2627A

Color Graphics Terminal

HEWLETT PACKARD

FEDERAL COMMUNICATIONS COMMISSION
RADIO FREQUENCY INTERFERENCE STATEMENT

The Federal Communications Commission (in 47 CFR 15.818) has specified that the following notice be brought to the attention of the users of this product.

This equipment generates, uses, and can radiate radio frequency energy and if not installed and used in accordance with the instructions manual, may cause interference to radio communications. It has been tested and found to comply with the limits for a Class A computing device pursuant to Subpart J of Part 15 of FCC Rules, which are designed to provide reasonable protection against such interference when operated in a commercial environment. Operation of this equipment in a residential area is likely to cause interference in which case the user at his own expense will be required to take whatever measures may be required to correct the interference.

X-RAY RADIATION NOTICE
Model 2627A

WARNING

When operating, this terminal emits x-rays; however, it is well shielded and meets safety and health requirements of various countries, such as the X-ray Radiation Act of Germany.

Radiation emitted by this terminal is less than 0.5 mR/hr at a distance of five (5) centimeters from the surface of the cathode-ray tube. The x-ray radiation primarily depends on the characteristics of the cathode-ray tube and its associated low-voltage and high-voltage circuitry. To ensure safe operation of the terminal, adjust both the low-voltage and high-voltage power supplies as outlined in the service manual for this terminal.

Replace the cathode-ray tube with an identical CRT only. Refer to the service manual for proper HP part number.

Number of German License: BW/133/82Ro

ACHTUNG

Wahrend des Betriebs erzeugt dieses Terminal Rontgenstrahlung. Das Terminal ist so abgeschirmt, da*B* die Dosisleistung weniger als 36 pA/kg (0,5 mR/h) in 5cm Abstand von der Oberflache der Katodenstrahlrohre betragt. Somit sind die Sicherheitsbestimmungen verschiedener Lander, u.A. der deutschen Rontgenverordnung eingehalten.

Die Starke der Rontgenstrahlung hangt im Wesentlichen von der Bauart der Katodenstrahlrohre ab, sowie von den Spannungen, welche an dieser anliegen. Um einen sicheren Betrieb zu gewahrleisten, durfen die Einstellungen der Niederspannungsund des Hochspannungsnetzteils nur nach der Anleitung im Wartungshandbuch dieses Terminals.

Die Katodenstrahlrohre darf nur durch den gleichen Typ ersetzt werden. (Siehe Wartungshandbuch fur HP — Ersatzteile).

Das Gerat ist in Deutschland zugelassen unter.

der Nummber: BW/133/82Ro

HP Computer Museum
www.hpmuseum.net

For research and education purposes only.

NOTICE

The information contained in this document is subject to change without notice.

HEWLETT-PACKARD MAKES NO WARRANTY OF ANY KIND WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Hewlett-Packard shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance or use of this material.

Hewlett-Packard assumes no responsibility for the use or reliability of its software on equipment that is not furnished by Hewlett-Packard.

Preface

The HP 2627A Color Graphics Terminal is a versatile character, line, or block mode CRT terminal. This reference manual contains detailed information for configuring, testing, and using the terminal. For your ease of use, the manual is divided into ten sections and five appendices:

Section I	*General Description*
Section II	*Configuring the Terminal*
Section III	*Keyboard Control*
Section IV	*Alphanumeric Display*
Section V	*Graphics Display*
Section VI	*Data Communications*
Section VII	*External Devices*
Section VIII	*Status*
Section IX	*Error Messages and Self-Tests*
Section X	*Terminal Maintenance Procedures*
Appendix A	*Color Technology*
Appendix B	*Escape Codes*
Appendix C	*Alternate and International Character Sets*
Appendix D	*Programming Examples*
Appendix E	*Handshaking Protocol*

Table of Contents

List of Illustrations

List of Tables

General Description

SECTION I

INTRODUCTION

The HP 2627A Color Graphics Terminal (figure 1-1) is a general purpose CRT terminal offering versatile graphics, alphanumeric, and terminal features:

GRAPHICS:

- High quality raster display of 512 dots by 390 rows in 3 color planes (red, green, blue).
- Eight independent colors: red, green, blue, yellow, cyan, magenta, black, and white.
- User-definable colors allowing you to programmatically mix the 8 basic colors to define new color combinations for area fills.
- Fast vector generation up to 9600 baud with nine predefined line types.
- Polygonal area filling using either predefined or user-defined area patterns.
- Rubberband line graphics giving visual feedback prior to storage in graphics memory.
- Graphics text composition providing variable character orientation and size; block or slanted; and international characters.
- Graphics software support for HP DSG/3000, HP Graphics 1000/II, HPDRAW, and HPEASYCHART.
- Tektronix® 4010 compatibility mode permits operation with TEKTRONIX® PLOT 10 software. Unscaled compatibility mode displays a 512 by 390 subset of the 1024 by 780 addressable points used in Tektronix terminals. Scaled compatibility mode displays a scaled down (512x390) version of the 1024 by 780 addressable points.

ALPHANUMERIC:

- Full color alphanumerics allowing you to use the 8 basic colors to select foregound/background color combinations (color pairs) for each character cell.
- Alphanumeric display enhancements: inverse video, blinking, and underline. Half-bright enhancement mapped to color pair 3 upon display.
- Character, line, page, or block mode operation.
- Full editing capabilities including line operations (insert/delete/clear) and character operation (insert/delete).
- Adjustable margins and tab stops.
- CRT alphanumeric screen size of 164 mm (6.5 inches) by 215 mm (8.5 inches).

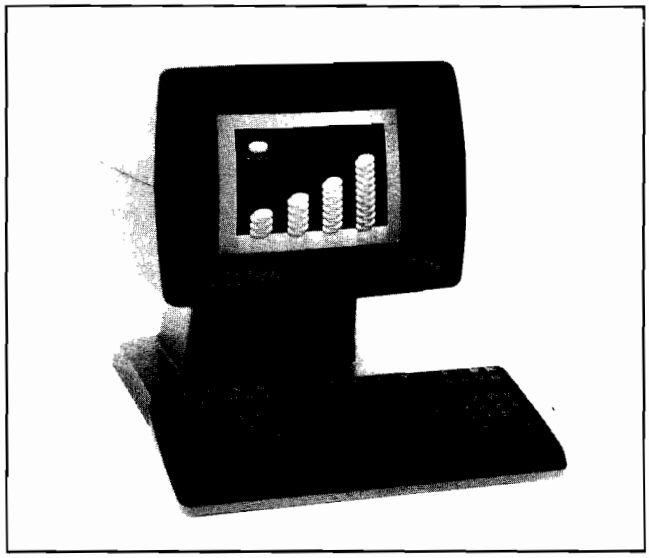


Figure 1-1. HP 2627A Color Graphics Terminal

- Alphanumeric screen capacity of 24 displayed lines of up to 80 characters each (1920 characters total). Two additional lines are provided for system soft key labels.
- Two 24 line pages of alphanumeric memory.

TERMINAL:

- Independent graphics and alphanumeric display memories.
- Screen labeled system function keys (for selecting operating modes and performing other terminal control functions).
- Eight user-definable keys (f1 through f8) are provided. The function of these keys is displayed in the two rows reserved for labels at the bottom of the display screen. You may define each label with a maximum of 16 characters, display enhancement characters included. The character strings returned when the keys are pressed may be defined with up to 80 characters.
- User-definable, with two-characters, RETURN key.
- All terminal configuration operations are performed through keyboard entries into formatted menus displayed on the screen. The configuration data is maintained in non-volatile memory. There are no physical straps.
- USASCII, line drawing, and extended Roman character sets are standard. (The Roman extension supports foreign languages; appropriate keyboards may be ordered as options.)
- Easy-to-use keyboard with separate keypad for graphics control and numeric data entry.

- Programmatic cursor sensing and addressing.
- Extensive self-test capability.
- Flexible data communications permitting a choice between RS232 compatible communications or HP Direct Connect Type 422 communications.

Table 1-1. Specifications

GENERAL	
Screen Size:	164 mm (6.5 inches) by 215 mm (8.5 inches)
Screen Capacity:	24 lines of 80 columns (1920 characters). Two additional lines for system soft key labels.
Character Generation:	7 by 11 character in 9 by 15 dot cell
Character Size:	2.4 mm (0.094 inch) by 3.5 mm (0.138 inch)
Character Set:	ASCII character set, extended Roman set, and line drawing set standard
Color:	Eight programmable foreground/background color pairs; eight fixed color graphics pens.
Alphanumeric Cursor:	Blinking-underline
Graphics Cursor:	Crosshair
Display Enhancements:	Eight color pairs, inverse, underline, and blinking. Half-bright mapped to color pair 3.
Refresh Rate:	Selectable 50 or 60Hz.
Memory:	48 K-byte ROM; Full 2 pages (48 rows) display memory; 256 bytes of RAM for data communication buffer; 128 bytes of non-volatile RAM (battery backup provided) 512 dots by 390 rows by 3 color planes (red, green, blue).
Keyboard:	Detached, with 1.2 m (4 feet) cable. Full ASCII keyboard; 8 screen labeled keys; auto-repeat; N-key rollover. Combination Numeric Pad and Graphics Control Pad.
Operating Modes:	Remote; Local; Character, Line, Page, Block; Forms, Non-Forms.
Transmission Modes:	Full duplex, asynchronous point-to-point. Optional external data communications via modems such as the HP 13265A.
Electrical Interface:	Electrical Industry Association (EIA) Standard RS-232-C. HP Direct Connect Type 422.
Data Rates:	110, 134.5, 150, 300, 600, 1200, 1800, 2400, 4800, 9600 baud.
Parity:	Selectable: Even, odd, zero, one, none
PHYSICAL CHARACTERISTICS	
Weight:	Display Monitor: 20.3 kg (44.6 pounds) standard Keyboard: 2.0 kg (4.4 pounds)

Table 1-1. Specifications (Continued)

Dimensions:	Display Monitor:	
	380 mm wide by 475 mm deep by 440 mm high (15.0 inches by 18.7 inches by 17.3 inches)	
	Keyboard:	
	430 mm wide by 190 mm deep by 75 mm high (17.0 inches by 7.5 inches by 3.0 inches)	
POWER REQUIREMENTS		
Input Voltage:	2627A	120V (+5%, −10%) at 60 Hz (±5%)
	−013	240V (+5%, −10%) at 50 Hz (±5%)
	−014	100V (+5%, −10%) at 60 Hz (±5%)
	−015	220V (+5%, −10%) at 50 Hz (±5%)
	−016	100V (+5%, −10%) at 50 Hz (±5%)
Power Consumption:	200 W	
OPTIONS		
-001	Swedish/Finnish Keyboard	
-002	Danish/Norwegian Keyboard	
-003	French Keyboard	
-004	German Keyboard	
-005	United Kingdom Keyboard	
-006	Spanish Keyboard	
-013	50 Hz, 240 V Power	
-014	60 Hz, 100 V Power	
-015	50 Hz, 220 V Power	
-016	50 Hz, 100 V Power	
-087	Video Interface (providing non-interlaced RGB video output)	
CABLE OPTIONS		
-301	U.S. HARDWIRED/MODEM CABLE (same as 13222N) Male (50 pin)/male (25 pin); 5 meters (16 feet). For use on Port 1.	
-302	EUROPEAN MODEM CABLE (same as 13222M) Male (50 pin)/male (25 pin); 5 meters (16 feet). For use on Port 1.	
-303	RS232C DATACOMM CABLE (same as 13222C) Male (50 pin)/female (25 pin); 5 meters (16 feet). For use on Port 1.	
-304	HP DIRECT CONNECT TYPE 232 CABLE (same as 13222X) Male (50 pin)/male (3 pin); 5 meters (16 feet). For use on Port 1.	
-305	EMP PROTECT CABLE (same as 13222Y) Male (50 pin)/male (25 pin); 5 meters (16 feet). For use on Port 1.	
-306	HP DIRECT CONNECT TYPE 422 CABLE (same as 13222P) Male (50 pin)/male (5 pin); 5 meters (16 feet). For use on Port 1.	

KEYBOARD

The terminal keyboard (see figure 1-2) is divided into five major groups of keys.

Alphanumeric Group—This group of keys is similar to a standard typewriter keyboard and consists of the alphabetic, numeric, and symbol keys. Included are lower and upper case alphabetic characters, ASCII control codes, punctuation characters, and some commercial symbols.

Graphics Control/Numeric Pad Group These keys are located to the right of the alphanumeric keys. The graphics keys control the graphics display, graphics cursor, and graphics copy to an external peripheral. The layout of the numeric key pad is similar to that of a standard office calculator. These keys may be used for high-speed entry of large quantities of numeric data. SHIFT NUM toggles the numeric and graphics functions. After a hard reset (CTRL SHIFT RESET) or at power-on, the graphics functions are in effect; the numeric functions are off.

Cursor Control Group—This group of keys is used for moving the cursor around on the screen (up, down, left, or right) and for controlling what portion of the display appears on the screen (home up, home down, roll up, roll down, next page, and previous page).

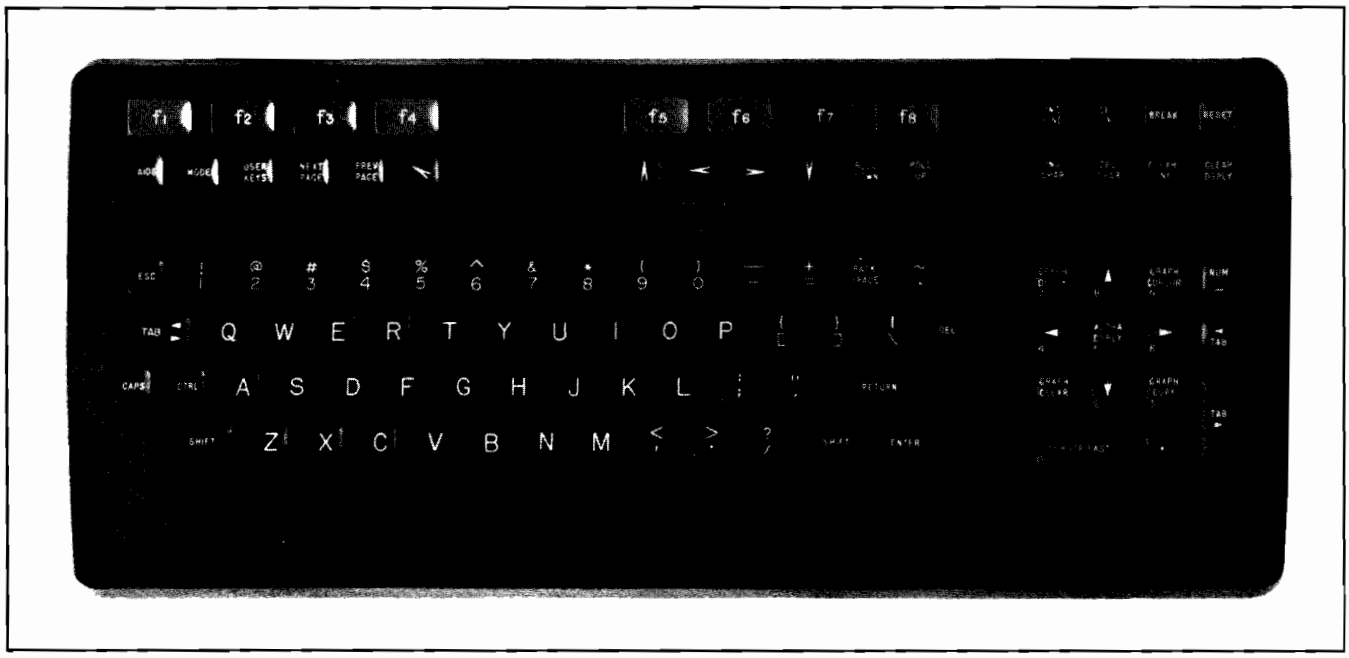


Figure 1-2. HP 2627A Keyboard

Edit Control Group—These keys are used for inserting and deleting characters and lines in relation to the current cursor position.

Function Group—This group of keys (f1 through f8) perform different functions depending upon which keystrokes have been performed. At any given time the applicable labels for these keys appear across the bottom of the display screen.

The United States (USASCII) keyboard is the standard keyboard. As an option you can order any of the following international keyboards instead:

Swedish/Finnish (Option 001)
Danish/Norwegian (Option 002)
French (Option 003)
German (Option 004)
United Kingdom (Option 005)
Spanish (Option 006)

The extended character set is standard on the HP 2627A terminal. You can select any of the following languages using the configuration process:

USASCII (United States)
Swedish/Finnish
Danish/Norwegian
French AZERTY layout with mute keys
French QWERTY layout with mute keys
French AZERTY layout without mute keys
French QWERTY layout without mute keys
German
United Kingdom
Spanish with mute keys
Spanish without mute keys

FUNCTION KEYS

Across the top of the keyboard are eight keys labeled f1 through f8 (figure 1-3). The functions performed by these keys change dynamically as you use the terminal. At any given time the applicable function labels for these keys appear across the bottom of the display screen.

