT PRINTED.
3	QUADRUPLE	ESC Z	240 DOTS/INCH; 1920 CHARACTER/LINE. CONSECUTIVE DOTS IN SAME ROW NOT PRINTED.
4	N/A	N/A	80 DOTS/INCH; 640 CHARACTERS/LINE.
5	1 TO 1 RATIO	N/A	72 DOTS/INCH; 576 CHARACTERS/LINE.
6	N/A	N/A	90 DOTS/INCH; 720 CHARACTERS/LINE.

(Continued on the next page)

ESC -**(Print Underline)****ESC -**

BASIC: CHR\$(27);" - ";CHR\$(1);**HEX:** 1B 2D 01**NAME:** Print Underline

FUNCTION: When ESC - 1 is received, subsequent characters and spaces are underlined by the 9th pin at the "10th" pin position as one continuous underline. Underline continues line to line until cancelled by ESC - 0. Individual words or characters may be underlined, as the underline can be turned on and off anywhere in a line.

NOTE

The underline is printed as a separate, continuous line after the text is completed. However, if character styles are changed, the underline is drawn after each change. To draw the underline, paper is moved up one dot width.

```
PROGRAM:      10  REM  UNDERLINE
                20  PRINT "Let's underline an entire
                  line!"
                25  PRINT
                30  PRINT CHR$ (27);"-"; CHR$ (1);
                40  PRINT "An entire line!"
                50  PRINT CHR$ (27);"-"; CHR$ (0);
                60  PRINT : PRINT "No underline here"
                70  PRINT : PRINT "Let's underline ";
                80  PRINT CHR$ (27);"-"; CHR$ (1);
                  "one";
                90  PRINT CHR$ (27);"-"; CHR$ (0);
                  "word"
```

```
RESULT:      Let's underline an entire line!
                An entire line!
                No underline here
                Let's underline one word
```

BASIC: CHR\$(27);"/";CHR\$(n);**HEX:** 1B 2F (n)**NAME:** Select VFU Channel (n = 0 to 7)

FUNCTION: ESC / (n) selects one of eight channels for vertical tab format (called VFU; Vertical Format Unit). This function is used with ESC b, which establishes the vertical tab stops within each channel. The basic purpose of this function is to divide a page into two or more channels with independent tab stops and independent Top of Form.

PROGRAM:

```
10 REM ESC / VFU CHANNEL 0-7
15 LPRINT CHR$(27);"C";CHR$(10);:REM FORM LENGTH 10
20 FOR Z = 0 TO 47: LPRINT Z:NEXT
25 FOR X = 1 TO 8: LPRINT CHR$(27);"j";CHR$(180);
30 NEXT X: LPRINT " <--- TOP OF FORM #1"
35 LPRINT CHR$(27);"b";CHR$(1);:REM CHANNEL 1
40 LPRINT CHR$(2);CHR$(6);CHR$(9);CHR$(0);:REM CH1 TABS
45 LPRINT CHR$(27);"/";CHR$(1);:REM SELECT CH1
50 LPRINT CHR$(12);:REM TOP OF FORM COMMAND
55 LPRINT " <--- VFU 1 TOP OF FORM"
60 FOR Y = 1 TO 3: LPRINT CHR$(11);:REM VERT TAB
65 LPRINT " <--- VFU 1; TAB ";Y:NEXT Y
70 LPRINT CHR$(27);"b";CHR$(2);:REM CHANNEL 2
75 LPRINT CHR$(3);CHR$(5);CHR$(8);CHR$(0);:REM CH2 TABS
80 LPRINT CHR$(27);"/";CHR$(2);:REM SELECT CH2
85 LPRINT CHR$(12);:REM TOP OF FORM COMMAND
90 LPRINT " <--- VFU 2 TOP OF FORM"
95 FOR N = 1 TO 3: LPRINT CHR$(11);:REM VERT TAB
100 LPRINT " <--- VFU 2; TAB ";N:NEXT
105 END
```

RESULT: (See result of program on next page)

(ESC / continued)

0 <--- TOP OF FORM #1
1
2
3
4
5
6
7
8
9
10 <--- VFU 1 TOP OF FORM
11
12 <--- VFU 1; TAB 1
13
14
15
16 <--- VFU 1; TAB 2
17
18
19 <--- VFU 1; TAB 3
20
21
22
23
24
25
26
27
28
29
30 <--- VFU 2 TOP OF FORM
31
32
33 <--- VFU 2; TAB 1
34
35 <--- VFU 2; TAB 2
36
37
38 <--- VFU 2; TAB 3
39
40
41
42
43
44
45
46
47

(program on previous page)

ESC 0

(1/8-Inch Line Spacing)

ESC 0

BASIC: CHR\$(27);"0";

HEX: 1B 00

NAME: 1/8-Inch Line Spacing (8 lpi)

FUNCTION: When ESC 0 is received, line spacing is set at 1/8 inch. This spacing corresponds with 8 lines per inch. This spacing is in effect until cancelled by ESC 2 (normal 1/6 inch spacing) or by ESC A, ESC 1, ESC 2, or ESC 3. ESC J is in effect for one line only, so does not cancel ESC 0.

PROGRAM:

```
10 REM 1/8 INCH LINE SPACE
20 PRINT CHR$ (27);"0";
30 FOR X = 1 TO 4
40 PRINT "-----": NEXT
50 PRINT CHR$ (27);"2";
60 FOR Z = 1 TO 4
70 PRINT "-----": NEXT
```

RESULT:

```
-----
-----
-----
-----
-----
-----
-----
-----
```

ESC 1

(8-Pin Graphics Line Spacing 7/60")

ESC 1

BASIC: CHR\$(27);"1";

HEX: 1B 01

NAME: 8-Pin Graphics Line Spacing (7/60")

FUNCTION: When ESC 1 is received, the line-spacing value is set at 7/60". This is the spacing normally used between 8-pin graphic lines. This spacing is cancelled by ESC A, ESC 0, ESC 2 or ESC 3. ESC J is in effect for only line only, so does not cancel ESC 1.

PROGRAM:

```
10 REM 8-PIN GRAF 7/60
20 PRINT CHR$ (27);"1";
30 FOR X = 1 TO 4
40 PRINT "-----": NEXT
50 PRINT CHR$ (27);"2";
60 FOR Z = 1 TO 4
70 PRINT "-----": NEXT
```

RESULT:

```
-----
-----
-----
-----
-----
-----
-----
-----
```

ESC 2

(1/6-Inch Line Spacing)

ESC 2

BASIC: CHR\$(27);"2";

HEX: 1B 02

NAME: Normal (1/6 Inch) Line Spacing

FUNCTION: ESC 2 sets the normal line spacing (6 lpi) and is normally used to cancel special line spacing functions such as ESC A, ESC 0, ESC 1, and ESC 3.

PROGRAM:

```
10 REM 1/6 INCH LF
20 PRINT CHR$(27);"A"; CHR$(5);
30 FOR X = 1 TO 6
40 PRINT "-----": NEXT
50 PRINT CHR$(27);"2";: REM 1/6"
60 FOR Z = 1 TO 4
70 PRINT "-----": NEXT
```

RESULT:

```
-----
-----
-----
-----
-----
-----
-----
-----
```

ESC 3

(N/180-Inch Line Spacing)

ESC 3

BASIC: CHR\$(27);"3";(n);

HEX: IB 03 (n)

NAME: N/180-Inch Line Spacing (n = 1 to 180)

FUNCTION: This function allows the setting of line spacing in increments of 1/3 dot diameter. This function is in effect until cancelled by ESC A, ESC 0, ESC 1 or ESC 2.

NOTE

ESC J is similar in function to ESC 3 but is effective for one line only.

PROGRAM:

```
10 REM ESC 3 - 1/3 DOT LINE SPACING
20 LPRINT CHR$(27);"3";CHR$(60);
30 FOR X = 1 TO 4
40 LPRINT "-----":NEXT X
50 LPRINT CHR$(27);"2";:REM 1/6" LINE SPACE
```

RESULT:

```
-----
-----
-----
-----
```

ESC 4

(Select Alternate *Italic* Characters)

ESC 4

BASIC: CHR\$(27);"4";

HEX: 1B 04

NAME: Select Alternate (Italic) Characters

FUNCTION: When ESC 4 is received, subsequent characters are italicized until cancelled by ESC 5.

NOTE

All characters styles except NLQ (Near Letter Quality) can be italicized.

PROGRAM:

```
10 REM ITALICS
20 PRINT "This is Normal"
30 PRINT CHR$ (27);"4";
40 PRINT "This is Italic"
50 PRINT CHR$ (27);"5";
60 PRINT "Back to Normal"
```

RESULT:

```
This is Normal
This is Italic
Back to Normal
```

ESC 5

(Cancel Alternate *Italic* Characters)

ESC 5

BASIC: CHR\$(27);"5";

HEX: 1B 05

NAME: Cancel Alternate (Italic) Characters

FUNCTION: When ESC 5 is received, italicized characters are cancelled and subsequent characters are printed in the normal style.

PROGRAM: (see above)

ESC 6

(Printable Area Code Expansion)

ESC 6

BASIC: CHR\$(27);"6";

HEX: 1B 36

NAME: Printable Area Code Expansion

FUNCTION: ESC 6 enables the printer to print international characters in the code area of 128 to 159 (plus 255). ESC 6 is cancelled by ESC 7.