When you press the MODES key, the eight function keys become mode selection keys (figure 1-4). In this capacity you may use the keys to enable and disable various terminal operating modes (such as remote mode and display functions mode). Each mode selection key alternately enables and

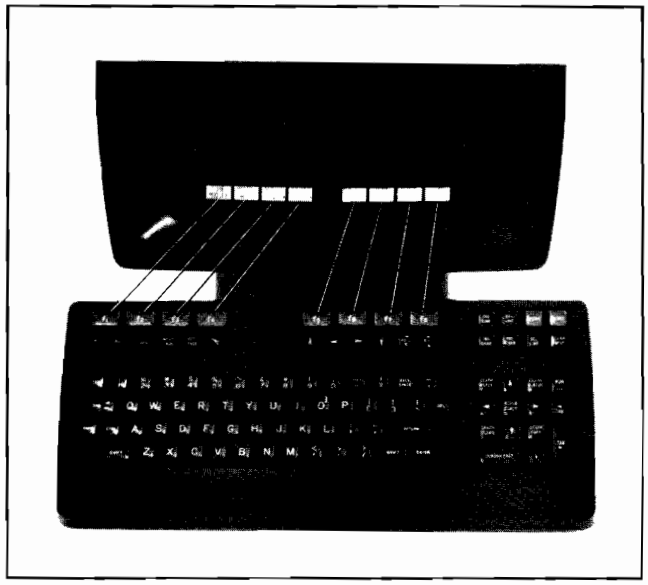


Figure 1-3. Function Keys and Screen Labels

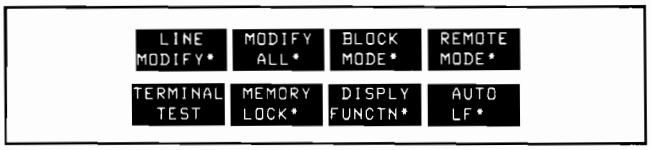


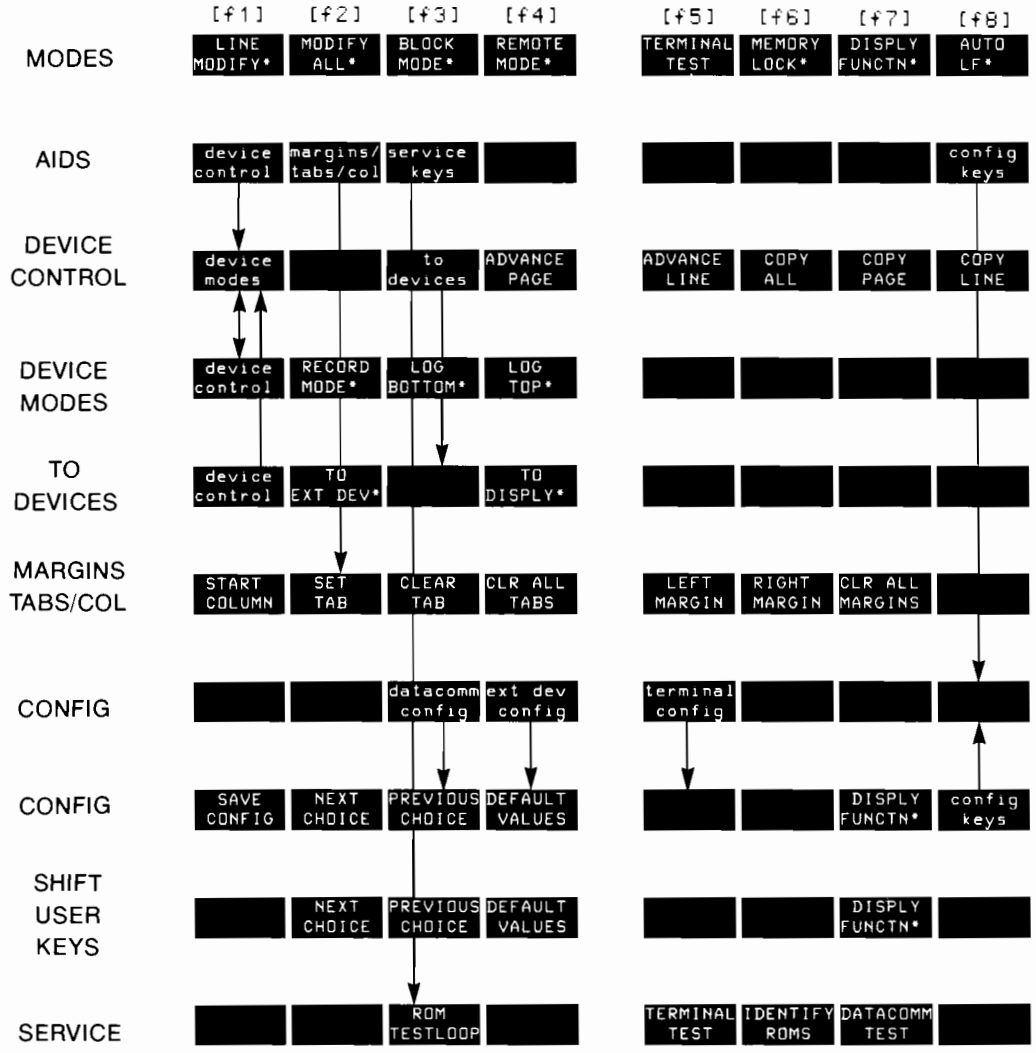
Figure 1-4. Mode Selection Key Labels

disables a particular mode. When the mode is enabled, an asterisk appears in the associated key label on the screen. At power-on, f1 through f8 are automatically initialized as mode selection keys.

When you press the AIDS key, the eight function keys become general control keys that you use for configuring the terminal, setting and clearing margins and tab stops, accessing the Service Keys, and accessing the Device Control group of keys. The entire set of system function key labels for each terminal is illustrated in figure 1-5. Pressing AIDS always reinitializes f1 through f8 to the second row of functions (labels) shown in figure 1-5.

In using the system function keys, keep in mind the following two conventions:

1. If a key label contains any lowercase letters, pressing the key will transfer you to another level of system function keys.
2. If a key label contains only uppercase letters, pressing the key will perform the function defined in the key label.



NOTE

An asterisk (*) in a label indicates an asterisk will be present in the label on the display when the function is active or enabled.

Figure 1-5. HP 2627A Function Key Hierarchy

The key corresponding to AUTO LF for example, sets the automatic line feed function on; whereas the key corresponding to config keys transfers you to the configuration function keys.

When you press the SHIFT and USER KEYS keys simultaneously, the user keys definition menu (figure 1-6) appears on the screen. By filling in this menu you can define the screen label and functional characteristics for eight user keys. To enable the eight user keys, press the USER KEYS keys. Figure 1-7 shows the default user key screen labels and figure 1-8 shows some sample user-defined user key screen labels.

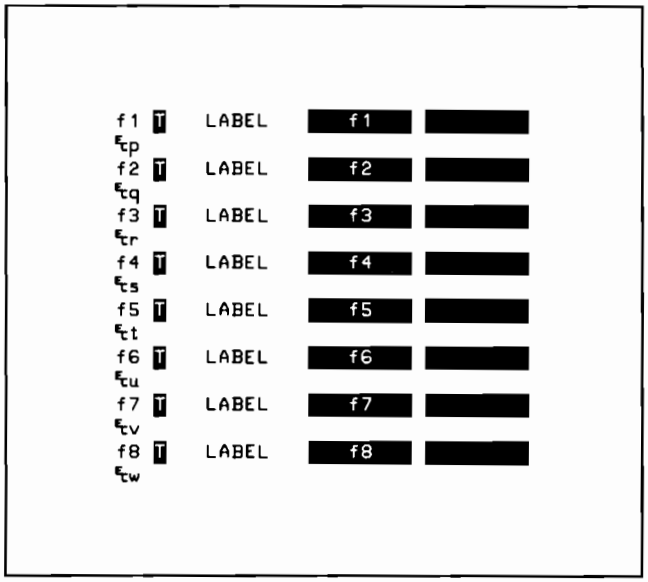


Figure 1-6. User Keys Definition Menu

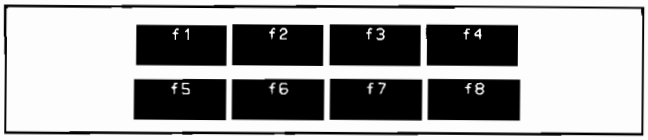


Figure 1-7. Default User Key Labels

Figure 1-8. Sample User-Supplied User Key Labels

You may leave the user-key definition menu by pressing one of three keys: AIDS, MODES, or USER KEYS. Pressing the AIDS or MODES key returns the terminal to the normal screen display. Pressing the USER KEYS grants immediate access to the user key functions.

The key labels are displayed in color pair"7" (default setting is black on yellow). Changing the color of color pair "7" will change the color of the key labels.

Refer to Section III for a complete description of the function keys and information on defining these keys.

CONFIGURING THE TERMINAL

The terminal contains no physical straps or switches (other than the ON/OFF switch on the rear panel). You configure the terminal through the use of a configuration menu displayed on the screen.

Refer to Section II for complete information on configuring the terminal.

EXTERNAL PRINTER PORT

The external printer port at the rear of the HP 2627A terminal provides for interfacing RS-232C printers. The port may be configured to operate from 110 to 9600 baud; choice of 0's, 1's, odd, even, or no parity; 0 to 255 nulls; choice of XON/OFF, SRR, or CS/CB handshaking.

The terminal's electrical interface conforms to EIA RS-232C and CITT V.24 communications interface specifications.

Information on configuring the external printer port is contained in Section VII.

Using either the system function keys or escape sequences, you can select an external printer as the destination device for data transfers. Having done so, you can then perform any of the following types of data transfers using either the system function keys or escape sequences:

- Copy one line from the display.
- Copy all lines visible in the display.
- Copy all data from the current cursor line through the end of display memory.
- Copy all of display memory.
- Copy all of graphics memory.

You can also enable data logging to occur from either the top or bottom of display memory. With top logging, any data that is forced off the top of display memory is directed to the external printer. With bottom logging, any data added to display memory, either from the keyboard or from a datacomm port, is also directed to the external printer.

DATA COMMUNICATIONS

The terminal can operate at speeds ranging from 110 to 9600 baud.

Transmission can be performed in character mode, block line mode, or block page mode; in all cases the data may be either formatted (a data entry form with unprotected and protected fields) or unformatted.

Using the configuration process, you may enable the following forms of parity generation and checking:

None
Odd
Even
Ones (8th bit forced to 1)
Zeros (8th bit forced to 0)

The terminal's electrical interface adheres to EIA RS-232C and CCITT V.24 communications interface specifications.

Refer to Section VI for complete information on data communications.

SELF-TEST

The terminal is engineered for high reliability, ease of testing, and rapid repair.

When the terminal's power is first turned on, a power-on self-test automatically verifies the integrity of all ROM (Read-Only Memory) and RAM (Random-Access Memory) chips within the terminal. The power-on self-test also does a verification of the configuration data stored in the non-volatile memory.

Using the system function keys, you may also initiate any of the following self-tests:

1. Datacomm Test. This is a very flexible forms-selected test that verifies the integrity of either data communications port. Loop-back via a test hood or a modem may be performed.
2. Terminal Test. This self-test does a CRC verification of all ROM chips within the terminal, non-destructively verifies the integrity of all RAM chips within the terminal, performs a graphics system test, and then displays a test pattern on the screen. The test pattern includes all characters (and segments in the case of the line drawing set) as well as all the character enhancements.
3. ROM Test Loop. Checks the positioning of the firmware ROMs and performs a continual CRC verification on the ROMs.
4. Identify ROMs. This self-test generates a listing (on the display screen) of all installed ROMs.

See Section IX for complete information concerning the various self-tests.

Configuring the Terminal

SECTION II

INTRODUCTION

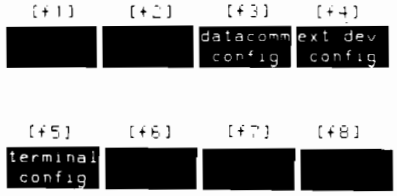
The terminal is designed so that the various terminal characteristics can be configured quickly and easily by displaying configuration "menus" on the screen and then using system function switches to change the content of these menus. Once altered, the terminal's configuration characteristics may be stored in non-volatile memory. (The values stored in non-volatile memory become the active values when you power on the terminal or perform a hard reset.)

The content of these menus may also be altered from a program executing in a host computer through the use of escape sequences. The changes made by the host computer are temporary and will be lost through hard reset or power down.

CONFIGURATION FUNCTION KEYS

To gain access to the configuration menus through the keyboard, press the SHIFT key. This causes the following label display across the bottom of the screen (f1 , f2 , etc., refer to the function key corresponding to the label):

Pressing `config keys` changes the label display to



Pressing `datacomm config` (f3), `ext dev config` (f4), or `terminal config` (f5) causes a configuration "menu" to appear on the screen and redefines the function keys to a set of functions that will assist you in manipulating the various parameters within the menu.

The datacomm menu is described in Section VI, Data Communications. The external device configuration menu is described in Section VII, External Device Control. The terminal configuration menu is described in this section. In addition, information is provided which will enable you to change the terminal configuration menu from the keyboard.

TERMINAL CONFIGURATION MENU

When you press the `terminal config` (f5) function key, the menu and function key display shown in figure 2-1 appears on the screen. Note that the menu as shown in figure 2-1 contains the default settings for all the fields. If you had previously changed the content of any of the fields and then saved the menu in non-volatile memory, the menu would appear on the screen as you last configured it.

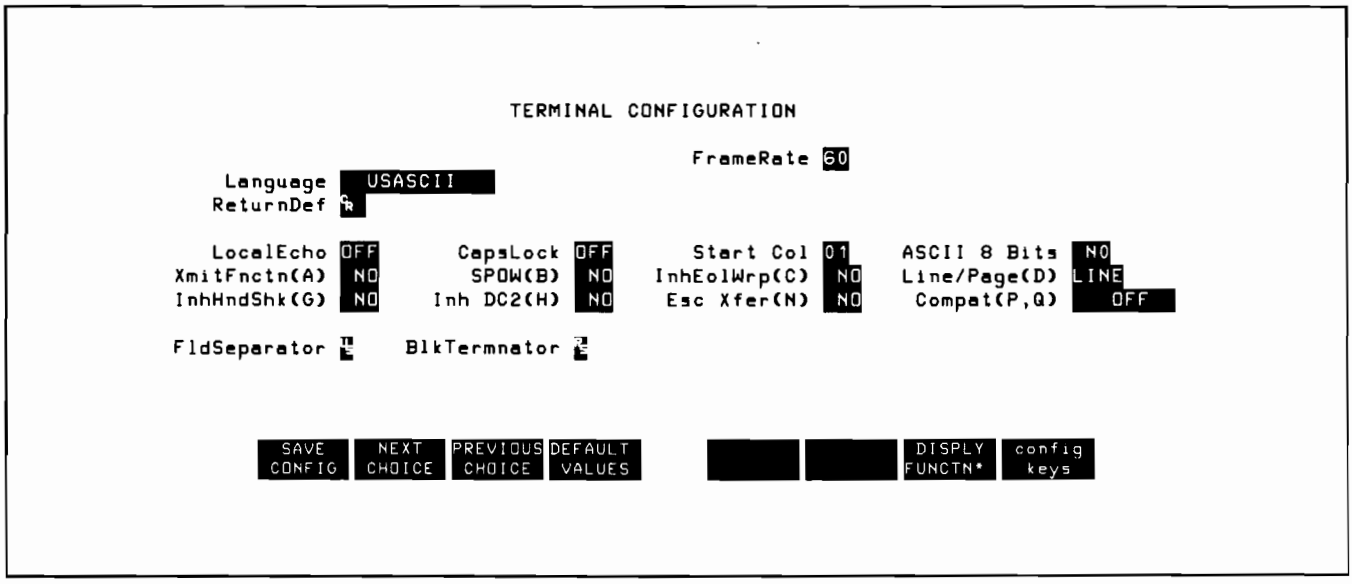


Figure 2-1. HP 2627A Terminal Configuration Menu

NOTE

Whenever a configuration menu is on the screen, no escape sequences are processed, and no incoming or outgoing data is processed, until the menu is exited.

The menu contains a set of unprotected fields. You may access these fields by using the tab keys or the alphanumeric cursor positioning keys. Except when the cursor is positioned in the fields labeled "Return Def", "FldSeparator", or "BlkTermnator", the alphanumeric keys are disabled and you select the desired parameters using the NEXT CHOICE (f2) and PREVIOUS CHOICE (f3) function keys.

The meanings of the various fields are described in table 2-1.

Table 2-1. Terminal Configuration Menu Fields

Language	As will be described in Appendix C, Keyboards and Character Sets, the HP 2627A can be ordered with any of the following keyboards: United States (standard) Swedish/Finnish (option 001) Danish/Norwegian (option 002) French (option 003) German (option 004) United Kingdom (option 005) Spanish (option 006) The terminal automatically includes an extended character set that supports the special characters associated with all of the international languages. With the extended character set you may configure the terminal so that the various keys are interpreted (and displayed) in any desired language regardless of which physical keyboard is being used. For example, with a United States keyboard you could configure the terminal so that it responds to the keys as though they were on a German or French or Danish/Norwegian keyboard. This field specifies which national keyboard format is to be used in interpreting keystrokes. Values: USASCII (United States) SVENSK/SUOMI (Swedish/Finnish) FRANCAIS azM (French AZERTY layout with mutes) FRANCAIS qwM (French QWERTY layout with mutes) FRANCAIS az (French AZERTY layout) FRANCAIS qw (French QWERTY layout) DEUTSCH (German) UK (United Kingdom) ESPANOL M(Spanish with mutes) ESPANOL (Spanish) For the French keyboard layouts, the AZERTY and QWERTY designations refer to the location of the A, Z, Q, and W keys as follows: AZERTY: Row 3 = A Z E R T Y Row 2 = Q S D (etc.) Row 1 = W X C (etc.) QWERTY: Row 3 = Q W E R T Y Row 2 = A S D (etc.) Row 1 = Z X C (etc.)