PROGRAM:

```
10 REM ESC 6 PRINT AREA EXPANSION
20 LPRINT CHR$(27);"6";CHR$(27);"(";
30 FOR Z= 128 TO 159
40 LPRINT CHR$(Z);" ";;NEXT
50 LPRINT CHR$(255);: LPRINT
60 LPRINT CHR$(27);"7";:REM CANCEL
70 LPRINT CHR$(128);:REM WON'T PRINT
```

RESULT:

à è ù ò ì ° £ ï ï Ñ ñ ð Å å ç § ß æ ø ø " Ä Ö Ü ä ö ü é é ¥ Ø

NOTE: Characters are italicized when in Pica style per this sample.

à è ù ò ì ° £ ï ï Ñ ñ ð Å å ç § ß æ ø ø " Ä Ö Ü ä ö ü é é ¥ Ø

CODE CHART (DECIMAL)

128	129	130	131	132	133	134	135	136	137	138
<i>à</i>	<i>è</i>	<i>ù</i>	<i>ò</i>	<i>ì</i>	<i>°</i>	<i>£</i>	<i>ï</i>	<i>ï</i>	<i>Ñ</i>	<i>ñ</i>
90	81	82	83	84	85	86	87	88	89	8A
139	140	141	142	143	144	145	146	147	148	149
<i>ð</i>	<i>Å</i>	<i>Å</i>	<i>å</i>	<i>ç</i>	<i>§</i>	<i>ß</i>	<i>Æ</i>	<i>æ</i>	<i>ø</i>	<i>ø</i>
8B	8C	8D	8E	8F	90	91	92	93	94	95
150	151	152	153	154	155	156	157	158	159	255
<i>--</i>	<i>Ä</i>	<i>Ö</i>	<i>Ü</i>	<i>ä</i>	<i>ö</i>	<i>ü</i>	<i>é</i>	<i>é</i>	<i>¥</i>	<i>Ø</i>
96	97	98	99	9A	9B	9C	9D	9E	9F	FF

ESC 7

(Cancel Printable Area Code Expansion)

ESC 7

BASIC: CHR\$(27);"7";

HEX: 1B 37

NAME: Cancel Printable Area Code Expansion

FUNCTION: ESC 7 cancels the printing of codes in the 128 to 159 area (plus 255). Control codes such as BEL (7 + 128 = 135) will respond, but not as printable characters.

PROGRAM: (see above)

ESC 8

(Remote Paper End De-Select)

ESC 8

BASIC: CHR\$(27);"8";

HEX: 1B 08

NAME: Paper End Deselect (Remote)

FUNCTION: When ESC 8 is received, the Paper End condition does not result in the printer going OFF LINE as in the normal case. In this manner, printing can continue to within the last few lines of a form. This function is cancelled by ESC 9.

NOTE

DIP switch SW2-1 must be ON in order for this code sequence to perform as described. Also, the slide switch under the tractors must be in the "OTHERS" position.

PROGRAM: LPRINT CHR\$(27);"8";

RESULT: Paper End alarm and OFF LINE condition are bypassed.

ESC 9

(Remote Paper End Select)

ESC 9

BASIC: CHR\$(27);"9";

HEX: 1B 09

NAME: Paper End Select (Remote)

FUNCTION: ESC 9 cancels the Paper End condition bypass set by ESC 8. In this manner, the Paper End switch functions normally, setting the printer OFF LINE and sounding the alarm.

NOTE

DIP switch SW2-1 must be ON in order for this code sequence to perform as described. Also, the slide switch under the tractors must be in the "OTHERS" position.

PROGRAM: LPRINT CHR\$(27);"9";

RESULT: Paper End alarm and OFF LINE condition function normally.

ESC : (Copy ROM CG to Download CG)**ESC :**

BASIC: CHR\$(27);“:”;CHR\$(0);CHR\$(0);

HEX: 1B 3A 00 00 00

NAME: Copy ROM CG to Download CG Set

FUNCTION: ESC : 0 0 0 causes the fonts in the standard ROM Character Generator to be copied into the download area. The basic purpose of this function is to provide most of the standard characters for use with special characters defined by ESC &, which would replace some of the standard characters.

In the program example below, the character “!” is replaced by the Greek letter omega, the symbol for ohms in electronics.

```
PROGRAM: 10 REM ESC : COPY CG ROM TO DOWNLOAD AREA
15 LPRINT CHR$(27);“:”;CHR$(0);CHR$(0);CHR$(0);
20 REM NOW DEFINE OMEGA AS “!”
25 LPRINT CHR$(27);“&”;CHR$(0);“!”;:REM ! = OMEGA
30 LPRINT CHR$(139);:REM ATTRIBUTE FOR OMEGA
35 REM HERE COMES OMEGA
40 LPRINT CHR$(50);CHR$(72);CHR$(6);CHR$(64);
45 LPRINT CHR$(0);CHR$(64);CHR$(6);CHR$(72);
50 LPRINT CHR$(50);CHR$(0);CHR$(0);
55 REM NOW SELECT DOWNLOAD CG
60 LPRINT CHR$(27);“%”;CHR$(1);CHR$(0);
65 FOR X = 33 TO 57
70 LPRINT CHR$(X);: NEXT X: LPRINT
75 REM NOW SELECT ROM CG
80 LPRINT CHR$(27);“%”;CHR$(0);CHR$(0);
85 FOR X = 33 TO 57
90 LPRINT CHR$(X);: NEXT X: LPRINT
95 END
```

RESULT: Ω“#%&’()*+,-./0123456789
!“#%&’()*+,-./0123456789

NOTE

For details on the definition of download characters, their attributes etc. refer to ESC &.

ESC <**(Unidirectional Print, One Line)****ESC <**

BASIC: CHR\$(27);"<";**HEX:** 1B 3C**NAME:** Unidirectional Print, one line only**FUNCTION:** Receipt of ESC < moves the print head to the left-most position and prints subsequent characters left to right for one line, then resumes bi-directional printing.**NOTE**

Unidirectional printing is used when positional accuracy is more important than speed. The line to line positional indexing is best in unidirectional print mode. See ESC U.

```
PROGRAM:      10 REM UNIDIRECTIONAL 1 LINE
                20 PRINT CHR$ (27); "<";
                30 PRINT "UNIDIRECTIONAL -->"
                35 PRINT CHR$ (27); "<";
                40 PRINT "UNIDIRECTIONAL -->"
                50 PRINT "BI-DIRECTIONAL <--"
                60 PRINT "BI-DIRECTIONAL -->"
```

```
RESULT:      UNIDIRECTIONAL -->
                UNIDIRECTIONAL -->
                BI-DIRECTIONAL <--
                BI-DIRECTIONAL -->
```

ESC U**(Unidirectional Print Mode)****ESC U**

BASIC: CHR\$(27);"U";CHR\$(1);**HEX:** 1B 55 01**NAME:** Unidirectional Print Mode**FUNCTION:** Receipt of ESC U 1 moves the print head to the leftmost position and prints subsequent characters left to right, line after line until cancelled by ESC U 0. See note above.

```
PROGRAM:      10 REM UNIDIRECTIONAL MODE
                20 PRINT CHR$ (27); "U"; CHR$ (1);
                30 PRINT "UNIDIRECTIONAL -->"
                40 PRINT "UNIDIRECTIONAL -->"
                50 PRINT CHR$ (27); "U"; CHR$ (0);
                60 PRINT "BI-DIRECTIONAL <--"
                70 PRINT "BI-DIRECTIONAL -->"
```

```
RESULT:      UNIDIRECTIONAL -->
                UNIDIRECTIONAL -->
                BI-DIRECTIONAL <--
                BI-DIRECTIONAL -->
```

ESC =**(Set MSB to 0)****ESC =**

BASIC: CHR\$(27);"=";**HEX:** 1B 3D**NAME:** Set MSB to 0 (force 8th bit to zero)**FUNCTION:** ESC = forces the MSB to a 'zero' condition, thus codes 128 and above are not accessible (do not print). ESC = is changed by ESC > (MSB=1) or cancelled by ESC #.

PROGRAM:

```
10 REM ESC = SET MSB TO 0
20 LPRINT CHR$(27);">";:REM MSB = 1
25 LPRINT "This is italic (MSB = 1)"
30 LPRINT CHR$(27);"=";:REM MSB = 0
35 LPRINT "This is normal (MSB = 0)"
40 LPRINT CHR$(27);"#";:REM CANCEL
```

RESULT:

```
This is italic (MSB = 1)
This is normal (MSB = 0)
```

ESC >**(Set MSB to 1)****ESC >**

BASIC: CHR\$(27);">";**HEX:** 1B 3E**NAME:** Set MSB to 1 (force 8th bit to one)**FUNCTION:** ESC > forces the MSB to a 'one' condition, thus codes lower than 128 are not accessible (do not print). ESC > is changed by ESC = (MSB=0) or cancelled by ESC #.