Table 2-1. Terminal Configuration Menu Fields (Continued)

	For the French and Spanish keyboard layouts, the mutes designation refers to the manner in which certain accent character keystrokes are handled (' and " on the French layout and ' on the Spanish). If the mutes are enabled, those keystrokes will generate the particular accent character but will NOT move the cursor. If you then type an applicable vowel, the vowel will appear in the same character position as the accent and the cursor then moves to the next column. However, if you type any character other than an applicable vowel, (*or the space character*), that character replaces the accent character. Default: USASCII
FrameRate	This field specifies what line frequency (50 or 60 Hz) the terminal is designed to operate at. The screen refresh rate is then synchronized to the specified frequency. If this field is set to the wrong value, the images on the screen will pulsate visibly (this will not happen until the save config key is pressed). Values: 50 (for 50 Hz power source) 60 (for 60 Hz power source) The FrameRate can be temporarily selected with the following escape sequences: $^{E}_{c}$&k 1J = 50 Hz $^{E}_{c}$&k 0J = 60 Hz Default: 60
RETURN def	This field specifies the definition of the RETURN key. The default definition is an ASCII $^{C}_{R}$. The definition may consist of up to two characters. If the second character is a space, it is ignored. Default: $^{C}_{R}$
LocalEcho	This field specifies whether characters entered through the keyboard are both displayed on the screen and transmitted to the host computer. ON ($^{E}_{c}$&k 1L) Characters entered through the keyboard are both displayed on the screen and transmitted to the host computer. OFF ($^{E}_{c}$&k 0L) Characters entered through the keyboard are transmitted to the host computer only. (If they are to appear on the screen, the host computer must "echo" them back to the terminal). Default: OFF
Caps Lock	This field specifies whether the terminal generates the full 128-character ASCII set or only Teletype-compatible codes. ON ($^{E}_{c}$&k 1C) The terminal generates only Teletype-compatible codes: uppercase ASCII (00-5F, hex) and DEL (7F, hex). Unshifted alphabetic keys (a-z) generate the codes for their uppercase equivalents, the {, |, and } keys generate the codes for [, \, and], respectively. The key for generating ~ and ` is disabled. OFF ($^{E}_{c}$&k 0C) The terminal generates the full 128-character ASCII set of codes. Default: OFF
Start Col	If the line in which you are entering data is the bottommost used line in display memory (there are no printing or non-printing characters following the current line in display memory), the terminal automatically generates a logical start-of-text pointer to designate the leftmost character that you enter in the line. This pointer remains with the line in display memory until the line is deleted.

Table 2-1. Terminal Configuration Menu Fields (Continued)

	When you are operating in MODIFY LINE or MODIFY ALL mode and you press ENTER or RETURN, the data transmission from the terminal normally begins at the logical start-of-text pointer in the particular line. If the line has no logical start-of-text pointer, however, the data transmission begins at the designated start column. This column can be defined and saved in non-volatile memory using the `StartCol` field of the terminal configuration menu. The active value of this field can also be temporarily redefined through the "`margins/tabs/col`" function keys. Values: 1 - 80 Default: 1
`ASCII 8 Bits`	When this operating mode is enabled (= YES), the terminal transmits 8-bit ASCII codes in which the eighth (high-order) bit, when set (= 1), indicates that the character is from the Roman extension character set. Values: YES (Esc&k 1I) = 8-bit Roman extension code. NO (Esc&k 0I) = Standard 7-bit ASCII code. NOTE: When using 8-bit mode, parity should be set to NONE; otherwise, the eighth bit will be used as the parity bit. Default: NO
`XmitFnctn(A)`	This field specifies whether escape code functions are both executed at the terminal and transmitted to the host computer. YES (Esc&s 1A) The escape code sequences generated by control keys such as ROLL UP and ROLL DOWN are transmitted to the host computer. If local echo is ON, the function is also performed locally. NO (Esc&s 0A) The escape code sequences for the major function keys are executed locally but NOT transmitted to the host computer. Default: NO
`SPOW(B)`	This field specifies whether or not spaces entered through the keyboard will overwrite existing characters. NO (Esc&s 0B) Spaces entered through the keyboard will overwrite existing characters. YES (Esc&s 1B) Enable SPace OverWrite (SPOW) latch. When the SPOW latch is off, overwriting occurs as normal. When the SPOW latch is on, spaces entered through the keyboard move the cursor forward but do not overwrite existing characters. Once the SPOW latch has been enabled by the above escape sequence, it is turn on by a carriage return and is turned off by a line feed, home up, or tab. It may also be turned on and off programmatically through the use of an Esc&k sequence as follows: ON: Esc&k 1N OFF: Esc&k 0N Default: NO

Table 2-1. Terminal Configuration Menu Fields (Continued)

InhEolWrp(C)	This field specifies whether or not the end-of-line wrap is inhibited. NO (Esc&s 0C) When the cursor reaches the right margin it automatically moves to the left margin in the next lower line (a local carriage return and line feed are generated). YES (Esc&s 1C) When the cursor reaches the right margin it remains in that screen column until an explicit carriage return or other cursor movement function is performed (succeeding characters overwrite the existing character in that screen column). Default: NO
Line/Page(D)	This field specifies whether or not the terminal, when operating in block mode, will transmit data a line at a time or a page at a time. Line (Esc&s 0D) When operating in block mode, the terminal will transmit data a line at a time. Page (Esc&s 1D) When operating in block mode, the terminal will transmit data a page at a time. For a detailed description of the differences between block line and block page mode, refer to "ENTER KEY" in Section III of this manual. Default: LINE
InhHndShk(G) and Inh DC2(H)	Together, these fields determine what type of handshaking is to be used when transferring blocks of data from the terminal to the host computer. (Appendex E summarizes the different handshaking types.) The various types of block transfers that may occur are as follows: • A data transfer initiated by pressing the ENTER key in character, block line, or block page mode. • A data transfer initiated by pressing the ENTER or RETURN key in modify mode. • A data transfer initiated by pressing a transmit only (T) user key (f1 through f8). • The terminal's response to a cursor sense, terminal ID status, primary status, secondary status, color pair status, or device status request issued from the host computer. • The device control completion code (S, F, or U) transmitted by the terminal in conjunction with a device control operation initiated by the host computer. When performing block transfers, there are three possible handshakes: 1. No handshake; terminal merely transmits block of data. 2. Computer sends $^{D}_{1}$; terminal transmits block of data. 3. Computer sends $^{D}_{1}$; terminal responds with $^{D}_{2}$; computer responds with another $^{D}_{1}$; terminal transmits block of data. NOTE When the $^{D}_{1}$ $^{D}_{2}$ $^{D}_{1}$ handshake is enabled and the line/page field of terminal config is selected to be "line", a $^{C}_{R}$ or $^{C}_{R}$ $^{L}_{F}$ is transmitted after the $^{D}_{2}$. If "line" is not selected, no $^{C}_{R}$ or $^{C}_{R}$ $^{L}_{F}$ is transmitted after $^{D}_{2}$.

Table 2-1. Terminal Configuration Menu Fields (Continued)

In general, the InhHndShk(G) and Inh DC2(H) fields have the following effects:

InhHndShk(G) = YES	Eliminates the use of the P_1 handshake (terminal will either use the $P_1/P_2/P_1$ handshake or no handshake at all).
Inh DC2(H) =YES	Eliminates the use of the $P_1/P_2/P_1$ handshake (terminal will either use the P_1 handshake or no handshake at all).
Both = YES	No handshake.

Specifically, however, the type of handshaking used for block transfers is determined by a combination of the following factors:

1. The type of block transfer to be performed.
2. What mode the terminal is currently operating in (character, block line, block page, or modify mode).
3. The setting of the InhHndShk(G) and Inh DC2(H) fields.

If your terminal is connected to a Hewlett-Packard computer system, you will find that the default settings for these fields (both NO) are usually adequate for your purposes. If you are concerned about the specific type of handshake to be used for one or more of the particular types of block transfers, however, you should use the following summary to verify (or alter) the settings of the InhHndShk(G) and Inh DC2(H) fields:

1. ENTER key in block mode; or

   Transmit only (T) user key in block page mode.

   InhHndShk(G) (ignored)

   Inh DC2(H) = NO → $P_1/P_2/P_1$

   Inh DC2(H) = YES → no handshake

2. ENTER key in character mode, or

   ENTER or RETURN key in modify mode.

   InhHndShk(G) = YES

   Inh DC2(H) = NO → $P_1/P_2/P_1$

   Any other combination → no handshake

3. Transmit only (T) user key in block line or character mode; or

   Cursor sense, terminal ID status, primary status, secondary status, display transfer initiated by ESC d, graphics status transfer, color pair status transfer, or device status request; or

   Device control completion code.

   InhHndShk(G) = NO → P_1

   InhHndShk(G) = YES

   Inh DC2(H) = NO → $P_1/P_2/P_1$

   InhHndShk(G) = YES

   Inh DC2(H) = YES → no handshake

Table 2-1. Terminal Configuration Menu Fields (Continued)

	The G and H selections can be temporarily selected by the following escape sequences: Esc&s0G disables G Esc&s0H disables H Esc&s1G enables G Esc&s1H enables H Defaults: InhHndShk(G) = NO Inh DC2(H) = NO
Esc Xfer(N)	YES (Esc&s 1N) = When transferring data from display memory to an external printer, escape sequences relating to the display (such as those specifying display enhancements, format mode fields, and alternate character sets) are sent to the external printer if encountered within the data. NO (Esc&s 0N) = Escape sequences relating to the display are not sent to the external printer. NOTE: The Esc Xfer(N) field only affects data transfers between display memory and an external printer. It does NOT affect <Esc>&pW data transfers that go directly from the host computer to the external printer. Default: OFF
Compat(P,Q)	This field is the functional equivalent of keyboard straps P and Q in the HP 2647A and HP 2648A Graphics Terminals. Either UNSCALED or SCALED entries cause the terminal to enter Compatibility Mode which allows the terminal to respond to graphics control codes for the Tektronix 4010 and 4012 terminals. (Compatibility Mode is explained in detail in Section V.) UNSCALED (Esc&s0p 1Q) = Graphics display shows a 512 by 390 subset of the 1024 by 780 addressable points used in the Tektronix terminals. SCALED (Esc&s1p 0Q) = Graphics display shows a scaled down (512 x 360) version of the 1024 x 780 addressable points used in the Tektronix terminals. The entire picture is displayed at approximately ½ the resolution. OFF (Esc&s0p 0Q or Esc&s1p 1Q) = Turns off Compatibility Mode. Default: OFF
FldSeparator	When you press the ENTER key while the terminal is in block page and format mode and display memory contains a formatted display, the terminal automatically transmits the specified field separator character at the end of each unprotected field (except the final one). Value: Any ASCII character Default: ␣
BlkTermnator	For data transfers between the terminal and a host computer, the terminal (under certain circumstances) transmits the specified block terminator character at the end of the transfer operation. For details, see "The ENTER Key" in Section III. This character, when encountered in display memory, terminates a data transfer (ENTER key transmissions). Value: Any ASCII character Default: ␍

Note that as you alter the fields of a configuration menu on the screen, the selected values do NOT alter the content of non-volatile memory nor do they have any effect on the operation of the terminal.

When you have set all the fields to the desired values, you may then save them in non-volatile memory using the save config (f1) function key.

When you do this, the chosen values take effect immediately.

While the terminal configuration menu is displayed on the screen, the f4, f7, and f8 function keys perform as shown in table 2-2.

Lock/Unlock Configuration Menus

Using an escape sequence, you can "lock" the current configuration menus (terminal config, ext dev config, or datacomm config) so that the menu can not be accessed from the keyboard. Any attempt to access a locked menu from the keyboard will result in a "beep" from the bell. Note that when the configuration menus are locked, the MODIFY ALL (f2), BLOCK MODE (f3), REMOTE MODE (f4), and AUTO LF (f8) mode selection keys are also locked.

To lock the menus, use the following escape sequence:

Ec&q1L

To unlock the menus, use the following escape sequence:

Ec&q0L

SETTING CONFIGURATION PARAMETERS WITH ESCAPE CODES

To set the terminal configuration parameters using escape codes, you must use an Ec&k or Ec&s sequence, depending upon which parameters you wish to set.

Parameter Name As Shown in Menu	Type of Escape Sequence Used
FrameRate	Ec&k
LocalEcho	
Caps Lock	
ASCII 8-Bit	
XmitFnctn(A)	Ec&s
SPOW(B)	
InhEolWrp(C)	
Line/Page(D)	
InhHndShk(G)	
Inh DC2(H)	
Esc Xfer(N)	
Compat (P,Q)	

The Ec&k and Ec&s sequences alter the particular parameter in the menu, and the new setting takes effect immediately, but they do NOT alter the content of non-volatile memory. (See Table 2-1 for parameter description.)

Table 2-2. Configuration Function Keys f4, f7, f8

[f4] DEFAULT VALUES	Pressing this key causes all fields in the menu on the screen to be filled with their default values. (For default values, see figure 2-1.)
[f7] DISPLY FUNCTN	Pressing this key alternately enables and disables display functions mode. When enabled, an asterisk appears in the function key display. You use display functions mode for entering ASCII control characters in the Return Def, FldSeparator, or BlkTermnator fields. Note that this implementation of display functions mode is separate from that which is enabled/disabled via the mode selection keys. Enabling or disabling display functions mode using this function key does NOT alter the effect of the DISPLY FUNCTN mode selection key (and vice versa).
[f8] config keys	Pressing this key removes the menu from the screen (WITHOUT activating it or saving it in non-volatile memory) and returns the function key labels to the following: [f1] [f2] [f3] datacomm config [f4] ext dev config [f5] terminal config [f6] [f7] [f8]

To change the active values of the FrameRate, LocalEcho, Caps Lock, ASCII 8-Bit, or SPOW parameters, use an escape sequence of the following form:

FrameRate = 60:	^&k 0J
FrameRate = 50:	^&k 1J
LocalEcho = OFF:	^&k 0L
LocalEcho = ON:	^&k 1L
Caps Lock = OFF:	^&k 0C
Caps Lock = ON:	^&k 1C
ASCII 8-Bit = YES:	^&k 1I
ASCII 8-Bit = NO:	^&k 0I

You may combine these and other ^&k parameters within one escape sequence. If you do, *the final identifier (such as* C *or* I *or* L*) must be uppercase and all preceding identifiers must be lowercase.* For example, to set LocalEcho = ON and Caps Lock = ON, you could use either of the following escape sequences:

^&k 1l 1C
^&k 1c 1L

To change the active values of any of the following parameters, use an escape sequence of the following form:

XmitFnctn(A) = NO:	^&s 0A
XmitFnctn(A) = YES:	^&s 1A
SPOW(B) = NO:	^&s 0B
SPOW(B) = YES:	^&s 1B
InhEolWrp(C) = NO:	^&s 0C
InhEolWrp(C) = YES:	^&s 1C
Line/Page(D) = LINE:	^&s 0D
Line/Page(D) = PAGE:	^&s 1D
InhHndShk(G) = NO:	^&s 0G
InhHndShk(G) = YES:	^&s 1G
Inh DC2(H) = NO:	^&s 0H
Inh DC2(H) = YES:	^&s 1H
Esc Xfer(N) = NO:	^&s 0N
Esc Xfer(N) = YES:	^&s 1N
Compat (P,Q) = OFF:	^&s0p 0Q or ^&s1p 1Q
Compat (P,Q) = UNSCALED:	^&s0p 1Q
Compat (P,Q) = SCALED:	^&s1p 0Q

You may combine these and other ^&s parameters within one escape sequence. If you do, *the final identifier (such as* A *or* G *or* H*) must be uppercase and all preceding identifiers must be lowercase.* For example, to set Line/Page(D) = PAGE, InhHndShk (G) = NO, and Inh DC2(H) = YES, you could use any of the following excape sequences:

^&s 1d 0g 1H
^&s 0g 1h 1D
^&s 1h 1d 0G

Keyboard Control

SECTION III

INTRODUCTION

The terminal keyboard is a separate unit that is linked to the display portion of the terminal by a flexible cable. Included within the keyboard unit is a speaker that is used for sounding the terminal's bell tone. Except for two keys (RESET and BREAK), the overall keyboard can be logically divided into a character set group, a numeric pad or graphics pad group, a cursor control group, an edit control group, and a function key group. The function key group includes eight keys labeled "f1" through "f8" and the keys labeled "AIDS", "MODES" and "USER KEYS". The f1 through f8 keys are multi-purpose keys in that the functions they perform vary from one situation to another. At any given time the applicable labels for the function keys are displayed across the bottom of the screen (figure 3-1).

SELECTING MODES

Pressing the MODES key enables the mode selection keys and changes the f1 through f8 screen labels to the following:

[f1]	[f2]	[f3]	[f4]
LINE MODIFY*	MODIFY ALL*	BLOCK MODE*	REMOTE MODE*
[f5]	[f6]	[f7]	[f8]
TERMINAL TEST	MEMORY LOCK*	DISPLY FUNCTN*	AUTO LF*

Except for the TERMINAL TEST key (which initiates the terminal self-test), these keys act as toggle switches in that they alternately enable and disable the designated mode. When a particular mode is enabled, an asterisk is displayed in the label.

The graphics functions of the graphics/numeric keypad are in effect when the terminal is powered on or after a hard reset. Also, SHIFT NUM toggles the keypad functions between graphics and numeric functions.

Remote/Local Modes

When a communications link exists between the terminal and a remote host computer, the terminal is in either of the following two modes:

- **Remote Mode.** In this mode, when you press an alphanumeric key the associated ASCII code is transmitted to the host computer.
- **Local Mode.** In this mode, when you press an alphanumeric key the associated character is displayed at the current cursor position on the screen (nothing is transmitted to the host computer).

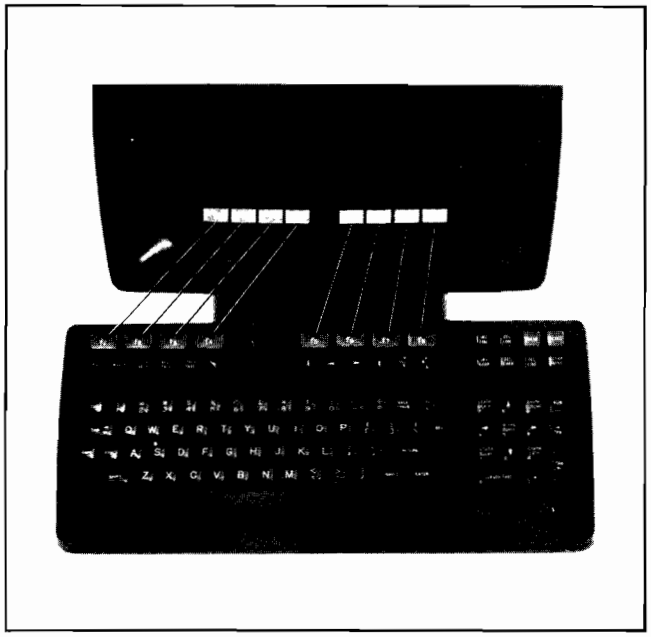


Figure 3-1. Screen-Labeled Function Keys

From the keyboard, you switch the terminal back and forth between local and remote modes using the REMOTE MODE (f4) key.

From the keyboard or a user-definable key, you can switch the terminal from local to remote (and vice versa) using the following escape sequences:

Local: Esc&k 0R
Remote: Esc&k 1R

A remote/local mode designator is maintained in non-volatile memory. When you change modes using the REMOTE MODE key, you also alter that mode designator in non-volatile memory. When you change modes using the escape sequences, however, the designator in non-volatile memory is NOT altered.

After a hard reset or turning off the power, the terminal reverts to the mode specified by the remote/local designator in non-volatile memory.

Character/Block Modes

When the terminal is connected on-line to a remote host computer, it operates in either of the following data transmission modes:

- **Character Mode.** In this mode, data is transmitted a character at a time as it is entered through the keyboard. ASCII control codes (such as CR and LF) are transmitted.

- **Block Mode.** In this mode, data is NOT transmitted at the time it is entered through the keyboard. Instead, you transmit an entire block of data by first typing the data (after initially typing the data you can move the cursor around and edit the data as desired) and then pressing the ENTER key.

When the terminal is in block mode, ASCII control codes (such as ^R and ^F) are acted upon locally but NOT transmitted with the data block.

From the keyboard, you enable and disable block mode using the BLOCK MODE (f3) key.

From a program executing in a host computer, you enable and disable block mode using the following escape sequences:

ENABLE: Esc&k1B
DISABLE: Esc&k0B

A character/block mode designator is maintained in non-volatile memory. When you change modes using the BLOCK MODE key, you also alter that mode designator in non-volatile memory. When you change modes using the escape sequences, however, the designator is NOT altered.

After a hard reset or turning off the power, the terminal reverts to the mode specified by the character/block designator in non-volatile memory.

The relationship between block, line, page, and format modes is described under ENTER key later in this section.

Format Mode

The terminal includes a format mode in which elaborate, custom-designed forms containing protected and unprotected fields can be displayed on the screen and used for data entry.

When format mode is enabled, the terminal operator may only enter data into unprotected fields. If the operator positions the cursor in a protected area and then attempts to type data, the cursor automatically moves to the start of the next subsequent unprotected field before the terminal accepts the data.

The designing of forms and the use of format mode are described in Section IV.

From a program executing in a host computer or from the keyboard, you enable and disable format mode using the following escape sequences:

ENABLE: EscW
DISABLE: EscX

Once format mode is enabled, it remains enabled until explicitly disabled, until a hard reset is performed, or until the power is turned off.

Line Modify Mode

When the terminal is in remote mode and character mode, and you are communicating interactively with a host computer, you may sometimes enter an erroneous command string to which the computer responds with an error message. If the command string is a lengthy one and the error consists of only a few characters, it is a nuisance to have to retype the entire string. In such a case, you may instead enable line modify mode (which temporarily switches the terminal to a special form of block mode). You may then move the cursor to the erroneous line on the display and correct the command string. When the string is edited to your satisfaction, you retransmit the line to the host computer by pressing either the RETURN key or the ENTER key.

Note that while line modify mode results in a block transmission, it is completely independent of the block mode function described earlier in this section (you do NOT have to first enable block mode). In fact, line modify mode is a feature that was specifically designed for use when the terminal is operating in character mode.

From the keyboard, you enable line modify mode using the LINE MODIFY (f1) key. Line modify mode is automatically disabled during a hard reset or when you press either RETURN or ENTER. If you change your mind and wish to disable line modify mode before retransmitting the command string, press the LINE MODIFY key again and the terminal will return to normal character mode.

The terminal remembers which character was the first (leftmost) one that you entered through the keyboard. This means that when you retransmit a line in modify mode, only the keyboard entry portion of the line (the entire edited command string) is retransmitted; any prompt characters preceding the command string are ignored by the terminal. For more detailed information about this feature refer to the discussion of the Start Col field of the terminal configuration menu in Section II.

Modify All Mode

When the terminal is in character mode, you can enable modify all mode, which switches the terminal to a special form of block mode. Modify all mode is the same as line modify mode except that it is NOT disabled when you press RETURN or ENTER or when a hard reset is performed.

From the keyboard, you enable and disable modify all mode using the MODIFY ALL (f2) key.

From a program executing in a host computer, you enable and disable modify all mode using the following escape sequences:

ENABLE: Esc&k1M
DISABLE: Esc&k0M

A modify all mode designator is maintained in non-volatile memory. When you change modes using the MODIFY ALL

key, you also alter that mode designator in non-volatile memory. When you change modes using the escape sequences, however, the designator is NOT altered.

After a hard reset or turning off the power, the terminal reverts to the mode specified by the modify all designator in non-volatile memory.

NOTE

Modify mode uses the same type of handshaking as that used when the ENTER key is pressed while the terminal is in Character mode (that is, while Block mode is disabled). See Table 3-1. Also, Appendix E summarizes which type of handshaking is used for different terminal configurations.

Auto Line Feed Mode

When auto line feed mode is enabled, an ASCII line feed control code is automatically appended to each ASCII carriage return control code generated through the keyboard. That is, every CR code generated through the keyboard becomes a CR LF.

ASCII carriage return control codes can be generated through the keyboard in any of the following ways:

- By pressing the RETURN key, provided that a CR code is included in the key definition.
- By simultaneously pressing the CTRL and M keys.
- By pressing any of the user keys (f1 through f8), provided that a CR code is included in the particular key definition.
- By pressing the ENTER key when the terminal is in block mode, line modify mode, or modify all mode (in these cases a CR code is transmitted as the line terminator).

From the keyboard, you enable and disable auto line feed mode using the AUTO LF (f8) key.

From a program executing in a host computer, you enable and disable auto line feed mode using the following escape sequences:

ENABLE: Esc&k 1A
DISABLE: Esc&k 0A

When you enable or disable auto line feed mode using the "AUTO LF" key, you also alter the content of the "AUTO LF" field in non-volatile memory. When you enable or disable auto line feed mode using the escape sequence, however, you do not change the content of the non-volatile memory.

After a hard reset or turning off the power, the terminal reverts to the mode specified by the "AUTO LF" field in non-volatile memory.

Memory Lock Mode

Memory lock mode provides two separate functions: overflow protect and display lock.

OVERFLOW PROTECT. This feature prevents you from losing data when display memory is full. If you home the cursor and then enable memory lock mode, display memory becomes "protected" so that no data can be lost off the top. In such a case, when you have used all available lines in display memory, any attempt to use more memory is rejected with an audible "beep". You may, however, use the cursor control keys to go back and alter any of the existing data. To continue entering new data, merely disable memory lock mode and reposition the cursor immediately below the last line. Before doing so you may wish to enable data logging (described in Section VII) so that data that is then forced off the top of display memory will be retained in printed form.

DISPLAY LOCK. If you position the cursor below the top line of the screen and then enable memory lock mode, the lines above the cursor's current line become "locked" on the screen. As the screen becomes full, the locked lines remain on the screen while subsequent lines roll past the locked rows. This allows you to retain column headings or instructions on the screen as you continue to enter new data. It also provides a useful means of changing the sequence of text blocks as follows:

a. Press ↖, CLEAR DISPLY, and then type the following data:

```
3. This is paragraph 3. It
   should be the third one.
1. This is paragraph 1. It
   should be the first one.
2. This is paragraph 2. It
   should be the second one.
4. This is paragraph 4. It
   should be the last one.
```

b. Using the alphanumeric cursor movement keys, position the cursor in the first line of paragraph 1.

c. Enable memory lock mode by pressing function key f6 if an asterisk is not present in the corresponding screen label.

d. Use the ROLL UP key until the first line of paragraph 4 is in the same line as the cursor.

e. Disable memory lock mode and home the cursor. The display should appear as follows:

```
1. This is paragraph 1. It
   should be the first one.
2. This is paragraph 2. It
   should be the second one.
3. This is paragraph 3. It
   should be the third one.
4. This is paragraph 4. It
   should be the last one.
```

From the keyboard, you enable and disable memory lock mode using the MEMORY LOCK (f6) key. The rows above the line containing the cursor are locked.

Normal editing can be performed within the locked rows. That is, the rows are locked by row number only, so if lines are inserted among the locked rows, they become locked while all "excess" rows are forced out of the locked area.

From a program executing in a host computer, you enable and disable memory lock mode using the following escape sequences:

ENABLE: ESC l
DISABLE: ESC m

Once enabled, memory lock mode remains enabled until explicitly disabled, until a hard reset is performed, or until the power is turned off.

Display Functions Mode

When display functions mode is enabled the terminal operates as follows:

- In local mode, it displays ASCII control codes and escape sequences but does not execute them. For example, if you press the ← key the terminal displays ESC D on the screen but does not perform the "cursor left" function.
- In remote mode, it transmits ASCII control codes and escape sequences but does not execute them locally. For example, if you press the ROLL UP key the terminal transmits an ESC S but does not perform the "roll up" function. If local echo is enabled (ON) then the ESC S is also displayed on the screen.

There are two exceptions to the foregoing descriptions:

1. An ESC Z, which disables display functions mode, or ESC Y, which enables display functions mode, is executed locally and either transmitted (in remote mode) or displayed (in local mode).
2. A CR (or CR LF if auto line feed mode is enabled) is executed in addition to being transmitted and displayed.

From the keyboard, you enable and disable display functions mode using the DISPLY FUNCTN (f7) key.

From a program executing in a host computer, you enable and disable display functions mode using the following escape sequences:

ENABLE: ESC Y
DISABLE: ESC Z

NOTE

There is interaction between display functions and the XmitFnctn(A) field of the terminal configuration menu. If XmitFnctn(A) is on, the DISPLY FUNCTN key transmits ESC Y, ESC Z. As a result of this action, the receiver of the transmission can never exit display functions unless the ESC Z sequence is explicitly sent.

Once enabled, display functions mode remains enabled until explicitly disabled, until a soft or hard reset is performed, or until the power is turned off.

Caps Mode

When caps mode is enabled, all unshifted alphabetic keys generate uppercase letters and all shifted alphabetic keys generate lowercase letters. This mode is used primarily as a typing convenience and only affects the 26 alphabetic keys.

From the keyboard, you enable and disable caps mode using the CAPS key. This key alternately enables and disables caps mode. The CAPS key has no effect if "Caps Lock" in terminal configuration is enabled (ON).

From a program executing in a host computer, you enable and disable caps mode using the following escape sequences:

ENABLE: ESC &k 1P
DISABLE: ESC &k 0P

Once enabled, caps mode remains enabled until explicitly disabled, until a hard reset is performed, or until the power is turned off.

Caps Lock Mode

When caps lock mode is enabled, the terminal generates only Teletype-compatible codes: uppercase ASCII (00-5F, hex) and DEL (7F, hex). Unshifted alphabetic keys (a-z) generate the codes for their uppercase equivalents, the { , | , and } keys generate the codes for [, \ , and] (respectively), and the ` and ~ keys are ignored.

From the keyboard, you enable and disable caps lock mode using the "Caps Lock" field of the terminal configuration menu as described in Section II.

From a program executing in a host computer, you enable and disable caps lock mode using the following escape sequences:

ENABLE: ESC &k 1C
DISABLE: ESC &k 0C

At any given time the current state (enabled/disabled) of caps lock mode is reflected in the "Caps Lock" field of the terminal configuration menu. When you enable or disable the mode by altering the menu field from the keyboard and then pressing the SAVE CONFIG key, you alter both the active and non-volatile memory versions of that field. When you enable or disable the mode using the escape sequence, however, you only change the active value of the "Caps Lock" field in the terminal configuration menu.

After a hard reset or turning off the power, the terminal reverts to the mode specified by the "Caps Lock" field in the terminal configuration menu in non-volatile memory.

GRAPHICS/NUMERIC KEYPAD

The graphics functions on the graphics/numeric keypad are enabled when the terminal is powered on, after a hard reset is performed, or when toggled by the SHIFT NUM keys.

- The [GRAPH DSPLY] key toggles the graphics memory display on and off. This allows you to view the alphanumeric memory without the graphics memory overlaying it.
- The [ALPHA DSPLY] key toggles the alphanumeric memory on and off. This allows you to view the graphics memory without the alphanumeric memory overlaying it.
- The [GRAPH CURSOR] key toggles the graphics cursor on and off. Initially, the cursor display is inhibited. When first toggled on, the cursor is positioned in the lower-left corner of the display.
- The ↑, ←, ↓, → keys control the position of the graphics cursor. Two orthogonal keys may be pressed to cause diagonal movement.
- The [CURSOR FAST] key allows you to position the cursor (using the ↑, ←, ↓, → keys simultaneously) at a more rapid rate.
- The [GRAPH CLEAR] key clears the contents of graphics memory.
- The [GRAPH COPY] key initiates a copy of the contents of graphics memory to the external printer port.

You can toggle the graphics and numeric functions under program control by sending the following escape sequences:

Enable numeric functions: Esc&k0D
Enable graphics functions: Esc&k1D

USER-DEFINABLE KEYS

The eight function keys (f1 through f8), besides performing their usual terminal control functions, can be defined either locally by the terminal operator or remotely by a program executing in a host computer. By "defined" it is meant that:

1. You can assign to each key a string of alphanumeric characters and/or control codes (such as CR or LF).
2. You can specify each key's operational attribute: whether its content is to be executed locally at the terminal, transmitted to a host computer, or both.
3. You can assign to each key an alphanumeric label (up to 16 characters) which, in user keys mode, is displayed across the bottom of the screen.
4. You can select different color pairs for each user-definable label.
5. You can select various video enhancement combinations (blinking, underline, and inverse video) for each user-definable label.

When defining a key from the keyboard, the key content may include explicit escape sequences (entered using display functions mode) that control or modify the terminal's operation.

The definition of each user key may contain up to 80 characters (alphanumeric characters, ASCII control characters, and explicit escape sequence characters).

Defining Keys Locally

To define one or more keys from the keyboard, first press the SHIFT and USER KEYS keys simultaneously, or use Esc j. The user keys menu shown in figure 3-2 then appears on the screen. The screen is displayed as color pair 0 with the ATTRIBUTE field in inverse video; the LABEL fields are displayed in the currently-defined label enhancements. By default, color pair 0 is a white foreground on a black background. However, since colors may be changed programmatically, the actual screen display depends on the currently-defined values for color pair 0. The soft key labels at the bottom of the screen are displayed in color pair 7. You may also programmatically change this color setting. (An example in Section IV shows the effects of changing color pair 7.) Be aware that a poor color selection may obscure information normally displayed on the screen. To prevent this from occurring in the configuration menus, the configuration fields are permanently defined and may not be altered programmatically. Note that the menu in figure 3-2 contains the default values for all of the fields. While the menu is displayed on the screen, you can reset the entire menu to the default values by pressing the DEFAULT VALUES function key (f4).

The menu contains a set of unprotected fields that you access using the tab keys or the alphanumeric cursor-positioning keys.