PROGRAM:

```
10 REM ESC > SET MSB TO 1
20 LPRINT "These are Standard Characters"
40 LPRINT CHR$(27);">";:REM MSB = 1
50 LPRINT "These are Alternate Characters"
60 LPRINT CHR$(27);"#";:REM CANCEL MSB = 1
70 LPRINT "Standard Characters Again!"
```

RESULT:

```
These are Standard Characters
These are Alternate Characters
Standard Characters Again!
```

ESC ?**(Multiple Function Graphics Mode)****ESC ?**

BASIC: CHR\$(27);"?";"n";CHR\$(M);**HEX:** 1B 3F**NAME:** Multiple Function Graphics (n = K, L, Y or Z)(M = 0 to 6)**FUNCTION:** This special code sequence allows the changing of pitch (density) to 1 of 6 other pitches without changing the current program. In addition, this code sequence allows sequences ESC K, ESC L, ESC Y or ESC Z to function as any of the following:

MODE	DENSITY	TOTAL (MAXIMUM) 8" LINE
0	• 60 DOTS/INCH	480 GRAPHIC CHAR/LINE
1	• 120 DOTS/INCH	960 GRAPHIC CHAR/LINE
2	• 60 DOTS/INCH	960 GRAPHIC CHAR/LINE
3	• 240 DOTS/INCH	1920 GRAPHIC CHAR/LINE
4	• 80 DOTS/INCH	640 GRAPHIC CHAR/LINE
5	• 72 DOTS/INCH	576 GRAPHIC CHAR/LINE
6	• 90 DOTS/INCH	720 GRAPHIC CHAR/LINE

PROGRAM:

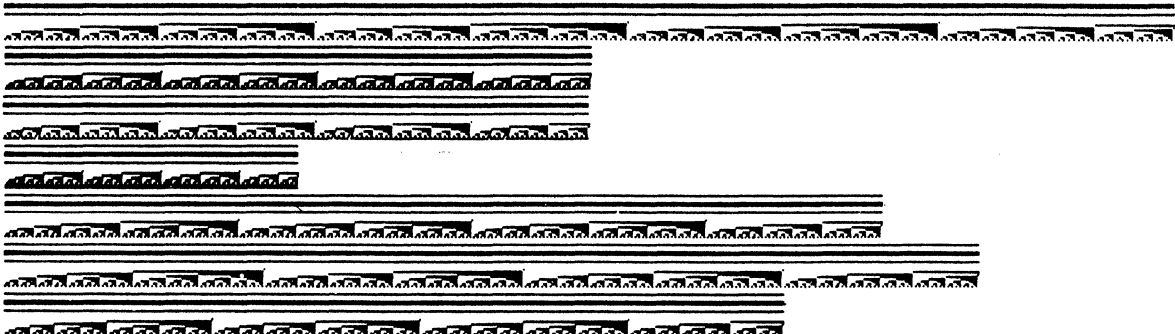
```
10 REM CHANGING PITCH OF GRAPHICS MODE
15 WIDTH "LPT1:",255
20 A = 1 : B = 480 : J = 1
30 IF J = 1 THEN LPRINT "SINGLE DENSITY GRAPHICS MODE" : C$="K"
40 IF J = 2 THEN LPRINT "DOUBLE DENSITY GRAPHICS MODE" : C$="L"
50 IF J = 3 THEN LPRINT "DOUBLE DENSITY, DOUBLE SPEED GRAPHICS MODE" : C$="Y"
60 IF J = 4 THEN LPRINT "QUADRUPLE DENSITY GRAPHICS MODE" : C$="Z"
70 FOR M = 0 TO 6
75 LPRINT CHR$(27);"?" ; C$ ; CHR$(M);
80 LPRINT CHR$(27);C$;CHR$(B MOD 256);CHR$(INT(B/256));
90 FOR K = 1 TO B
100 LPRINT CHR$(&H99);
110 NEXT K
120 LPRINT
125 LPRINT CHR$(27);C$;CHR$(B MOD 256);CHR$(INT(B/256));
130 FOR K = 1 TO B
140 LPRINT CHR$(A);
150 A = A +1
155 IF A > 128 THEN A = 1
160 NEXT K
170 LPRINT
180 A = 1
190 NEXT M
195 J = J + 1
200 IF J < 5 THEN GOTO 30 ELSE LPRINT CHR$(12); : END
```

(Continued on the next page)

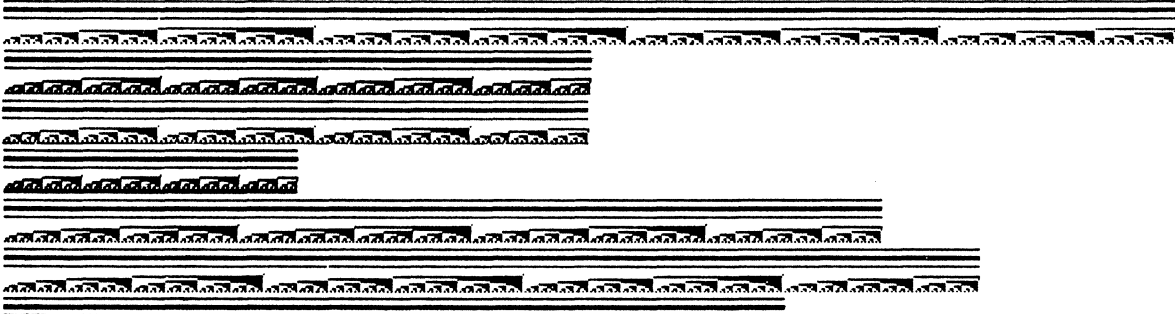
(ESC ? continued)

RESULT:

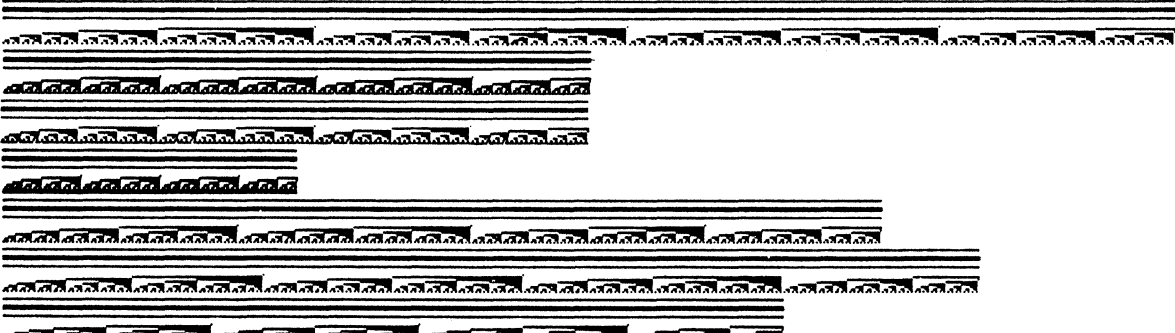
SINGLE DENSITY GRAPHICS MODE



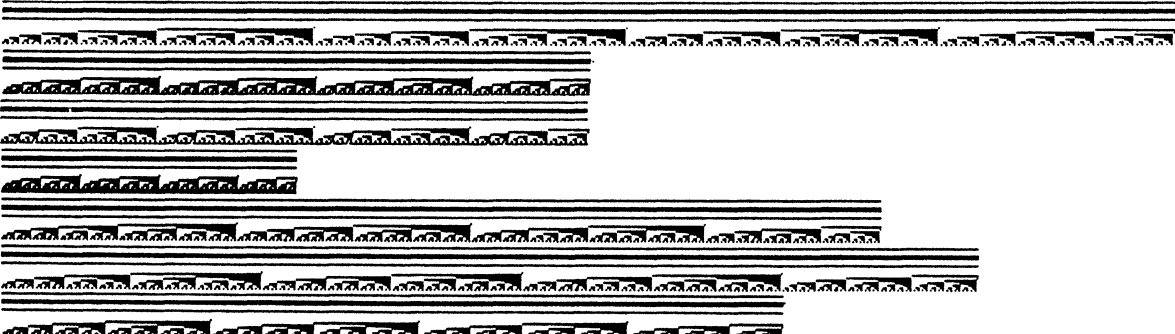
DOUBLE DENSITY GRAPHICS MODE



DOUBLE DENSITY, DOUBLE SPEED GRAPHICS MODE



QUADRUPLE DENSITY GRAPHICS MODE



ESC @**(Printer Initialization)****ESC @**

BASIC: CHR\$(27);"@";**HEX:** 1B 40**NAME:** Printer Initialization**FUNCTION:** ESC @ initializes the printer, clears data in the print buffer and cancels any special print style, line spacing, etc.

```
PROGRAM:      10 REM PRINTER INITIALIZATION
                20 PRINT CHR$ (27);"!"; CHR$ (32);
                30 PRINT "A STITCH IN TIME"
                40 PRINT "SAVES NINE";
                50 PRINT "A ROLLING STONE";
                60 PRINT CHR$ (27);"@";
                70 PRINT CHR$ (27);"!"; CHR$ (32);
                80 PRINT "GATHERS NO MOSS!"
```

RESULT: **A STITCH IN TIME**
 GATHERS NO MOSS!

ESC A**(1 Dot Line Spacing)****ESC A**

BASIC: CHR\$(27);"A";CHR\$(n);**HEX:** 1B 41 (n)**NAME:** One Dot Line Spacing (n = 1 to 60)**FUNCTION:** ESC A (n) establishes the line feed value in one dot increments for fine control of line spacing.**NOTE**

The line spacing value established by this function will also be executed when the LF button is pressed while OFF LINE.

```
PROGRAM:      10 REM N/60 LINE SPACING
                20 FOR X = 2 TO 12 STEP 2
                30 PRINT CHR$ (27);"A"; CHR$ (X);
                40 PRINT "-----"
                50 NEXT X
                60 PRINT "-----"
```

RESULT: =====

 =====

ESC B

(Vertical Tab Setting, 16 Positions)

ESC B

BASIC: CHR\$(27);"B";CHR\$(n)...CHR\$(n,16);CHR\$(0);

HEX: 1B 42 (n) 00

NAME: Vertical Tab Setting (up to 16 positions)

FUNCTION: ESC B establishes the vertical tab stops that are executed sequentially with each VT code received. The tab settings are terminated by CHR\$(0);. This function will track the original top of form and tab stops even if the line spacing is changed between stops, as the original line measure is stored and executed with VT.