For each user key the menu contains four unprotected fields:

ATTRIBUTE FIELD. This one-character field always contains an uppercase L, T, or N signifying whether the content of the particular user key is to be:

a. Executed locally only (L).
b. Transmitted to the host computer only (T).
c. Treated in the same manner as the alphanumeric keys (N). If the terminal is in local mode, the content of the key is executed locally. If the terminal is in remote mode and local echo is disabled (OFF), the content of the key is transmitted to the host computer. If the terminal is in remote mode and local echo is enabled (ON), the content of the key is both transmitted to the host computer and executed locally.

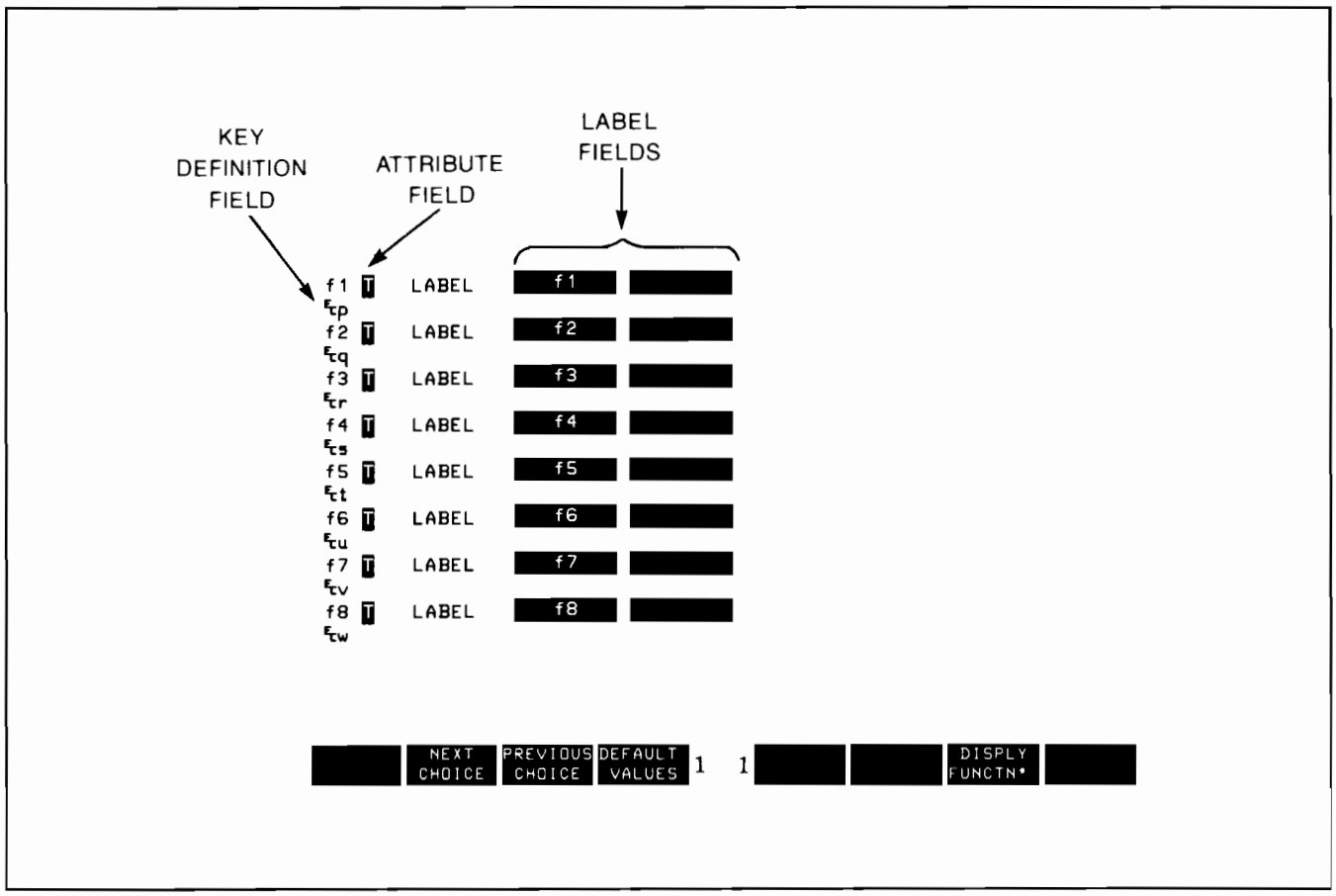


Figure 3-2. User Keys Definition Menu

The alphanumeric keys are disabled when the cursor is positioned in this field. You change the content of this field by pressing the NEXT CHOICE and PREVIOUS CHOICE keys (f2 and f3, respectively).

TWO LABEL FIELDS. The two eight-character fields to the right of the word "LABEL" allow you to supply the user key's label. When the terminal is in user keys mode, the key labels are displayed from left to right in ascending order across the bottom of the screen (each displayed key label occupies two lines). The first LABEL field in the user keys menu supplies the upper portion of the particular key label while the second supplies the lower portion.

KEY DEFINITION FIELD. The entire line (80 characters) immediately below the attribute and label fields is available for specifying the character string that is to be displayed, executed, and/or transmitted whenever the particular key is pressed. When entering characters into this field you may use display functions mode.

When entering the label and key definition you may access display functions mode by way of the DISPLY FUNCTN function key (f7). Note that this implementation of display functions mode is separate from that which is enabled/disabled via the mode selection keys.

With display functions mode enabled, the RETURN key can be used for including ^k codes in key definitions. If not enabled, RETURN repositions the cursor to the first column on the line. If auto line feed mode is also enabled, the RETURN key will generate a ^k^f.

When the user keys menu is displayed on the screen you may use the INS CHAR, DEL CHAR, and CLEAR LINE keys for editing the content of the label and key definition fields.

When you are finished defining all the desired keys, press the AIDS, MODES, or USER KEYS key (in all three cases the user keys menu disappears from the screen). When you press USER KEYS, or enter Ek, the defined user key labels are displayed across the bottom of the screen and the f1 through f8 user keys, as defined by you, are enabled.

Defining Keys Programmatically

From a program executing in a host computer, you can define one or more keys using the following escape sequence format:

Ek&f <attribute><key><color pair><label length><string length><video enh><label><string>

where:

<attribute>	=	0a: normal 1a: local only 2a: transmit only	(0 is the default)
<key>	=	1-8k: f1-f8, respectively	(1 is the default)
<label length>	=	0-16d	(0 is the default)
<string length>	=	0-80L	(1 is the default) (−1 causes field to be erased)
<color pair>	=	0-7C	(7 is the default)
<video enh>	=	0-15v	(0 is the default)
<label>	=	the character sequence for the label field	
<string>	=	the character sequence for the key definition field	

The <attribute>, <key>, <label length>, and <string length> parameters may appear in any sequence but must precede the label and key definition strings. Furthermore, since the <key> parameter automatically resets color pair and video enhancements to their default values, you must place <color pair> and <video enh> after <key> if you want these values to take effect. Besides this constraint and the preceding constraint that the strings defining the label and key must come last, <color pair> and <video enh> may likewise appear in any sequence. You must use an uppercase identifier (A, K, C, D, L or V) for the final parameter and a lowercase identifier (a, k, c, d, l or v) for all preceding parameters. Following the parameters, the first 0 through 16 characters, as designated by <label length>, constitute the key's label and the next 0 through 80 characters, as designated by <string length>, constitute the key's definition string. The total number of displayable characters (alphanumeric data, ASCII control codes such as $^{G}_{R}$ and $^{L}_{F}$, and explicit escape sequence characters) in the label string must not exceed 16, and in the definition string must not exceed 80.

COLOR PAIR. The color pair parameter specifies which color pair to use for the user-definable labels.

NOTE

In the following escape sequences, you must explicitly state a "zero" length parameter as the length parameter defaults to "1." Also, by omitting the key parameter, the new enhancements affect softkey #1.

To specify the color pair programmatically, use the following escape sequence:

^{E}&f<color pair>c 0L

where <color pair> may be in the range −32768 to 32767. The terminal uses the lowest three binary bits of the <color pair> to yield the color pair index number. This value always falls within the octal range of zero through seven. When you omit the optional <color pair>, the index defaults to color pair "7."

VIDEO ENHANCMENT. The user-definable labels can be displayed in one or more video enhancement combinations (blinking, inverse video, and underline). "Half-bright", if selected, is displayed as the currently-defined color pair "3", unless you also specifically select a color enhancement. In this latter case, the half-bright enhancement is ignored. Table 3-1 gives the values for "v" in the video enhancement escape sequence:

^{E}&f<video enh>v 0L (default is 0)

Table 3-1. Video Enhancement Values

"v" Value	Blinking	Inverse Video	Underline	"Half-Bright"
0				
1	X			
2		X		
3	X	X		
4			X	
5	X		X	
6		X	X	
7	X	X	X	
8				X
9	X			X
10		X		X
11	X	X		X
12			X	X
13	X		X	X
14		X	X	X
15	X	X	X	X

Example: Assign LOG-ON as the label and HELLO USER.ACCOUNT as the definition for the f5 user key. The key is to have the attribute "N". The label is to be displayed as color pair 6 and blink.

```
Ec&f5k6c1v6d19LLOG-ONHELLO USER.ACCOUNT^REc&jB
```

After issuing the foregoing escape sequence from your program to the terminal, the f5 portion of the user keys menu is as follows:

```
f5 N LABEL LOG-ON
HELLO USER.ACCOUNT^R
```

The Ec&jB sequence turns on the user labels. If this escape sequence is not sent, the labels remain unchanged until they are redisplayed upon the screen. (This may involve pressing USER KEYS to remove the labels, then pressing USER KEYS again to redisplay the "updated" labels.)

If the transmit only attribute (2) is designated, the particular user key will have no effect unless the terminal is in remote mode. A transmit only user key may (when subsequently pressed) invoke a block transfer handshake and append the appropriate terminator to the string.

Controlling The User Keys Menu Programmatically

From a program executing in a host computer, you can display the user keys menu on the screen and remove it from the screen using the following escape sequences:

DISPLAY MENU: Ecj
REMOVE MENU: Eck

Controlling The Function Key Labels Programmatically

From a program executing in a host computer, you can control the display of the function key labels as follows:

- You can remove the key labels from the screen entirely (this is the equivalent of simultaneously pressing the SHIFT and AIDS keys).
- You can enable the mode selection keys (this is the equivalent of pressing the MODES key).
- You can enable the user keys (this is the equivalent of pressing the USER KEYS key).
- You can "lock" the current set of labels on the screen (i.e., disable the AIDS, MODES, and USER KEYS keys).
- You can reenable the AIDS, MODES, and USER KEYS keys.

Appropriate escape sequences are:

Ec&j@	Enable the user keys and remove all key labels from the screen.
Ec&jA	Enable and display the mode selection keys.
Ec&jB	Enable and display the user keys.
Ec&jR	Unlock screen labels.
Ec&jS	Lock screen labels.
Ec&j<xx>L<message>	Remove the key labels from the screen and display the character string <message> (which consists of <xx> characters; where <xx> may be up to 160 characters).
Ec&jC	Remove your <message> from the screen and restore the current key labels.

ENTER KEY

When the terminal is in remote mode, pressing the ENTER key sets pending a block transfer of data from display memory to the host computer (in such a case the ENTER key also locks the keyboard until the resultant data transfer is complete).

The type of handshaking used and precisely what data gets transmitted depends on the following factors:

1. Whether the terminal is in character mode, block line mode, or block page mode.
2. Whether or not the terminal is in format mode.
3. The settings of the InhHndShk(G), Inh DC2(H), and Line/Page(D) fields in the terminal configuration menu.

Table 3-2 summarizes the effect of the ENTER key in each of the possible mode/strap combinations. Appendix E provides a complete summary of handshaking protocol.

Table 3-2. ENTER Key Operation

CHARACTER MODE

The cursor is repositioned to column 1 of the current line.

All characters through the first subsequent block terminator or through the end of the line (whichever is encountered first) are transmitted to the host computer as a block.

ASCII control codes, color enhancement escape sequences, video enhancement escape sequences, alternate character set escape sequences, and field definition escape sequences are transmitted if encountered.

If the operation is terminated by encountering the end of the line, the terminal sends a CR (or a CR LF if auto line feed mode is enabled). The cursor is repositioned to column 1 and a line feed is performed if auto line feed mode is enabled.

If the operation is terminated by encountering a block terminator, the terminal sends a block terminator followed by a CR (or a CR LF if auto line feed mode is enabled). The cursor remains positioned immediately following the terminator.

If there is no data to be transmitted, the terminal sends a block terminator followed by a CR (or a CR LF if auto line feed mode is enabled).

The type of handshaking used is determined as follows:

InhHndShk(G) = YES
Inh DC2 (H) = NO → $P_1/P_2/P_1$
Any other combination → no handshake

CHARACTER MODE, FORMAT MODE

"Character mode, format mode" means format mode is enabled but block mode is disabled. In this combined mode, when the cursor is within an unprotected field, all characters from the current cursor position through the end of the field are transmitted to the host computer as a block. Otherwise, the terminal searches for the next subsequent unprotected field and transmits the content of that field.

ASCII control codes within the field are transmitted.

Video enhancement escape sequences, color enhancement escape sequences, alternate character set escape sequences, and field definition escape sequences within the field are NOT transmitted.

If the operation is terminated by encountering the end of the unprotected field, the terminal sends a CR (or a CR LF if auto line feed mode is enabled). The cursor remains at the first character position after the end of the field.

If the operation is terminated by encountering a block terminator, the terminal sends a block terminator followed by a CR (or a CR LF if auto line feed mode is enabled). The cursor remains positioned immediately following the terminator.

If there is no data to be transmitted, the terminal sends a block terminator followed by a CR (or a CR LF if auto line feed mode is enabled). The CR that is transmitted has no effect on the terminal locally, and the cursor remains unmoved.

The type of handshaking used is determined as follows:

InhHndShk(G) = YES
Inh DC2 (H) = NO → $P_1/P_2/P_1$
Any other combination → no handshake

BLOCK LINE MODE

Block line mode means that Block Mode is on and that the Line/Page (D) field in terminal configuration is set to "line".

Inh DC2(H) = YES

The cursor is repositioned to column one within the current line. All characters through the first subsequent block terminator or through the end of the line (whichever is encountered first) are then transmitted to the host computer as a block.

Inh DC2(H) = NO

The cursor is NOT repositioned. All characters through the first subsequent block terminator or through the end of the line (whichever is encountered first) are transmitted to the host computer as a block.

ASCII control codes, color enhancement escape sequences, video enhancement escape sequences, alternate character set escape sequences, and field definition escape sequences are all transmitted if encountered.

If the operation is terminated by encountering the end of the line, the terminal sends a CR (or a CR LF if auto line feed mode is enabled). The cursor is repositioned to column 0 and a line feed is performed if auto line feed mode is enabled.

Table 3-2. ENTER Key Operation (Continued)

If the operation is terminated by encountering a block terminator, the terminal sends a block terminator followed by a CR (or a CR LF if auto line feed mode is enabled). The cursor remains positioned immediately following the terminator.

If there is no data to transmit, a block terminator followed by a CR or CR LF is transmitted. The cursor is not moved.

The type of handshaking used is determined as follows:

InhHndShk(G) is ignored

Inh DC2(H) = NO → $P_1/P_2/P_1$

Inh DC2(H) = YES → no handshake

BLOCK LINE MODE, FORMAT MODE

"Block line mode, format mode," means that format mode is enabled and block mode is likewise enabled with the Line/Page (D) field in terminal configuration set to "line".

In this configuration, all characters from the current cursor position through the first subsequent block terminator or through the end of the line (whichever comes first) are transmitted as a block to the host computer.

ASCII control codes within the field are transmitted.

Video enhancement escape sequences, color enhancement escape sequences, alternate character set escape sequences, and field definition escape sequences within the field are NOT transmitted.

If the operation is terminated by encountering the end of the unprotected field, the terminal sends a CR (or a CR LF if auto line feed mode is enabled). The cursor remains positioned at the end of the field.

If the operation is terminated by encountering a block terminator, the terminal sends a block terminator followed by a CR (or a CR LF if auto line feed mode is enabled). The cursor remains positioned immediately following the terminator.

If there is no data to be transmitted, the terminal sends a block terminator followed by a CR (or a CR LF if auto line feed mode is enabled). The CR that is transmitted has no effect on the terminal locally, and the cursor remains unmoved.

The type of handshaking used is determined as follows:

InhHndShk(G) (ignored)

Inh DC2(H) = NO → $P_1/P_2/P_1$

Inh DC2(H) = YES → no handshake

BLOCK PAGE MODE

Block page mode means that Block Mode is on and the Line/Page (D) field in terminal configuration is set to "page".

Inh DC2(H) = YES

The cursor is repositioned to the "home up" position. All characters through the first subsequent block terminator or through the end of display memory (whichever is encountered first) are transmitted to the host computer as a series of blocks, each block corresponding to one line in display memory.

Inh DC2(H) = NO

The cursor is NOT repositioned. All characters from the cursor position through the first subsequent block terminator or through the end of display memory (whichever is encountered first) are transmitted to the host computer as a series of blocks. Each block corresponds to one line in display memory.

ASCII control codes, color enhancement escape sequences, video enhancement escape sequences, alternate character set escape sequences, and field definition escape sequences are all transmitted if encountered.

After each line (except the final one) the terminal sends a CR LF. If the operation is terminated by encountering the end of display memory, the terminal sends a CR LF followed by a block terminator after the last line. If the operation is terminated by encountering a block terminator, the terminal sends only a block terminator after the last line.

If there is no data to be transmitted, the terminal sends only a block terminator.

The type of handshaking used is determined as follows:

InhHndShk(G) (ignored)

Inh DC2(H) = NO → $P_1/P_2/P_1$

Inh DC2(H) = YES → no handshake

BLOCK PAGE MODE, FORMAT MODE

"Block page mode, format mode" means that format mode is enabled and block mode is likewise enabled with the Line/Page (D) field in terminal configuration set to "page".

Inh DC2(H) = YES

The cursor is repositioned to the "home up" position. All unprotected characters through the first subsequent block terminator or through the end of display memory (whichever is encountered first) are

Table 3-2. ENTER Key Operation (Continued)

transmitted to the host computer as a series of blocks. Each block corresponds to one unprotected field.

InhDC2(H) = NO

The cursor is NOT repositioned. All unprotected characters through the first subsequent block terminator or through the end of display memory (whichever is encountered first) are transmitted to the host computer as a series of blocks. Each block corresponds to one unprotected field.

ASCII control codes within the fields are transmitted.

Video enhancement escape sequences, color enhancement escape sequences, alternate character set escape sequences, and field definition escape sequences within the fields are NOT transmitted.

After each field (except the final one), the terminal sends a field separator. After the final field, the terminal sends a block terminator.

If the end of display memory is encountered before locating an unprotected field, the terminal merely sends a block terminator.

The type of handshaking used is determined as follows:

InhHndShk(G) (ignored)

InhDC2(H) = NO → $P_1/P_2/P_1$

InhDC2(H) = YES → no handshake

MODIFY MODE

You enter modify mode through the mode selection keys. (If the "MODES" labels are not currently displayed on the screen, press the MODES key to enable and display these softkey labels.) You may select Line Modify mode by toggling f1 to display an asterisk in the LINE MODIFY label. Similarly, you may select Modify All mode by toggling f2 to display an asterisk in the MODIFY ALL label.

Note that modify line and modify all modes are functional only when the terminal is configured for character mode operation. When either block mode or format is enabled, the ENTER key operates as described for block mode earlier in this table.

In modify mode, the cursor is repositioned as follows:

1. To the logical start-of-text pointer; or
2. To the designated start column (Start Col) if there is no logical start-of-text pointer.

For more information on the logical start-of-text pointer and start column, refer to Table 2-1, "Terminal Configuration Menus Fields" in Section II.

All characters through the first subsequent block terminator or through the end of the line (whichever is encountered first) are transmitted to the host computer as a block.

ASCII control codes, color enhancement escape sequences, video enhancement escape sequences, alternate character set escape sequences, and field definition escape sequences are all transmitted if encountered.

If the operation is terminated by encountering the end of the line, the terminal sends a CR (or a CR LF if auto line feed mode is enabled). If LocalEcho = OFF, the cursor is repositioned to the column at which the transmission began, otherwise the cursor is repositioned to Column 1. A line feed is performed if auto line feed mode is enabled.

If the operation is terminated by encountering a block terminator, the terminal sends a block terminator followed by a CR (or a CR LF if auto line feed mode is enabled). The cursor remains positioned immediately following the terminator.

The type of handshaking used is determined as follows:

InhHndShk(G) = YES

InhDC2(H) = NO → $P_1/P_2/P_1$

Any other combination → no handshake

SEND DISPLAY (Ectd)

From a program executing in a host computer, you can trigger a block transfer of data from display memory to the host computer by issuing the following escape sequence:

Ectd

This escape sequence is only responded to when received over a datacomm line; it is ignored if entered through the keyboard or issued from a user key (unless block mode is enabled). With the following two exceptions, the resultant data transfer is performed as though the ENTER key had been pressed:

1. The cursor is NOT repositioned. The data transfer always begins at the current cursor position.
2. The type of handshaking used is determined as follows:

   InhHndShk(G) = NO → DI handshake
   InhHndShk(G) = YES
   Inh DC2(H) = NO → $D_1/D_2/D_1$
   InhHndShk(G) = YES
   Inh DC2(H) = YES → no handshake

The Ectd sequence also temporarily disables the keyboard so that the ENTER key cannot be used until the current data transfer is completed. If the Ectd sequence is received while an ENTER key data transfer is in progress, the escape sequence is ignored.

Note that an Ectd sequence resets the "block trigger received" flag. This means, for example, that if you are using the D_1 handshake and the terminal receives a D_1 followed by the Ectd, it "forgets" that a block trigger was just received and thus will NOT send the data immediately. The terminal must receive another D_1 before it will start the data transfer.

The amount of data transferred depends on the following terminal settings: Page/Line field in terminal configuration and whether Block Mode is enabled. For more detailed information, refer to Table 3-2 "ENTER Key Operation" in this section.

ENABLE/DISABLE KEYBOARD

You can enable and disable the terminal's keyboard by executing escape sequences. When the keyboard is disabled all keys EXCEPT the following are ignored:

SHIFT CTRL RESET BREAK

The escape sequences for enabling and disabling the keyboard are as follows:

ENABLE: Ectb
DISABLE: Ectc

Once disabled, the keyboard remains disabled until explicitly enabled, until a soft or hard reset is performed, or until the power is turned off.

AUTO KEYBOARD LOCK MODE

Some applications may require that pressing the ENTER key or a Transmit-only ("T") softkey automatically lock the keyboard. This prevents the triggering of future block transfers until the host computer is prepared to accept new data. This can be accomplished by enabling Auto Keyboard Lock mode from the host computer. The following escape sequences enable and disable Auto Keyboard Lock mode.

ENABLE: Ec&k1K
DISABLE: Ec&k0K

Once Auto Keyboard Lock mode is enabled, when either an ENTER Key or a Transmit-only softkey transfer begins, the keyboard becomes locked and remains locked until the host program sends an Ectb sequence to explicitly unlock it.

SOFT RESET

A soft reset does the following:

1. Rings the terminal's bell.
2. Halts any device operations currently in progress.
3. Enables the keyboard (if disabled).
4. Clears any existing error conditions and removes the error message display (if present) from the bottom of the screen.
5. Disables display functions mode (if enabled).
6. Halts any datacomm transfers currently in progress, clears the datacomm buffers.
7. Turns off record mode, if on.

The data on the screen, all terminal operating modes (except display functions mode), and all active configuration parameters are unchanged.

From the keyboard, you perform a soft reset by pressing the RESET key.

From a program executing in a host computer, you perform a soft reset using the following escape sequence:

Ectg

HARD RESET

A hard reset has the same effect as turning the terminal's power off and then back on except that the power-on self-test is not performed.

A hard reset does the following:

1. Rings the terminal's bell.
2. Halts any device operations currently in progress.
3. Enables the keyboard (if disabled).
4. Clears all of alphanumeric memory.
5. Clears any existing error conditions and removes the error message display (if present) from the bottom of the screen.
6. Halts any datacomm transfers currently in progress, clears the datacomm buffer, and reinitializes the datacomm port according to the appropriate power-on datacomm configuration parameters.
7. Resets the terminal configuration menu parameters to values saved in non-volatile memory (their default power-on values).
8. Resets certain operating modes and parameters as follows:

   Disables display functions mode, send cursor position mode, caps mode, auto keyboard lock mode, format mode, data logging, and modify line.

   Resets the left margin to column 1.

   Resets the right margin to column 80.

   Clears all tabs.

   Turns off the "insert character" function edit.

   Resets the User Keys to default values.

   Turns off record mode, if on.

   Turns off foreign characters mode, if enabled.

   Selects Line Drawing as alternate character set.

   Turns on alphanumeric display and homes the cursor up.

   Turns on graphics display.

   Clears graphics memory.

   Turns off graphics cursor and repositions it to 0, 0.

   Enables graphics functions of the graphics/numeric keypad.

   Forces RGB color method.

   Sets all alphanumeric color pairs to their default power-on values.

   Sets all graphics functions to their default power-on values.

From the keyboard, you perform a hard reset by simultaneously pressing the SHIFT, CTRL and RESET keys.

From a program executing in a host computer, you perform a hard reset using the following escape sequence:

Esc E

BREAK

Pressing the BREAK key transmits a 200 ms space on the asynchronous data communications line, or sets the secondary channel low for 200 ms (depending on whether the terminal is in transmit or receive state). This serves as a "break" signal to interrupt computer operation.

BELL

The keyboard includes an embedded speaker for sounding an audible tone in response to the ASCII Bell (<BELL>) control code and for alerting the terminal operator when certain error conditions occur.

From the keyboard, you generate the Bell code by simultaneously pressing the CTRL and G keys.

From a program executing in a host computer, you trigger the bell tone by transmitting an ASCII Bell control code (decimal 7).

WAIT

From a user key or from a program executing in a host computer, you can cause the terminal to pause for approximately 1 second using the following escape sequence:

Esc @

Multiple uses of this escape sequence in succession can be used to obtain virtually any desired time delay.

Note that while an Esc @ is in effect, the cursor disappears from the screen, the keyboard is locked, and the passing of data from the datacomm firmware to display memory is inhibited.

For example, if you want to sound the bell tone twice in succession with a two-second delay between tones, you could do so using the following control sequence:

<BELL> Esc @ Esc @ <BELL>

MODEM DISCONNECT

You can direct the terminal to "hang up" the modem by sending an Esc f. The terminal accomplishes the modem disconnect by lowering the TR/CD (Terminal Ready) line for 2 seconds.

Alphanumeric Display

SECTION IV

INTRODUCTION

This section discusses the alphanumeric display control of the terminal. The graphics display control of the HP 2627A is discussed in Section V.

The display portion of the terminal consists of a display screen and display memory. The display cursor (a blinking underscore mark on the screen) indicates where the next character entered will appear. As you enter characters, each is displayed at the cursor position, the ASCII code for the character is recorded at the associated position in display memory, and the cursor moves to the next character position on the screen. Once the screen becomes full, newly entered data causes existing lines to be forced off the screen. Data lines forced off the screen are still maintained in display memory and can subsequently be moved back onto the screen. After 48 lines of data are entered and display memory becomes full, you must take other steps (data logging, for example) to prevent data from being lost.

You can perform the following display control operations either locally from the keyboard or remotely from a program executing in a host computer:

- Move the cursor up, down, left, or right on the screen.
- Move the displayed data up or down in relation to the current cursor position. When a roll operation forces data off the top or bottom edge of the screen, additional data rolls onto the screen at the opposite edge from display memory.
- Change the content of the screen to the next or previous "page" of data in display memory. (A page consists of 24 lines.)
- Set or clear margins.
- Set or clear one or more tab stop positions.
- Move the cursor forward to the next tab stop position or backward to the preceding tab stop position.
- Enable or disable display enhancements (inverse video, underline, and blinking).
- Select color pair.
- Change the foreground and/or background color of any color pair.
- Change from one character set to another (Base or Line Drawing).

In addition, you can do the following screen edit operations either locally or remotely:

- Delete all characters from the current cursor position through the end of the line.
- Delete all characters from the current cursor position through the end of display memory.
- Delete the line containing the cursor (subsequent lines are rolled up).
- Turn off screen display.
- Turn off softkey labels.
- Delete the character at the current cursor position. Characters to the right of the cursor move left.
- Insert a blank line immediately preceding (above) the line currently containing the cursor.
- Enable or disable "insert character" mode. When this editing mode is enabled, succeeding characters entered through the keyboard or received from the host computer are inserted to the left of the character at the current cursor position.

CURSOR CONTROL

The following topics describe how to alter the cursor/data relationship either manually by using the cursor control keys or programmatically by using escape sequences.

Home Up

Pressing the ↖ key moves the cursor to the left margin in the top row of the screen and rolls the text in display memory down as far as possible so that the first line in display memory appears in the top row of the screen.

When format mode is enabled, the ↖ key also rolls the text down as far as possible but leaves the cursor positioned at the beginning of the first unprotected field. If no fields have been defined, the cursor will appear at the first column of the first row on the screen.

To perform this function programmatically, use the following escape sequence:

Esc h or Esc H

When memory lock is enabled, the ↖ key rolls the text down as far as possible below the locked area of the screen, instead of below the top of the screen, and leaves the cursor positioned at the left margin of the first unlocked row on the screen. When both format mode and memory lock are active simultaneously, the cursor will go to the first unprotected field on the screen (including the locked area), after rolling all the text down.

NOTE

If memory lock is on and the cursor is within the locked area, ↖ will cause the cursor to go down to the first character of the first line of text under the locked area, after rolling the text down.

Home Down

Pressing the SHIFT and ↖ keys moves the cursor to the left margin and rolls the text in display memory up as far as possible without losing any lines. The cursor is positioned immediately below the last line.

To perform this function programmatically, use the following escape sequence:

ESC F

Move Cursor Up

Each time you press the ∧ key, the cursor moves upward one screen row in the current column position. If you hold the key down, the cursor movement continues row-by-row until the key is released. When the cursor is in the top row of the screen, pressing this key moves the cursor to the same column position in the bottom row of the screen. When moving the cursor, data is not altered. Cursor movement is independent of memory lock.

To perform this function programmatically, use the following escape sequence:

ESC A

Move Cursor Down

Each time you press the ∨ key, the cursor moves downward one screen row in the current column position. If you hold the key down, the cursor movement continues row-by-row until the key is released. When the cursor is in the bottom row of the screen, pressing this key moves the cursor to the same column position in the top row of the screen. Moving the cursor does not alter the data on the screen. Cursor movement is independent of memory lock.

To perform this function programmatically, use the following escape sequence:

ESC B

Move Cursor Right

Each time you press the > key, the cursor moves one column to the right in the current screen row. If you hold the key down, the cursor movement continues column-by-column until the key is released.

This function is performed without regard for existing margins. When the cursor reaches the rightmost column of the screen, pressing this key moves the cursor to the leftmost column in the next lower row (from the rightmost column in the bottom row of the screen, the cursor moves to the leftmost column in the top row of the screen). Moving the cursor does not alter the data on the screen. Cursor movement is independent of memory lock.

To perform this function programmatically, use the following escape sequence:

ESC C

Move Cursor Left

Each time you press the < key, the cursor moves one column to the left in the current screen row. If you hold the key down, the cursor movement continues column-by-column until the key is released.

This function is performed without regard for existing margins. When the cursor reaches the leftmost column of the screen, pressing this key moves the cursor to the rightmost column in the next higher row (from the leftmost column in the top row of the screen, the cursor moves to the rightmost column in the bottom row of the screen). Moving the cursor does not alter the data on the screen. Cursor movement is independent of memory lock.

To perform this function programmatically, use the following escape sequence:

ESC D

Roll Text Up

Each time you press the ROLL UP key, the text in display memory rolls up one row on the screen. The top row rolls off the screen (but this data is not lost), the remaining data rolls up one line on the screen, and a new line of data rolls from display memory into the bottom line of the screen. If you hold this key down, the text continues to roll upward until you release the key or until the final line of data in display memory appears in the top row of the screen. In the latter case, pressing or continuing to hold down the key has no further effect. The "roll up" function is illustrated in figure 4-1A.

In the configuration and user softkey definition menus, this key is disabled. In memory lock mode, the unlocked text rolls behind the locked text, as if the bottom line of the locked text is the top of the screen.

To perform this function programmatically, use the following escape sequence:

ESC S

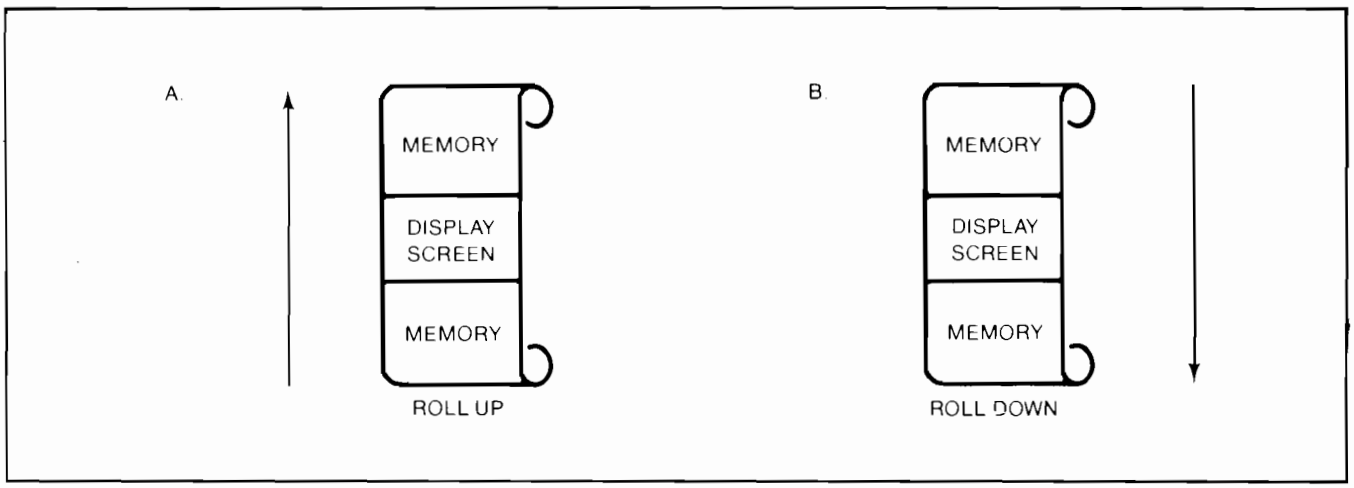


Figure 4-1. The "Roll" Data Functions

Roll Text Down

Each time you press the Roll Down key, the text in display memory rolls down one row on the screen. The bottom row rolls off the screen (but the data is not lost), the remaining data rolls down one line on the screen, and a new line of data rolls from display memory into the top line of the screen. If you hold this key down, the text continues to roll downward until you release the key or until the first line of data in display memory appears in the top row of the screen. In the latter case, pressing or continuing to hold down the key has no further effect. The "roll down" function is illustrated in figure 4-1B.