NOTE

Tab positions in ESC B must be entered in numerical ascending order and terminated with CHR\$(0);. Also, if no tab positions are established, VT performs a line feed. The maximum tab value is 66 for normal (1/6") line spacing. Larger values default to 66. The largest number accepted is 254.

PROGRAM:

```
10 REM ESC B VERTICAL TAB
15 FOR Z = 0 TO 17
20 LPRINT Z: NEXT
25 FOR X = 1 TO 3
30 LPRINT CHR$(27); "j"; CHR$(180); : NEXT
35 LPRINT CHR$(27); "B"; CHR$(1); CHR$(6);
40 LPRINT CHR$(11); CHR$(17); CHR$(0);
45 LPRINT CHR$(11); "    <--VTAB @ 1"
50 LPRINT CHR$(27); "A"; CHR$(5);
55 FOR I = 1 TO 3
60 LPRINT "    -----": NEXT
65 LPRINT CHR$(11); "    <--VTAB @ 6"
70 LPRINT CHR$(11); "    <--VTAB @ 11"
75 LPRINT CHR$(11); "    <--VTAB @ 17"
80 LPRINT CHR$(27); "A"; CHR$(10);
```

RESULT:

```
0
1 <--VTAB @ 1
2 =====
3 =====
4
5
6 <--VTAB @ 6
7
8
9
10
11 <--VTAB @ 11
12
13
14
15
16
17 <--VTAB @ 17
```

ESC C**(Set Form Length # of Lines)****ESC C**

BASIC: CHR\$(27);"C";CHR\$(n);**HEX:** 1B 43 (n)**NAME:** Set Form Length by # of lines (n = 1 to 127)

FUNCTION: ESC C (n) sets the absolute number of lines that will be fed when a FF (Form Feed) code is received. If the line spacing is the normal 1/6", then 66 lines would be standard Form Length (11"). If the line spacing value is changed by ESC A (n) etc. a FF will still result in feeding paper the **number of lines** established by ESC C (n).

PROGRAM:

```
10 REM ESC C FORM # LINES
20 PRINT CHR$ (27);"C"; CHR$ (6);
30 PRINT "*****";
40 PRINT CHR$ (12);: REM FF
50 PRINT "SIX LINES FORM FEED"
```

RESULT: *****

SIX LINES FORM FEED

ESC C 0**(Set Form Length # of Inches)****ESC C 0**

BASIC: CHR\$(27);"C";CHR\$(n);**HEX:** 1B 43 00 (n)**NAME:** Set Form Length by # of inches (n = 1 to 22)

FUNCTION: ESC C 0 (n) sets the number of inches that will be fed when a FF (Form Feed) code is received. 11 inches is a standard Form Length.

PROGRAM:

```
10 REM ESC C 0 FORM # INCHES
20 PRINT CHR$ (27);"C"; CHR$ (0
); CHR$ (1);
30 PRINT "*****";
40 PRINT CHR$ (12);: REM FF
50 PRINT "ONE INCH FORM FEED"
```

RESULT: *****

ONE INCH FORM FEED

ESC D (Horizontal Tab Setting, 32 Positions)

ESC D

BASIC: CHR\$(27);"D";CHR\$(n)...CHR\$(n,2);CHR\$(0);

HEX: 1B 44 (n) 00

NAME: Horizontal Tab Setting (32 positions maximum)

FUNCTION: ESC D establishes the horizontal tab stops that are executed with each HT code received. Tab settings are entered in ascending numerical order and are terminated with CHR\$(0);. Tab stops can be specified to 80 columns in 10 cpi and 132 in condensed mode.

NOTE

The tab stops are established by the character width at time of tab entry. This function stores these positions and executes proper positioning regardless of type style changes. Tab stops greater than the effective column width are executed at tab position zero on the next line.

```
PROGRAM: 10 REM ESC D HORIZONTAL TAB SETTING @ FICA 10 CPI
15 LPRINT "0123456789012345678901234567890123456789"
20 LPRINT CHR$(27);"D";CHR$(5);CHR$(15);CHR$(25);
25 LPRINT CHR$(30);CHR$(35);CHR$(0);:REM END TABS
30 LPRINT CHR$(27);"W";CHR$(1);:REM ENLARGED
35 FOR Z = 1 TO 2: LPRINT CHR$(9);"TAB";:NEXT Z
40 LPRINT CHR$(27);"W";CHR$(0);:REM CANCEL ENLARGED
45 LPRINT CHR$(15);:REM CONDENSED CHARACTERS
50 LPRINT CHR$(9);"TAB";
55 LPRINT CHR$(18);:REM CANCEL CONDENSED
60 LPRINT CHR$(27);"M";:REM ELITE 12 CPI CHARACTERS
65 LPRINT CHR$(9);"TAB";
70 LPRINT CHR$(27);"P";:REM BACK TO FICA 10 CPI
75 LPRINT CHR$(9);"TAB";:LPRINT :END
```

RESULT: 0123456789012345678901234567890123456789
 TAB TAB TAB TAB TAB

ESC E**(Select Emphasized Print)****ESC E**

BASIC: CHR\$(27);"E";

HEX: 1B 45

NAME: Select Emphasized Print

FUNCTION: When ESC E is received, all subsequent characters are emphasized by firing adjacent pins in one pass of the printhead. Emphasized characters are printed at reduced speed. This function is cancelled by ESC F or ESC !.

PROGRAM:

```
10 REM EMPHASIZED
20 PRINT "(NORMAL) LOOK JANE LOOK!!!"
30 PRINT
40 PRINT "(EMPHASIZED) ";
50 PRINT CHR$ (27);"E";
60 PRINT "SEE SPOT RUN!!!"
70 PRINT : PRINT CHR$ (27);"F";
80 PRINT "(NORMAL) RUN DICK RUN!!!"
```

RESULT:

```
(NORMAL) LOOK JANE LOOK!!!
(EMPHASIZED) SEE SPOT RUN!!!
(NORMAL) RUN DICK RUN!!!
```

ESC F**(Cancel Emphasized Print)****ESC F**

BASIC: CHR\$(27);"F";

HEX: 1B 46

NAME: Cancel Emphasized Print

FUNCTION: When ESC F is received, emphasized print is cancelled.

PROGRAM: (see above)

ESC G**(Double Strike Print Mode)****ESC G**

BASIC: CHR\$(27);"G";**HEX:** 1B 47**NAME:** Double Strike Print Mode

FUNCTION: When ESC G is received, all subsequent characters are printed in double strike. This print style is printed in two passes of the print head, the second pass a fraction of a dot lower. This fills the vertical strokes of the characters to improve quality. This function is cancelled by ESC H.

PROGRAM:

```
10 REM ESC G DOUBLE STRIKE
20 PRINT CHR$ (27);"G";
30 PRINT "DOUBLE STRIKE!"
40 PRINT CHR$ (27);"H";
50 PRINT
60 PRINT "WHILE THE IRON IS HOT"
```

RESULT: **DOUBLE STRIKE!**

WHILE THE IRON IS HOT

ESC H**(Cancel Double Strike Mode)****ESC H**

BASIC: CHR\$(27);"H";**HEX:** 1B 48**NAME:** Cancel Double Strike Mode

FUNCTION: Receipt of ESC H cancels the Double Strike print mode.

PROGRAM: (see above)

ESC I

(Enable Printable Control Codes)

ESC I

BASIC: CHR\$(27);"I";CHR\$(1);

HEX: 1B 49 01

NAME: Enable Printable Control Codes

FUNCTION: ESC I 1 enables the printer to print characters using ASCII codes in the control code area (ASCII 0 to 31 and 128 to 159). Note that standard control codes such as BEL (7 decimal) and CR (13 decimal) do not produce characters, but continue to function as normal. This function is cancelled by ESC I 0.

```

PROGRAM: 10 REM ESC I PRINTABLE CONTROL CODES
            20 LPRINT CHR$(0);CHR$(1);CHR$(128);CHR$(129);
            30 LPRINT "LINE 20 UNPRINTABLE CONTROL CODES"
            40 LPRINT CHR$(27);"I";CHR$(1);:REM OK TO PRINT
            50 LPRINT CHR$(0);CHR$(1);CHR$(128);CHR$(129);
            60 LPRINT " <-- LINE 50 PRINTABLE CONTROL CODES"
            70 LPRINT CHR$(27);"I";CHR$(0);:REM CANCEL
            80 LPRINT CHR$(0);CHR$(1);CHR$(128);CHR$(129);
            90 LPRINT "LINE 80 UNPRINTABLE CONTROL CODES"
            95 END

```

RESULT: LINE 20 UNPRINTABLE CONTROL CODES
àèàè <-- LINE 50 PRINTABLE CONTROL CODES
LINE 80 UNPRINTABLE CONTROL CODES

PRINTABLE CODES WITHIN CONTROL CODE AREA

DEC/CHAR	DEC/CHAR	DEC/CHAR	DEC/CHAR	DEC/CHAR	DEC/CHAR	DEC/CHAR	DEC/CHAR
0 à	8 BS	16 ß	24	128 à	136 BS	144 ß	152 ò
1 é	9 HT	17 ß	25 ù	129 é	137 HT	145 ß	153 ù
2 ù	10 LF	18 DC2	26 à	130 ù	138 LF	146 DC2	154 à
3 ò	11 VT	19 DC3	27 ESC	131 ò	139 VT	147 DC3	155 ESC
4 ì	12 FF	20 DC4	28 ù	132 ð	140 FF	148 DC4	156 ù
5 º	13 CR	21 ø	29 é	133 º	141 CR	149 ø	157 é
6 ₣	14 SO	22 ..	30 é	134 ₣	142 SO	150 ..	158 é
7 BEL	15 SI	23 Ä	31 ₣	135 BEL	143 SI	151 Ä	159 ₣