In the configuration and user softkey definition menus, this key is disabled. In memory lock mode, the unlocked text rolls behind the locked text, as if the bottom line of the locked text is the top of the screen. The cursor position relative to the screen remains unchanged after this operation.

To perform this function programmatically, use the following escape sequence:

Esc T

Next Page/Previous Page

You may access the data stored in display memory in blocks that are known as "pages." A page consists of the number of unlocked rows of screen data. The maximum size for all pages is 24 lines. The current page is that sequence of lines which appears on the screen at any given time. If Memory Lock protects the screen's first three lines, a page consists of 21 lines of text. The previous page is the preceding displayable portion of display memory. The next page is the succeeding displayable portion of display memory.

The concept of display "pages" is illustrated in figure 4-2.

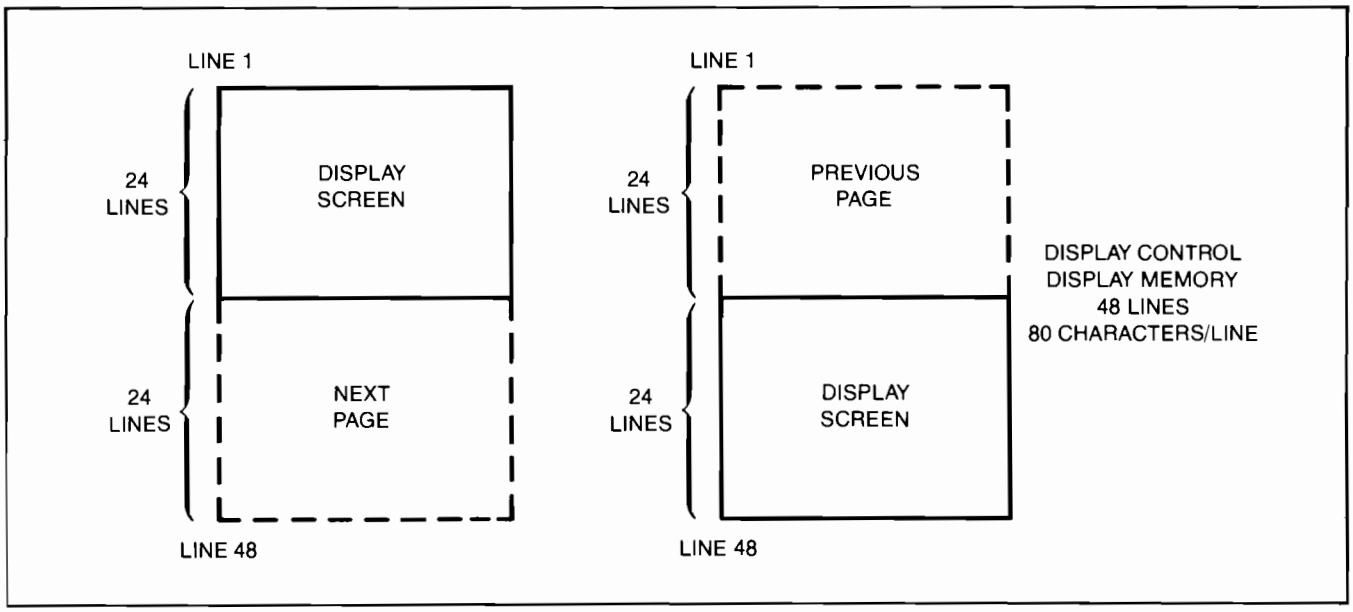


Figure 4-2. Previous Page and Next Page Concepts

Pressing the NEXT PAGE key rolls the text in display memory up so that the next page of data replaces the current page on the screen. If you hold the key down, the operation is repeated until you release the key or until the final line in display memory appears in the first unlocked screen row. In the latter case, pressing or continuing to hold down the key has no further effect.

In the configuration and user softkey definition menus, this key is disabled.

To perform the "next page" function programmatically, use the following escape sequence:

ESC U

Pressing the PREV PAGE key rolls the text in display memory down so that the previous page of data replaces the current page on the screen. If you hold the key down, the operation is repeated until you release the key or until the first line in display memory appears in the top line of the screen. In the latter case, pressing or continuing to hold down the key has no further effect.

In the configuration and user softkey definition menus, this key is disabled.

To perform the "previous page" function programmatically, use the following escape sequence:

ESC V

At the completion of the "next page" or "previous page" function, the cursor is positioned at the left margin in the top line of the screen (the first unlocked screen row).

If format mode is on, the cursor will go to the first unprotected field on the new page.

MEMORY ADDRESSING SCHEME

Display memory positions can be addressed using absolute or relative coordinate values. Display memory is made up of 80 columns (0–79) and 48 lines (0–47). All escape sequences operate on this zero-based addressing scheme. Since the row/column indicator on the screen shows the screen's 24 rows as "1" through "24" (to "48" for two pages of display memory) and the screen's 80 column positions as "1" through "80", it becomes the programmer's responsibility to supply all necessary conversions to ensure correct positioning. Display memory may contain a maximum of 48 lines of 80 characters each (2 full screens).

The types of addressing available are:

- Absolute
- Screen Relative
- Cursor Relative

ROW ADDRESSING. Figure 4-3 illustrates the way the three types of addressing affect row or line numbers. The cursor is shown positioned in the fourth row on the screen. Screen row 0 is currently at row 6 of display memory. In order to reposition the cursor to the first line of the screen the following three destination rows could be used:

a. Absolute: row 6
b. Screen Relative: row 0
c. Cursor Relative: row −3

COLUMN ADDRESSING. Column addressing is accomplished in a manner similar to row addressing. There is no difference between screen and cursor relative column addressing. Figure 4-4 illustrates the difference between absolute and relative addressing. The cursor is shown in column 5.

Whenever the row or column addresses exceed those available, the largest possible value is substituted. In screen relative addressing, the cursor cannot be moved to a row position that is not currently displayed. For example, in figure 4-3c a relative row address of −10 would cause the cursor to be positioned at the top of the current screen (relative row −3). Column positions are limited to the

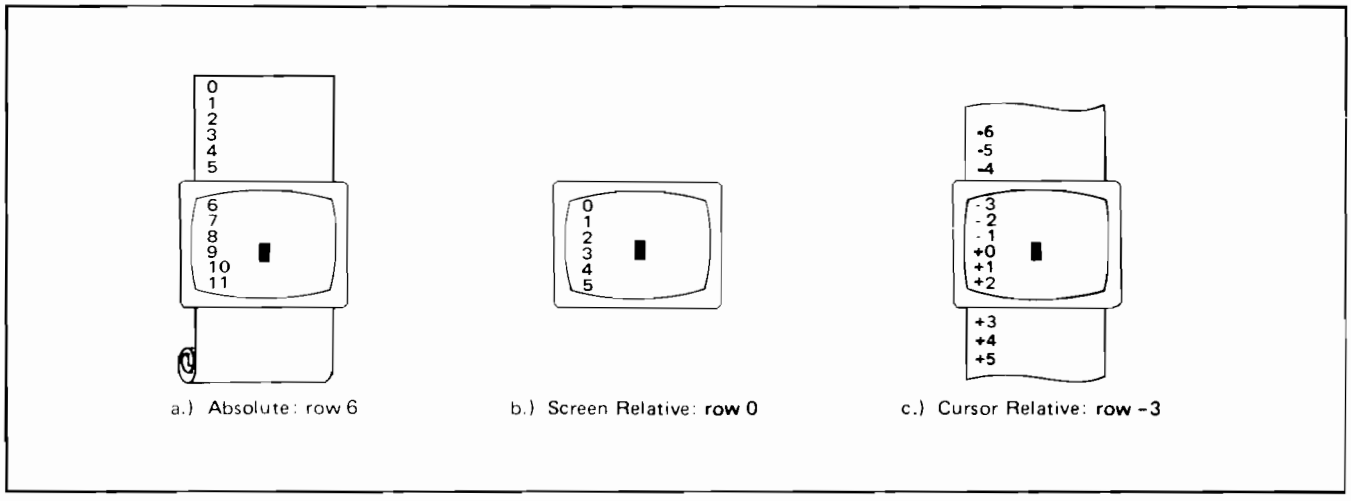


Figure 4-3. Row Addressing

available screen positions (0 to 79 in figure 4-4c and −5 to 74 in figure 4-4b). The cursor cannot be wrapped around from column 0 to column 79 by specifying large negative values for relative column positions.

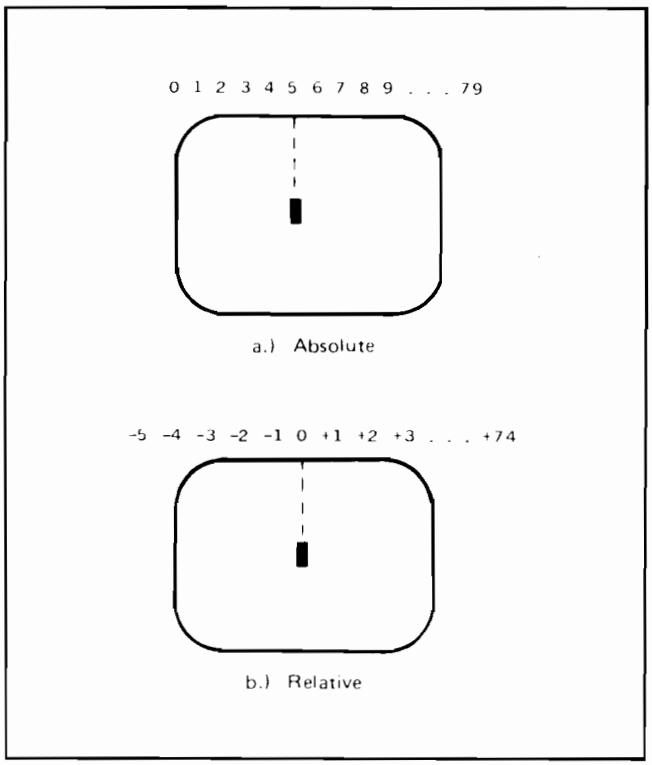


Figure 4-4. Column Addressing

Cursor Sensing

The current position of the screen cursor can be sensed. The position returned can be the absolute position in display memory or the location relative to the current screen position. (Absolute and relative addresses are discussed under Cursor Addressing.)

Absolute Sensing Ec a

Example: The row/column indicator shows the cursor at column 20, row 40.

computer: Ec a
terminal: Ec & a 019 c 039 R

Relative Sensing Ec `

Example: The cursor is again at column 20, row 40, but screen row 0 begins at row 35 of display memory.

computer: Ec `
terminal: Ec & a 019 c 005 Y

Send Cursor Position Mode

To institute a block transfer, the terminal repositions the cursor before it transmits the data block. However, in certain applications, it may be necessary to know the cursor's location when the transfer is initiated. In Send Cursor Position (SCP) Mode, when the ENTER key or a Transmit-only user softkey is pressed, the cursor's present position is inserted at the beginning of the block of text. The format of the cursor's position is identical to that returned from an absolute cursor position sense status request. The following rules apply:

1. You may only enter SCP mode programmatically through escape sequences.
   To enable SCP mode: Ec & x 1 C
   To disable SCP mode: Ec & x 0 C

2. The terminal must be in Block mode or Forms mode. SCP mode does not apply to Line Modify or Modify All.

3. A soft reset leaves SCP mode unchanged. A hard reset disables SCP mode (the default condition).

Cursor Positioning

The cursor can be positioned directly by giving memory or screen coordinates, or by sending the escape codes for any of the keyboard cursor positioning operations.

Screen Relative Addressing

To move the cursor to any character position on the screen, use any of the following escape sequences:

Ec & a <column number> c <row number> Y
Ec & a <row number> y <column number> C
Ec & a <column number> C
Ec & a <row number> Y

where:

<column number> is a decimal number specifying the screen column (0–79) to which you wish to move the cursor. Zero specifies the leftmost column, 79 the rightmost column. Values <0 default to 0 and values >79 default to 79.

<row number> is a decimal number specifying the screen row (0–23) to which you wish to move the cursor. Zero specifies the top row of the screen; 23 specifies the bottom row. Values <0 default to 0 and values >23 default to 23.

When using the above escape sequences, the data on the screen always remains unchanged. Screen relative addressing is independent of memory lock.

If you specify only a <column number>, the cursor remains in the current row. Similarly, if you specify only a <row number>, the cursor remains in the current column.

Example: The following escape sequence moves the cursor to the 20th column of the 7th row on the screen:

Ec & a 6 y 19 C

Absolute Addressing

You can specify the location of any character within display memory by supplying absolute row and column coordinates. (When memory lock mode is on, the row portion of the escape sequence is ignored and the column portion is executed.) To move the cursor to another character position using absolute addressing, use any of the following escape sequences:

```
Ec&a <column number> c <row number> R
Ec&a <row number> r <column number> C
Ec&a <column number> C
Ec&a <row number> R
```

where:

`<column number>` is a decimal number (0–79) specifying the column coordinate (within display memory) of the character at which you want the cursor positioned. Zero specifies the first (leftmost) column in display memory, 79 the rightmost column. Values >79 default to 79.

`<row number>` is a decimal number (0–47) specifying the row coordinate (within display memory) of the character at which you want the cursor positioned. Zero specifies the first (top) row in display memory, 47 specifies the last. Values >47 result in data being lost and the cursor repositioned to row 47.

When using the above escape sequences, the data visible on the screen will (if necessary) be rolled up or down in order to position the cursor at the specified data character. The cursor and data movement will occur as follows:

- If a specified character position lies within the boundaries of the screen, the cursor moves to that position and the data on the screen remains unchanged.
- If the absolute row coordinate is less than that of the top line currently visible on the screen, the cursor moves to the specified column in the top row of the screen and the text rolls downward until the specified row appears in the top line of the screen.
- If the absolute row coordinate exceeds that of the bottom line currently visible on the screen, the cursor moves to the specified column in the bottom row of the screen and the text rolls upward until the specified row appears in the bottom line of the screen. Data may be lost if the row number is greater than 47.

If you specify only a `<column number>`, the cursor remains in the current row. Similarly, if you specify only a `<row number>`, the cursor remains in the current column.

Example: The following escape sequence moves the cursor (and rolls the text if neccessary) so that it is positioned at the character residing in the 60th column of the 27th row in display memory:

```
Ec&a 26r 59C
```

Cursor Relative Addressing

You can specify the location of any character within display memory by supplying row and column coordinates that are relative to the current cursor position. (When memory lock mode is on, the row portion of the escape sequence is ignored.) To move the cursor to another character position using cursor relative addressing, use any of the following escape sequences:

```
Ec&a +/- <column number> c +/- <row number> R
Ec&a +/- <row number> r +/- <column number> C
Ec&a +/- <column number> C
Ec&a +/- <row number> R
```

where:

`<column number>` is a decimal number specifying the relative column to which you wish to move the cursor. A positive number specifies how many columns to the right you wish to move the cursor; a negative number specifies how many columns to the left.

`<row number>` is a decimal number specifying the relative row to which you wish to move the cursor. A positive number specifies how many rows down you wish to move the cursor; a negative number specifies how many rows upward.

When using the preceding escape sequences, the data will (if necessary) be rolled up or down in order to position the cursor at the specified data character. The cursor and data movement will occur as follows:

- If a specified character position lies within the boundaries of the screen, the cursor moves to that position and the data on the screen remains unchanged.
- If the specified cursor relative row precedes the top line currently visible on the screen, the cursor moves to the specified column in the top row of the screen and the text rolls downward until the specified row appears in the top line of the screen, or the top of display memory is found (whichever occurs first).
- If the specified cursor relative row follows the bottom line currently visible on the screen, the cursor moves to the specified column in the bottom row of the screen and the text rolls upward until the specified row appears in the bottom line of the screen.

NOTE

Display memory may be lost if the row number is specified outside the set bounds of display memory (48 lines).

If you specify only a `<column number>` the cursor remains in the current row. Similarly, if you specify only a `<row number>` the cursor remains in the current column.

Example: The following escape sequence moves the cursor (and rolls the text if necessary) so that it is

positioned at the character residing 15 columns to the right and 25 rows above the current cursor position within display memory:

Esc&a +15c -25R

Combining Absolute And Relative Addressing

You may use a combination of screen relative, absolute and cursor relative addressing within a single escape sequence.

Example: Move the cursor (and roll the text if necessary) so that it is positioned at the character residing in the 70th column of the 18th row below the current cursor position.

Esc&a 69c +18R

Example: Move the cursor so that it is positioned at the character residing 15 columns to the left of the current cursor position in the 4th row currently visible on the screen.

Esc&a -15c 3Y

Example: Move the cursor (and roll the text up or down if necessary) so that it is positioned at the character residing in the 10th column of absolute row 48 in display memory.