ESC J**(Paper Feed, 1/3 Dot Increment)****ESC J**

BASIC: CHR\$(27)"J";CHR\$(n);**HEX:** 1B 4A (n)**NAME:** Paper Feed Execution, 1/3 dot increment (n = 1 to 180)**FUNCTION:** ESC J (n) causes all data in the print buffer to be printed, then paper is fed the amount indicated by (n). ESC J (n) is cancelled by a line feed.

```
PROGRAM:      10 REM ESC J PAPER FEED 1/3 DOT
                20 LPRINT "-----"
                30 FOR X = 1 TO 20
                40 LPRINT CHR$(8);:NEXT
                50 LPRINT CHR$(27);"J";CHR$(90);
                60 LPRINT "-----"
                70 LPRINT "-----"
```

RESULT: -----

ESC j**(Reverse Paper Feed, 1/3 Dot Increment)****ESC j**

BASIC: CHR\$(27)"j";CHR\$(n);**HEX:** 1B 6A (n)**NAME:** Reverse Paper Feed Execution, 1/3 dot increment (n = 1 to 180)**FUNCTION:** ESC j (n) causes all data in the print buffer to be printed, then paper is fed (in reverse) the amount indicated by (n). ESC j (n) is cancelled by a line feed.

```
PROGRAM:      10 REM ESC j REVERSE PAPER 1/3 DOT
                20 FOR X = 6 TO 1 STEP -1
                40 LPRINT X:NEXT X
                45 LPRINT " 0 (REF)";
                50 LPRINT CHR$(27);"j";CHR$(180);
                60 LPRINT "<----"
```

RESULT: 6 <----
5
4
3
2
1
0 (REF)

BASIC: CHR\$(27);"K";CHR\$(N1);CHR\$(N2);**HEX:** 1B 4B**NAME:** Graphics Mode—Single Density

FUNCTION: When the above code sequence is received, the printer is placed into Single Density Graphics Mode and waits to receive the total number of graphic characters in the line designated by (N1) and (N2).

(N1) and (N2) are defined as follows:

(N2) is a whole number = total number of graphic characters ÷ 256

(N1) is the remainder of the above equation

Example: total number of graphic characters = 750

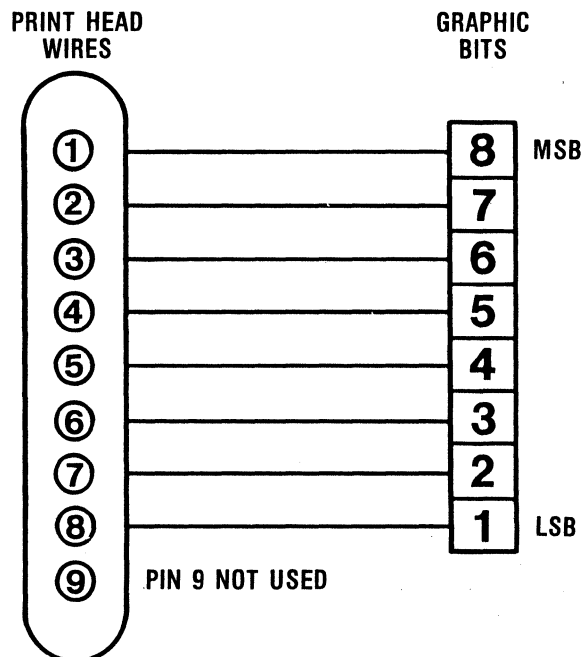
$$N2 = 750 \div 256 = 2$$

$$N1 = 750 - (256 \times N2) = 238$$

In Single Density Graphics the maximum number of graphic characters that can be printed per line is 480 (60 dots/inch). If the number of graphic characters exceeds this maximum, the excess graphic characters are ignored.

Graphic Character—A graphic character corresponds on a one-to-one basis with the print head wires as illustrated. This relationship is sometimes referred to as "bit image" graphics.

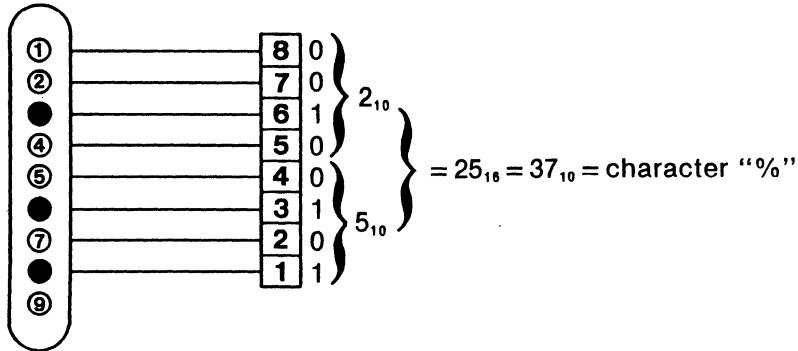
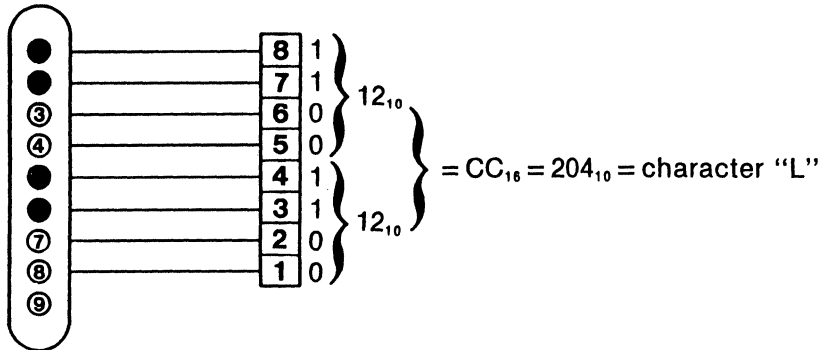
The ninth print head wire is used for underline and nine pin graphics only.



(Continued on the next page)

(ESC K continued)

The following are examples:



PROGRAM:

```

10 REM SINGLE DENSITY GRAPHICS MODE
15 WIDTH "LPT1:",255
20 A = 1 : B = 320
30 FOR I = 1 TO 8
40 LPRINT "START";
50 LPRINT CHR$(27);"K";CHR$(B MOD 256);CHR$(INT(B/256));
60 FOR J = 1 TO B
70 LPRINT CHR$(A);
80 NEXT J
90 A = A * 2
100 LPRINT "END"
110 NEXT I
120 LPRINT CHR$(12); : END

```

RESULT:

```

START_____END
START_____END
START_____END
START_____END
START_____END
START_____END
START_____END
START_____END

```

ESC M

(Select Elite Characters)

ESC M

BASIC: CHR\$(27);"M";

HEX: 1B 4D

NAME: Select Elite (12 cpi) Characters

FUNCTION: ESC M selects elite characters which are 12 characters per inch. This function is cancelled by ESC P, ESC ! and ESC (.

PROGRAM:

```
10 REM ESC M ELITE CHARACTERS
20 PRINT "PICA 10 CPI-20 H'S"
30 PRINT "HHHHHHHHHHHHHHHHHHHHHHHH"
40 PRINT CHR$ (27);"M";
50 PRINT "ELITE 12 CPI-20 H'S"
RESULT: 60 PRINT "HHHHHHHHHHHHHHHHHHHHHHHH"
70 PRINT CHR$ (27);"P";
80 PRINT "PICA 10 CPI-20 H'S"
90 PRINT "HHHHHHHHHHHHHHHHHHHHHHHH"
```

```
PICA 10 CPI-20 H'S
HHHHHHHHHHHHHHHHHHHHHHHH
ELITE 12 CPI-20 H'S
HHHHHHHHHHHHHHHHHHHHHHHH
PICA 10 CPI-20 H'S
HHHHHHHHHHHHHHHHHHHHHHHH
```

ESC P

(Select Pica Characters)

ESC P

BASIC: CHR\$(27);"P";

HEX: 1B 50

NAME: Select Pica (12 cpi) characters

FUNCTION: ESC P selects standard pica characters, which are 10 characters per inch. ESC P can be used to cancel Elite, NLQ, and Condensed characters.

PROGRAM: (see above)

ESC N (Enable Skip-Over Perforation Setting) ESC N

BASIC: CHR\$(27);"N";CHR\$(n);

HEX: 1B 4E (n)

NAME: Enable Skip-Over Perforation Setting (n = 1 to 127)

FUNCTION: ESC N (n) specifies the number of lines to be skipped at the bottom of the fanfold page. In this example, the form length is set at 8 lines by ESC C (8) and the skip-over is set at 3 lines by ESC N (3). Thus, the skip-over begins after line 5 (8 - 3 = 5). This function is in effect for each form length until cancelled by ESC O. Note that if DIP switch 3-3 is ON, the skip-over will always be 1 inch (6 lines at 1/6 inch) for a standard form length (11" or 12") with DIP switch 3-3 ON.

PROGRAM: 10 REM ESC N SKIP OVER PERF
 20 FOR X= 1 TO 18: LPRINT X:NEXT
 30 FOR Z = 1 TO 3
 40 LPRINT CHR\$(27);"j";CHR\$(180);:NEXT
 50 LPRINT CHR\$(27);"C"CHR\$(8);:REM FORM LENGTH
 60 LPRINT CHR\$(27);"N";CHR\$(3);:REM SKIP 3
 70 FOR Z = 1 TO 6
 80 LPRINT " SKIP OVER ";Z:NEXT
 90 LPRINT CHR\$(27);"O";:REM CANCEL SKIP OVER
 100 FOR I = 1 TO 9
 110 LPRINT " NO SKIP OVER ";I:NEXT

RESULT: 1 SKIP OVER 1
 2 SKIP OVER 2
 3 SKIP OVER 3
 4 SKIP OVER 4
 5 SKIP OVER 5
 6
 7
 8
 9 SKIP OVER 6
 10 NO SKIP OVER 1
 11 NO SKIP OVER 2
 12 NO SKIP OVER 3
 13 NO SKIP OVER 4
 14 NO SKIP OVER 5
 15 NO SKIP OVER 6
 16 NO SKIP OVER 7
 17 NO SKIP OVER 8
 18 NO SKIP OVER 9

ESC O (Cancel Skip-Over Perforation Setting) ESC O

BASIC: CHR\$(27);"O";

HEX: 1B 4F

FUNCTION: ESC O cancels the skip-over perforation setting established by ESC N (n).

PROGRAM: (see above)

ESC Q**(Right Margin Setting)****ESC Q**

BASIC: CHR\$(27);"Q";CHR\$(n);

HEX: 1B 51 (n)

NAME: Right Margin Setting (n = 1 to 255)

FUNCTION: ESC Q (n) establishes the column position that will end the line at the right margin, and is entered before any characters for that line. The maximum value for (n) is 132 in condensed mode, 80 in 10 cpi and 40 in enlarged mode. Excessive value for (n) is ignored and previous setting is in effect.

PROGRAM:

```
10 REM ESC Q RIGHT MARGIN SETTING
20 LPRINT "1234567890123456789012345678901234567890"
30 LPRINT CHR$(27);"Q";CHR$(20);
40 LPRINT "1234567890123456789012345678901234567890"
50 LPRINT CHR$(27);"Q";CHR$(40);
60 LPRINT "1234567890123456789012345678901234567890"
```

RESULT:

```
1234567890123456789012345678901234567890
12345678901234567890
12345678901234567890
1234567890123456789012345678901234567890
```

ESC I**(Left Margin Setting)****ESC I**

BASIC: CHR\$(27);"I";CHR\$(n);

HEX: 1B 6C (n)

NAME: Left Margin Setting (n = 0 to 255)

FUNCTION: ESC I (n) establishes the column position to start the line at the left margin, and is entered before any characters for that line. The character size in effect at the time establishes the left margin column position. Column zero (0) is the normal first column.

PROGRAM:

```
10 REM ESC I LEFT MARGIN SETTING
15 LPRINT "0123456789"
20 FOR Z = 1 TO 5
30 LPRINT CHR$(27);"I";CHR$(Z);"MOVE -->"
40 NEXT Z: LPRINT CHR$(27);"I";CHR$(0);
```

RESULT:

```
0123456789
MOVE -->
MOVE -->
MOVE -->
MOVE -->
MOVE -->
```

ESC R (International Character Set Mode)

ESC R

BASIC: CHR\$(27);"R";CHR\$(n);

HEX: 1B 52 (n)

NAME: International Character Set Mode (n=0 to 8)

FUNCTION: ESC R (n) selects one of nine international character sets. Refer to table on next page for decimal codes.

PROGRAM:

```
10 REM ESC R INTERNATIONAL CHARACTERS
15 LPRINT CHR$(27);"(";
20 X = 0:REM USA
25 LPRINT CHR$(27);"R";CHR$(X);
30 LPRINT "MODE ";X;" ";
35 X = X + 1
40 FOR Z = 33 TO 79:LPRINT CHR$(Z);
45 NEXT Z:LPRINT
50 LPRINT "          ";
55 FOR J = 80 TO 126: LPRINT CHR$(J);
60 NEXT J:LPRINT : LPRINT
65 IF X = 9 THEN END
70 GOTO 25
```

RESULTS:

MODE 0	!"#\$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNO PQRSTUVWXYZ[\]^_`abcdefghijklmnopqrstuvwxyz{ }~
MODE 1	!"#\$%&'()*+,-./0123456789:;<=>?àABCDEFGHIJKLMNO PQRSTUVWXYZ`ç\$^_`abcdefghijklmnopqrstuvwxyzéùè"
MODE 2	!"#\$%&'()*+,-./0123456789:;<=>?§ABCDEFGHIJKLMNO PQRSTUVWXYZÄÖÛ^_`abcdefghijklmnopqrstuvwxyzÿöüß
MODE 3	!"£\$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNO PQRSTUVWXYZ[\]^_`abcdefghijklmnopqrstuvwxyz{ }~
MODE 4	!"#\$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNO PQRSTUVWXYZÆØÅ^_`abcdefghijklmnopqrstuvwxyzæøå~
MODE 5	!"#¤%&'()*+,-./0123456789:;<=>?ÉABCDEFGHIJKLMNO PQRSTUVWXYZĂŎĂŮ_éabcdefghijklmnopqrstuvwxyzÿöäü
MODE 6	!"#\$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNO PQRSTUVWXYZ`è^_ùabcdefghijklmnopqrstuvwxyzàèèi
MODE 7	!"r\$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNO PQRSTUVWXYZ;ñ¿^_`abcdefghijklmnopqrstuvwxyz"ñ}~
MODE 8	!"#\$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNO PQRSTUVWXYZ[¥]^_`abcdefghijklmnopqrstuvwxyz{ }~

(See chart, next page)

(ESC R continued)

INTERNATIONAL CHARACTERS

	0	1	2	3	4	5	6	7	8
DEC	U.S.A.	FRANCE	GERMANY	U.K.	DENMARK	SWEDEN	ITALY	SPAIN	JAPAN
35	#	#	#	£	#	#	#	Pt	#
36	\$	\$	\$	\$	\$	¤	\$	\$	\$
64	@	à	§	@	@	É	@	@	@
91	[°	Ä	[Æ	Å	°	ı	[
92	\	ø	Ö	\	Ø	Ö	\	Ñ	¥
93]	§	Ü]	À	Å	é	¿]
94	^	^	^	^	^	ü	^	^	^
96	`	`	`	`	`	é	ù	`	`
123	{	é	ä	{	æ	ä	à	¨	{
124		ù	ö		ø	ö	ò	ñ	
125	}	è	ü	}	ä	ä	è	}	}
126	~	¨	ß	~	~	ü	ı	~	~

ESC S 0 (Select Superscript Characters)

ESC S 0

BASIC: CHR\$(27);"S";CHR\$(0);

HEX: 1B 53 00

NAME: Select Superscript Characters

FUNCTION: ESC S 0 selects superscript characters, which are half-size and printed in two passes of the print head. This function is cancelled by ESC T.

```
PROGRAM: 10 REM ESC S0 SUPERSCRIP
          20 PRINT "HERE IS SUPERSCRIP";
          30 PRINT CHR$ (27);"S"; CHR$ (0);
          40 PRINT "1234567890"
          50 PRINT : PRINT CHR$ (27);"T";
          60 PRINT "EINSTEIN'S FAMOUS FORM
              ULA: ";
          70 PRINT "E=MC";
          80 PRINT CHR$ (27);"S"; CHR$ (0);
          90 PRINT "2";: REM SUPERSCRIP
          95 PRINT CHR$ (27);"T";
```

RESULT: HERE IS SUPERSCRIP¹²³⁴⁵⁶⁷⁸⁹⁰
EINSTEIN'S FAMOUS FORMULA: E=MC²

ESC S 1 (Select Subscript Characters)

ESC S 1

BASIC: CHR\$(27);"S";CHR\$(1);

HEX: 1B 53 01

NAME: Select Subscript Characters

FUNCTION: ESC S 1 selects subscript characters, which are half-size and printed in two passes of the print head. This function is cancelled by ESC T.

```
PROGRAM: 10 REM ESC S1 SUBSCRIPT
          20 PRINT "HERE IS SUBSCRIPT";
          30 PRINT CHR$ (27);"S"; CHR$ (1);
          40 PRINT "1234567890"
          50 PRINT : PRINT CHR$ (27);"T";
          60 PRINT "FORMULA FOR WATER = H";
          70 PRINT CHR$ (27);"S"; CHR$ (1);
          80 PRINT "2"; CHR$ (27);"T";"O"
```

RESULT: HERE IS SUBSCRIPT₁₂₃₄₅₆₇₈₉₀
FORMULA FOR WATER = H₂O

ESC T

(Cancel Super/Sub Script)

ESC T

BASIC: CHR\$(27);"T";

HEX: 1B 54

NAME: Cancel Superscript/Subscript

FUNCTION: ESC T cancels superscript (ESC S 0) and subscript (ESC S 1) characters.

PROGRAM: 10 REM ESC T CANCEL SUPER/SUBSCRIPT
 20 LPRINT "THIS IS SUPER";
 30 LPRINT CHR\$(27);"S";CHR\$(0);
 40 LPRINT "SCRIPT";CHR\$(27);"T"
 50 LPRINT "NORMAL PICA STYLE"
 60 LPRINT "THIS IS SUB";
 70 LPRINT CHR\$(27);"S";CHR\$(1);
 80 LPRINT "SCRIPT";CHR\$(27);"T"
 90 LPRINT "NORMAL PICA STYLE"

RESULT: THIS IS SUPER^{SCRIPT}
 NORMAL PICA STYLE
 THIS IS SUB_{SCRIPT}
 NORMAL PICA STYLE

ESC W

(Enlarged Character Mode)

ESC W

BASIC: CHR\$(27);"W";CHR\$(1);

HEX: 1B 57 01

NAME: Enlarged Character Mode

FUNCTION: ESC W 1 causes any print style to be enlarged and is in effect until cancelled by ESC W 0 or ESC !.

PROGRAM: 10 REM ESC W ENLARGED PRINT MODE
 20 LPRINT CHR\$(27);"(";:REM NLQ
 30 LPRINT CHR\$(27);"W";CHR\$(1);
 40 LPRINT "THIS IS ENLARGED NLQ"
 50 LPRINT "AND SO IS THIS LINE!"
 60 LPRINT CHR\$(27);"W";CHR\$(0);
 70 LPRINT "NOW WE ARE PRINTING NLQ AT 10 CPI"

RESULT: THIS IS ENLARGED NLQ
 AND SO IS THIS LINE!
 NOW WE ARE PRINTING NLQ AT 10 CPI

ESC Y (Graphics Mode—Double Speed, Double Density) ESC Y

BASIC: CHR\$(27);"Y";CHR\$(N1);CHR\$(N2);

HEX: 1B 59

NAME: Graphics Mode—Double Speed, Double Density

FUNCTION: When the above code sequence is received, the printer is placed into Double Speed, Double Density Graphics Mode and waits to receive the total number of graphic characters designated by (N1) and (N2). (Refer to ESC K for definition of (N1) and (N2) values).

In Double Speed, Double Density Graphics Mode the total number of graphic characters that can be printed per line is 480 (60 Dots/in). The total number of graphic characters that can be sent to the printer in this mode is 960. In this mode adjacent dot columns are not printed, which increases printing time. If the number of graphic characters to be received exceed the 960 character maximum the excess are ignored.

Graphic Characters (Refer to ESC K)

PROGRAM:

```
10 REM DOUBLE DENSITY GRAPHICS MODE
15 WIDTH "LPT1:",255
18 LPRINT "DOUBLE DENSITY GRAPHICS MODE"
19 L$="L"
20 FOR I = 1 TO 4
30 A = 1 : B = 700
40 LPRINT "START";
50 LPRINT CHR$(27);L$;CHR$(B MOD 256);CHR$(INT(B/256));
60 FOR J = 1 TO B
70 LPRINT CHR$(A);
80 A = A + 1
90 IF A > 128 THEN A = 1
95 NEXT J
100 LPRINT "END"
110 NEXT I
120 IF L$="Y" GOTO 160
130 IF L$="L" THEN L$="Y"
140 IF L$="Y" THEN LPRINT "DOUBLE SPEED, DOUBLE DENSITY GRAPHICS MODE"
150 IF L$="Y" THEN GOTO 20
160 LPRINT CHR$(12); : END
```

RESULT:

```
DOUBLE DENSITY GRAPHICS MODE
START ██████████END
START ██████████END
START ██████████END
START ██████████END
DOUBLE SPEED, DOUBLE DENSITY GRAPHICS MODE
START ██████████END
START ██████████END
START ██████████END
START ██████████END
```

BASIC: CHR\$(27);"Z";CHR\$(N1);CHR\$(N2);

HEX: 1B 5A

NAME: Graphics Mode—Quadruple Density**FUNCTION:** When the above code sequence is received, the printer is placed into Quadruple Density Graphics Mode and waits to receive the total number of graphics characters for the line designated by (N1) and (N2). (Refer to ESC K for definition of (N1) and (N2) values).

In Quadruple Density Graphics Mode the maximum number of graphic characters per line is 1920 (240 Dots/in). If the number of graphic characters to be received exceeds this maximum, the excess is ignored.

Graphic Characters (Refer to ESC K)

PROGRAM:

```
10 REM QUADRUPLE DENSITY GRAPHICS MODE
15 WIDTH "LPT1:",255
20 A = 1 : B = 700
30 FOR J = 1 TO 4
40 IF J = 1 THEN LPRINT "SINGLE DENSITY GRAPHICS MODE" : C$="K"
50 IF J = 2 THEN LPRINT "DOUBLE DENSITY GRAPHICS MODE" : C$="L"
60 IF J = 3 THEN LPRINT "DOUBLE DENSITY, DOUBLE SPEED GRAPHICS MODE" : C$="Y"
70 IF J = 4 THEN LPRINT "QUADRUPLE DENSITY GRAPHICS MODE" : C$="Z"
80 LPRINT CHR$(27);C$;CHR$(B MOD 256);CHR$(INT(B/256));
90 FOR K = 1 TO B
100 LPRINT CHR$(&H99);
110 NEXT K
120 LPRINT
125 LPRINT CHR$(27);C$;CHR$(B MOD 256);CHR$(INT(B/256));
130 FOR K = 1 TO B
140 LPRINT CHR$(A);
150 A = A +1
155 IF A > 128 THEN A = 1
160 NEXT K
170 LPRINT
180 A = 1
190 NEXT J
200 LPRINT CHR$(12); : END
```

RESULT:

SINGLE DENSITY GRAPHICS MODE

DOUBLE DENSITY GRAPHICS MODE

DOUBLE DENSITY, DOUBLE SPEED GRAPHICS MODE

QUADRUPLE DENSITY GRAPHICS MODE

ESC ^**(Graphics Mode—9 Dot and 16 Dot)****ESC ^**

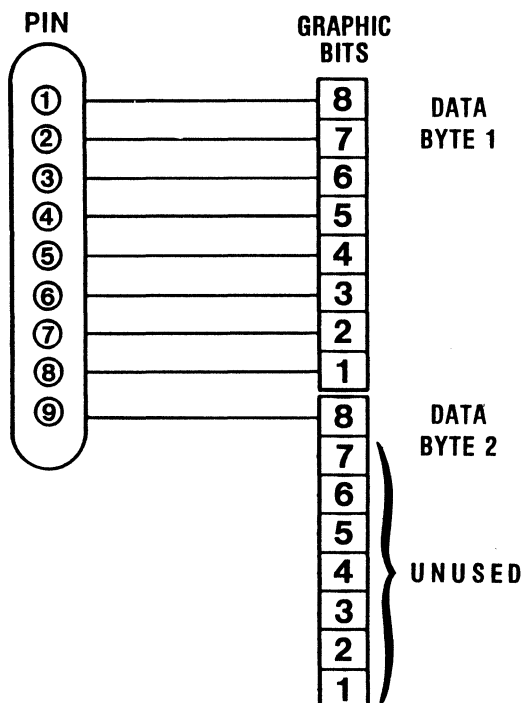
BASIC: CHR\$(27);“^”;CHR\$(M);CHR\$(N1);CHR\$(N2);**HEX:** 1B 5E**NAME:** Graphic Mode (9 Dot and 16 Dot Graphics)**FUNCTION:** When the above code sequence is received, the printer enters 9 dot or 16 dot graphic mode and waits to receive the total number of graphic characters for the line designated by (N1) and (N2). Refer to ESC K for the definition of (N1) and (N2) values. The value of (M) is defined as follows.

M	MODE
0	9 Dot Graphics, 480 Dots/Line, Single Density
1	9 Dot Graphics, 960 Dots/Line, Double Density
17	16 Dot Graphics, 960 Dots/Line, Double Density

If the number of graphic characters exceeds the above maximum, the excess graphic characters are ignored.

Data Byte Structure—In 9 Dot and 16 Dot Graphic Modes, each graphic character consists of two 8-bit data bytes as illustrated.

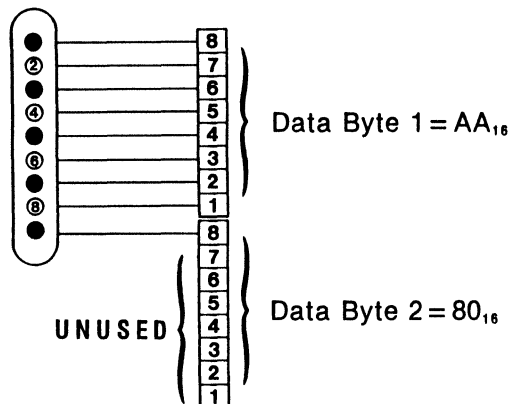
9 Dot Graphics—Pin to Data Bit Relationship



(Continued on the next page)

(ESC ^ continued)

Example



PROGRAM:

```
10 REM 9-DOT GRAPHICS MODE
20 A = 0 : B = 400
25 LPRINT "9-DOT GRAPHICS, SINGLE DENSITY
30 FOR J = 1 TO 4
40 LPRINT "START";
50 LPRINT CHR$(27);"~";CHR$(A);CHR$(B MOD 256);CHR$(INT(B/256));
60 FOR I = 1 TO B
70 LPRINT CHR$(&HAA);CHR$(&H80);
80 NEXT I
81 LPRINT "END"
85 LPRINT : NEXT J
90 IF A = 1 THEN LPRINT CHR$(12); : END
100 A = 1 : B = 700
110 LPRINT "9-DOT GRAPHICS, DOUBLE DENSITY"
120 GOTO 30
```

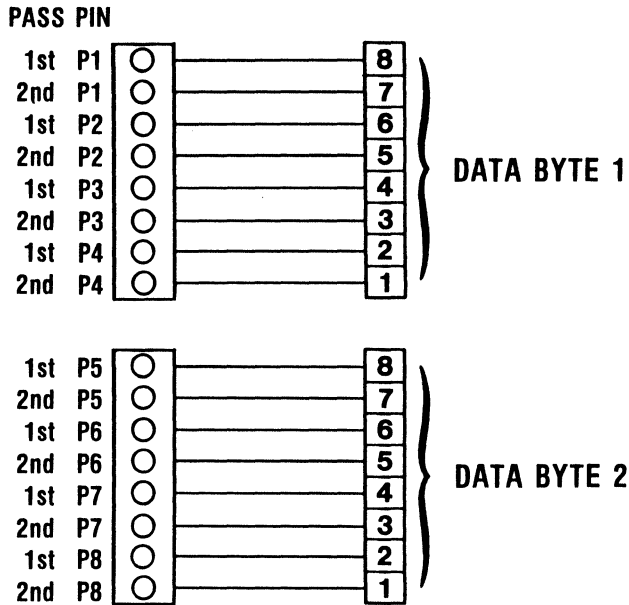
RESULT:

```
9-DOT GRAPHICS, SINGLE DENSITY
START=====END
START=====END
START=====END
START=====END
9-DOT GRAPHICS, DOUBLE DENSITY
START=====END
START=====END
START=====END
START=====END
```

(Continued on the next page)

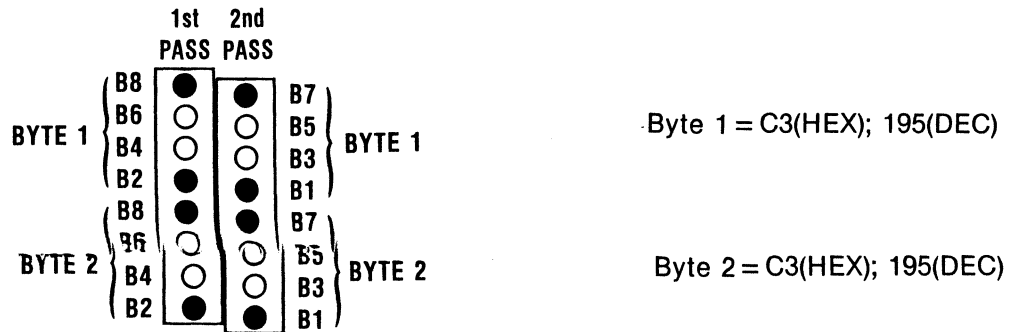
(ESC ^ continued)

16 Dot Graphics Pin to Data Bit Relationship



Note—16 Dot Graphics is a two-pass function with pin firing as shown above. Between first and second pass there is a 1/3 dot line feed then a 1/3 dot reverse line feed after the second pass.

Example



(Continued on the next page)

(ESC ^ continued)

PROGRAM:

```
10 REM 9-DOT GRAPHICS MODE
20 A = 0 : B = 400
25 LPRINT "9-DOT GRAPHICS, SINGLE DENSITY
30 FOR J = 1 TO 4
40 LPRINT "START";
50 LPRINT CHR$(27);"^";CHR$(A);CHR$(B MOD 256);CHR$(INT(B/256));
60 FOR I = 1 TO B
70 LPRINT CHR$(&HAA);CHR$(&H80);
80 NEXT I
81 LPRINT "END"
85 LPRINT : NEXT J
90 IF A = 1 THEN GOTO 130
100 A = 1 : B = 700
110 LPRINT "9-DOT GRAPHICS, DOUBLE DENSITY"
120 GOTO 30
130 A = 17 : B = 800
140 LPRINT "16-DOT GRAPHICS, DOUBLE DENSITY"
150 FOR J = 1 TO 4
160 LPRINT "START";
170 LPRINT CHR$(27);"^";CHR$(A);CHR$(B MOD 256);CHR$(INT(B/256));
180 FOR I = 1 TO B
190 LPRINT CHR$(&H90);CHR$(&H9);
200 NEXT I
210 LPRINT "END"
220 LPRINT : NEXT J
230 LPRINT CHR$(12); : END
```

RESULT:

```
9-DOT GRAPHICS, SINGLE DENSITY
START=====END
START=====END
START=====END
START=====END

9-DOT GRAPHICS, DOUBLE DENSITY
START=====END
START=====END
START=====END
START=====END

16-DOT GRAPHICS, DOUBLE DENSITY
START=====END
START=====END
START=====END
START=====END
```

ESC b**(VFU Tab Position Setting)****ESC b**

BASIC: CHR\$(27);"b";CHR\$(n);CHR\$(m₁);...CHR\$(m₁₆);CHR\$(0)

HEX: 1B 62

NAME: VFU Tab Position Setting (n=0 to 7) (m=1 to 16)**FUNCTION:** ESC b (n) (m) specifies the vertical tab stops (m₁ to m₁₆) within one of eight channels (n₀ to n₇). ESC / (n) selects the channel and VT activates the tab stops in sequence. See ESC / for additional programming.

```
PROGRAM: 10 REM ESC b VFU POSITION SETTING
15 LPRINT CHR$(27);"C";CHR$(10);:REM FORM LENGTH 10
20 FOR Z = 0 TO 23: LPRINT Z:NEXT
25 FOR X = 1 TO 4: LPRINT CHR$(27);"j";CHR$(180);
30 NEXT X: LPRINT " <--- TOP OF FORM
35 LPRINT CHR$(27);"b";CHR$(3);:REM CHANNEL 3
40 LPRINT CHR$(3);CHR$(7);CHR$(9);CHR$(0);:REM CH3 TABS
45 LPRINT CHR$(27);"/";CHR$(3);:REM SELECT CH3
50 LPRINT CHR$(12);:REM TOP OF FORM COMMAND
55 LPRINT " <--- CHAN 3 TOP OF FORM"
60 FOR Y = 1 TO 3: LPRINT CHR$(11);:REM VERT TAB
65 LPRINT " <--- CHAN 3; TAB ";Y:NEXT Y
70 END
```

```
RESULT: 0 <--- TOP OF FORM
1
2
3
4
5
6
7
8
9
10 <--- CHAN 3 TOP OF FORM
11
12
13 <--- CHAN 3; TAB 1
14
15
16
17 <--- CHAN 3; TAB 2
18
19 <--- CHAN 3; TAB 3
20
21
22
23
```

ESC i

(Incremental Typewriter Mode)

ESC i

BASIC: CHR\$(27);"i";CHR\$(1);

HEX: 1B 69 01

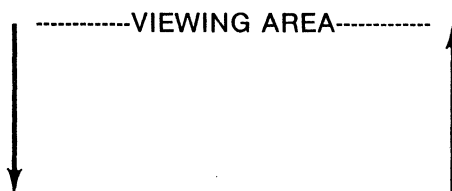
NAME: Incremental (Typewriter) Mode

FUNCTION: ESC i 1 sets the typewriter mode. The function is similar to the old tele-type printer where a view of the printed character is provided by reverse-feeding the paper after printing. The program below requires that a CR (Carriage Return) be sent whenever printing and viewing is desired. To exit the program, send "XXX (CR)". This mode is cancelled by ESC i 0.

NOTE

Some computers ignore trailing spaces and lone spaces sent in this mode. If a space is to be recognized, there must be a character before and after the space prior to the CR.

```
PROGRAM: 10 REM ESC i TYPEWRITER MODE
           20 LPRINT CHR$(27);"i";CHR$(1);
           30 INPUT A$:IF A#="XXX" THEN 50
           40 LPRINT A#;:GOTO 30
           50 LPRINT CHR$(27);"i";CHR$(0);:END
```



RESULT: THIS IS TYPEWRITER MODE.

ESC p**(Proportional Spacing Mode)****ESC p**

BASIC: CHR\$(27);"p";CHR\$(1)**HEX:** 1B 70 01**NAME:** Proportional Spacing Print Mode

FUNCTION: ESC p 1 causes subsequent printing to be spaced in proportion to the width of the characters; that is, they are not monospaced at 10 cpi. This results in more characters per line. Also, proportional printing is always emphasized to improve the quality of print. Proportional spacing is cancelled by ESC p 0. The BS and DEL functions are not valid with this function.

PROGRAM:

```
10 REM ESC p PROPORTIONAL PRINT
20 LPRINT "NORMAL PRINT"
30 LPRINT "ABCDEFGHIJKLMNOPQRSTUVWXYZ"
40 LPRINT "abcdefghijklmnopqrstuvwxyz"
50 LPRINT CHR$(27);"p";CHR$(1);
60 LPRINT : LPRINT "PROPORTIONAL"
70 LPRINT "ABCDEFGHIJKLMNOPQRSTUVWXYZ"
80 LPRINT "abcdefghijklmnopqrstuvwxyz"
90 LPRINT CHR$(27);"p";CHR$(0);
```

RESULT:

```
NORMAL PRINT
ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz

PROPORTIONAL
ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz
```

ESC s**(Half Speed Print Mode)****ESC s**

BASIC: CHR\$(27);"s";CHR\$(1);**HEX:** 1B 73 01**NAME:** Half-Speed (Quiet) Print Mode

FUNCTION: ESC s 1 causes the printer to operate at half normal speed, resulting in quieter operation. This function is cancelled by ESC s 0.

PROGRAM:

```
10 REM ESC s HALF SPEED MODE
20 LPRINT "THIS IS NORMAL SPEED"
30 LPRINT CHR$(27);"s";CHR$(1);
40 LPRINT "THIS IS HALF SPEED"
50 LPRINT CHR$(27);"s"CHR$(0);
60 LPRINT "BACK TO NORMAL SPEED"
```

RESULT:

```
THIS IS NORMAL SPEED
THIS IS HALF SPEED
BACK TO NORMAL SPEED
```