Esc&a 9c 47R

EDIT OPERATIONS

You can edit data displayed on the screen by simply overstriking the old data. In addition, the terminal provides the following edit functions which can be enabled and disabled either manually by using the edit control keys or programmatically by using escape sequences:

- Insert Line.
- Delete Line.
- Insert Character.
- Delete Character.
- Clear Display.
- Clear Line.

Insert Line

When you use the insert line edit function, the text line containing the cursor and all text lines below it roll downward one line, a blank line is inserted in the screen row containing the cursor, and the cursor moves to the left margin of the blank line. Note that when memory lock mode is active, inserting a line within the locked area of the screen does not change the size of the locked area.

From the keyboard, each time you press the INS LINE key the terminal inserts one blank line. If you hold the key down, the terminal continues to insert blank lines until the key is released.

This edit function is disabled in format mode, and is disabled in the configuration and user softkeys definition menus.

NOTE

When the 48 lines of display memory are full, inserting a line will cause data to be lost. The first line in display memory will always be the one to be released unless it happens to be on the screen, in which case the last line in display memory will be released.

From a program executing in a host computer, you insert a blank line at the current cursor position using the following escape sequence:

EscL

Delete Line

When you use the delete line edit function, the text line containing the cursor is deleted from display memory, all text lines below it roll upward one row, and the cursor moves to the left margin. Note that when memory lock mode is active, deleting a line within the locked area does not change the size of the locked area.

From the keyboard, each time you press the DEL LINE key the terminal deletes one line of text. If you hold the key down, the terminal continues to delete text lines until the key is released or until there are no subsequent text lines remaining in display memory. In the latter case, pressing or continuing to hold down this key has no further effect.

This edit function is disabled in format mode, and is disabled in the configuration and user softkeys definition menus.

From a program executing in a host computer, you delete the text line at the current cursor position using the following escape sequence:

EscM

Insert Character

When the insert character editing function is enabled, characters entered through the keyboard or received from the host computer are inserted into display memory at the cursor position. Each time a character is inserted, the cursor and all characters from the current cursor position through the right margin move one column to the right. Characters that are forced over the right margin are lost. When the cursor reaches the right margin, it moves to the

left margin in the next lower line and the insert character function continues from that point.

This edit function is meant to be used within that portion of the screen delineated by the left and right margins. If you position the cursor to the left of the left margin, the insert character function works as described above. If you position the cursor beyond the right margin, however, the insert character function affects those characters between the current cursor position and the right boundary of the screen. In such a case, when the cursor reaches the right boundary of the screen, it moves to the left margin in the next lower line and the insert character function continues from that point as described in the first paragraph above.

The movement of existing characters during an "insert character" editing operation is illustrated in figure 4-5.

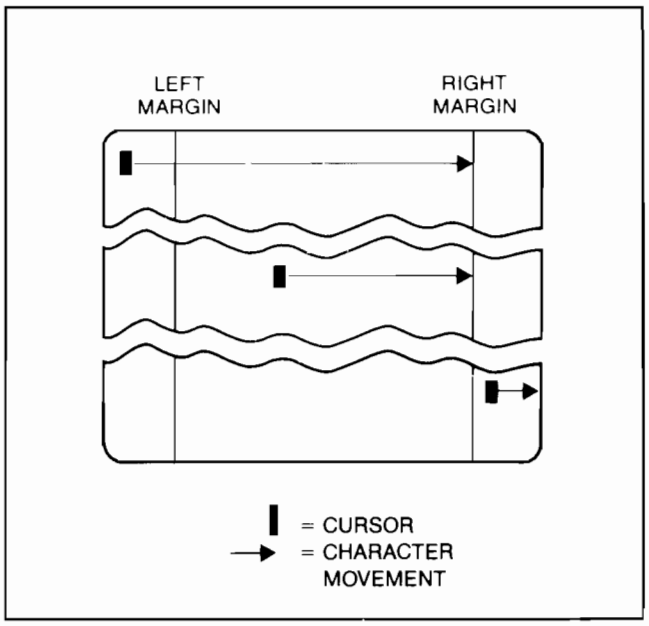


Figure 4-5. Character Insert with Margins

When format mode is off, any unprotected alternate character set, color enhancements, and/or video enhancement fields to the right of the cursor move to the right with the displayable characters. If the cursor is positioned within any such field the insert character function extends the range of the field by one position for each character inserted. Block terminators at or to the right of the cursor position move to the right along with the displayable characters.

When format mode is on and the cursor is positioned within an unprotected field, the insert character function affects only those characters from the cursor position through the end of the current subfield. Block terminators are treated the same as when format mode is off. If the cursor is not within an unprotected field, it automatically moves to the first character position of the next subsequent unprotected field when the first character is inserted.

In the user softkeys definition menu, insert character acts the same as in format mode; insert character is disabled in a configuration menu.

From the keyboard, you enable and disable the insert character editing function using the INS CHAR key. When enabled, the characters "IC" are displayed in the status line at the bottom of the screen.

From a program executing in a host computer, you enable and disable the insert character editing function using the following escape sequences:

ENABLE: ESC Q
DISABLE: ESC R

Delete Character

When you use the delete character edit function, the cursor remains stationary, the character at the cursor position is deleted, all characters between the cursor and the right margin move left one column, and a blank moves into the line from the right margin.

This edit function is meant to be used within that portion of the screen delineated by the left and right margins. If you position the cursor to the left of the left margin, the delete character function works as described above. If you position the cursor beyond the right margin, however, the delete character function affects those characters from the current cursor position through the right boundary of the screen.

The movement of existing characters during a "delete character" editing operation is illustrated in figure 4-6.

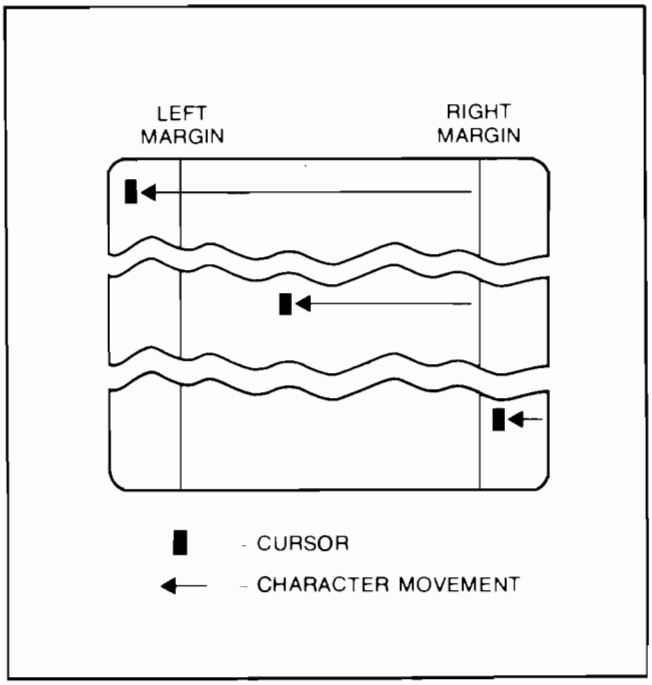


Figure 4-6. Character Delete with Margins

When format mode is off, any unprotected alternate character set, color enhancements, and/or video enhancement fields to the right of the cursor move to the left with the displayable characters. If the cursor is positioned within any such field, the delete character function shortens the range of the field by one position for each character deleted. Deleting the first character position of an unprotected field changes the rest of the field to protected. Deleting characters at the start of, or within, a color enhancement, video enhancement, and/or alternate character set field does NOT alter the characteristics of the rest of the field. Block terminators to the right of the cursor move to the left along with the displayable characters and are deleted if they are at the cursor position when this function is executed.

When format mode is on and the cursor is positioned within an unprotected field, this function affects only those characters from the cursor position through the end of the current subfield. If the subfield definition also includes a color enhancement, video enhancement, and/or an alternate character set, those characteristics are NOT altered by the delete character function. Block terminators are treated the same as when format mode is off. If the cursor is not within an unprotected field, the delete character function has no effect.

In the user softkeys definition menu, delete character acts the same as in format mode; delete character is disabled in a configuration menu.

From the keyboard, each time you press the DEL CHAR key the terminal deletes one character. If you hold the key down, the terminal continues to delete characters until either the key is released or there are no non-blank characters between the cursor position and the right margin. In the latter case, pressing or continuing to hold down this key has no further effect.

From a program executing in a host computer, you delete the character at the current cursor position using the following escape sequence:

ESC P

Clear Display

When format mode is off, pressing the CLEAR DSPLY key deletes all displaying and non-displaying characters, all color enhancements and video enhancements, and any line drawing characters from the current cursor position through the end of display memory.

When format mode is on, pressing the CLEAR DSPLY key deletes all unprotected displaying and non-displaying characters, and any unprotected line drawing characters from the current cursor position through the end of display memory. However, all color and video enhancements are protected. CLEAR DSPLY will not remove them, whether they are inside or outside an unprotected field.

This key is disabled in the user softkeys definition and configuration menus.

To perform this function programmatically, use the following escape sequence:

ESC J

Clear Line

When format mode is off, pressing the CLEAR LINE key deletes all displaying and non-displaying characters, all color enhancements and video enhancements, and any line drawing characters from the current cursor position through the end of the current line.

NOTE

Unlike delete line, display memory does not roll up when a row is cleared.

When format mode is on and the cursor is positioned within an unprotected field, pressing the CLEAR LINE key deletes all displaying and non-displaying characters and any line drawing characters from the current cursor position through the end of the current field. However, all color and video enhancements are protected, whether they are inside or outside an unprotected field. If the cursor is not within an unprotected field, the CLEAR LINE key has no effect.

In the user softkeys definition menu, clear line acts the same as in format mode; clear line is disabled in the configuration menus.

To perform this function programmatically, use the following escape sequence:

ESC K

SETTING AND CLEARING MARGINS

You can redefine the left and/or right margin. These margins affect the cursor positioning for certain functions (such as carriage return, home up, home down, etc.) and establish operational bounds for the insert character and delete character functions. In addition, the left margin is always an implicit tab stop. Data to the left of the left margin or to the right of the right margin is still accessible. Data transfers from display memory to a host computer or to a printer are performed without regard to margins. Format mode, when enabled, clears the margins, creating an 80 character line (shown as "1"-"80" by the row/column indicator).

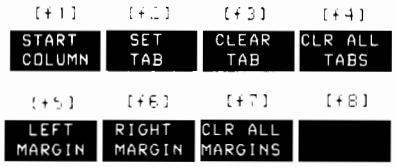
When you are entering data through the keyboard and the cursor reaches the right margin, it automatically moves to the left margin in the next lower line (note that this operating characteristic can be disabled through the use of the "InhEolWrp" terminal configuration parameter; refer to Section II). When you press RETURN, the cursor moves to the left margin in the current line if auto line feed mode is disabled or to the left margin in the next lower line if auto line feed mode is enabled.

When data is being received from a host computer, it enters display memory only within the defined margins. When the cursor reaches the right margin, it automatically

moves to the left margin in the next lower line (as mentioned above, this operating characteristic can be disabled through the use of the "InhEolWrp" configuration parameter). When an ASCII ␍ control code is received, the cursor always moves to the left margin in the current line regardless of whether or not auto line feed mode is enabled.

From the keyboard, you set and clear the margins using the margins/tabs/col set of function keys. To get to that set, use the following keystroke sequence:

This changes the function key labels to the following:



To set the left or right margin, move the cursor to the desired column and then press the appropriate function key (f5 or f6). To reset the left margin to column 1 and the right margin to column 80, press f7 .

If you attempt to set either margin incorrectly with relation to the other (e.g., the right margin to the left of the left margin), the terminal rejects it with an audible "beep".

From a program executing in a host computer, you set and clear the margins using the following escape sequences:

SET LEFT MARGIN: ␛&a4
SET RIGHT MARGIN: ␛&a5
CLEAR ALL MARGINS: ␛&a9

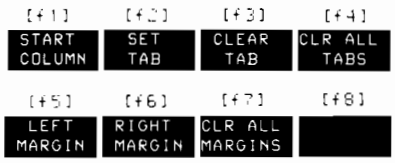
The first two escape sequences set the left and right margin (respectively) at the current cursor position. Before using them, therefore, you will first have to position the cursor at the desired column using one of the cursor control escape sequences described earlier in this section.

SETTING AND CLEARING TABS

You can define a series of tab stops to which you can move the cursor using the tab and back tab functions (described as separate topics later in this section).

From the keyboard, you set and clear tab stops using the margins/tabs/col set of function keys. To get to that set, use the following keystroke sequence:

This changes the function key labels to the following:



To set a tab stop, move the cursor to the desired column and then press f2 . To clear a tab stop, move the cursor to the particular tab stop position and then press f3 . To clear all existing tab stops, press f4 . Note that the left margin is always an implicit tab stop and is not affected by f4 .

Tab stops that do NOT lie within the area bounded by the left and right margins are ignored when the tab or back tab functions are performed.

From a program executing in a host computer, you set and clear tab stops using the following escape sequences:

SET TAB: ␛&a1
CLEAR TAB: ␛&a2
CLEAR ALL TABS: ␛&a3

The first two escape sequences set and clear (respectively) a tab stop at the current cursor position. Before using them, therefore, you will first have to position the cursor at the desired column using one of the cursor control escape sequences described earlier in this section.

TAB

From the keyboard, you can move the cursor ahead to the next subsequent tab stop using the TAB key. (If the graphics/numeric keypad is enabled for numeric operations, the TAB key also advances the cursor to the next tab stop.) In format mode, tab moves the cursor to the beginning of the next unprotected field. The last field wraps around to the beginning of the first field. Tab acts similarly in the user softkeys definition menu and the configuration menus.

From a program executing in a host computer, you can move the cursor ahead to the next tab stop issuing either an ASCII ␉ control code (Control "I") or the following escape sequence:

␛&aI

Tab stops that do NOT lie within the area bounded by the left and right margins are ignored by the tab function.

Note that the left margin is treated as a tab stop. When the cursor is positioned at or to the right of the rightmost tab stop, the tab function moves the cursor to the left margin in the next lower line. When the cursor is positioned to the left of the left margin, however, the tab function advances the cursor to the first explicit tab stop in the line (or to the left

margin in the next lower line if no explicit tab stops are defined). Note that tabbing the cursor to the next line is the equivalent of a linefeed.

BACK TAB

From the keyboard you can move the cursor backward to the previous tab stop using the SHIFT and TAB keys (or the TAB key in the numeric pad).

In format mode, configuration menus, and user keys definition menu, the cursor, if within a field, will move to the beginning of the field; otherwise it will move to the first character of the previous unprotected field. However, the first field does not wrap around to the last field.

From a program executing in a host computer you can move the cursor backward to the previous tab stop using the following escape sequence:

Esc i

Tab stops that do NOT lie within the area bounded by the left and right margins are ignored by the back tab function.

Note that the left margin is treated as a tab stop. When the cursor is positioned at or to the left of the left margin, the back tab function moves the cursor to the rightmost tab stop in the next higher line.

Performing a back tab with the cursor on the left margin of the first row on the screen (or the first unlocked row if memory lock mode is active) is equivalent to performing a roll down.

SCREEN BLANKING

Either manually or from a program executing in a host computer, you can turn off and on the alphanumeric display video. The same action may or may not remove the softkey labels. You can choose from the following options:

1. You may programmatically toggle alpha display video, while retaining the softkey labels.
   Turn off display video: Esc&w13F
   Turn on display video: Esc&w12F
2. You can manually remove the softkey labels by pressing SHIFT AIDS or programmatically remove them with the escape sequence Esc&j@ . In either case, the alpha display area (screen's first 24 lines) is unaffected.
3. You can blank the entire screen (alpha display and softkey labels) either manually by pressing the key ALPHA DSPLY in the graphics-enabled keypad or programmatically.
   Turn off entire alpha screen: Esc*dF
   Turn on entire alpha screen: Esc*dE

DISPLAY ENHANCEMENTS

The terminal includes as a standard feature the following display enhancement capabilities:

- Inverse Video—foreground and background colors are swapped.
- Underline Video—characters are underscored.
- Blink Video—characters blink on and off.
- Half Bright—Half-bright is shown as color pair "3" when displayed. (See "Selecting Color Pairs", this section).

You use these enhancements on a field basis. They may be used separately or in any combination. When used, they cause control bits to be set within display memory. If the content of display memory is subsequently transmitted in block mode to a host computer, these control bits are translated into escape sequences which are transmitted along with the displayable text characters.

From a program executing in a host computer, or from the keyboard, you enable and disable the various video enhancements by embedding escape sequences within the data. The general form of the escape sequence is as follows:

Esc&d <enhancement code>

where enhancement code is one of the uppercase letters A through O specifying the desired enhancement(s) or an @ to specify end of enhancement as follows:

Enhancement Character

	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Half-Bright Mapping									x	x	x	x	x	x	x	x
Underline					x	x	x	x					x	x	x	x
Inverse Video			x	x			x	x			x	x			x	x
Blinking		x		x		x		x		x		x		x		x
End Enhancement	x															

Note that the escape sequence for "end enhancement" (Ec&d @) or the escape sequence for another video enhancement, will end the previous enhancement.

Example: Define columns 10 through 14 of line 5 to be inverse video and blinking.

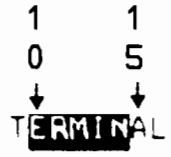
Step 1. Position the cursor at column 10 in line 5.

Step 2. Enter Ec&d C.

Step 3. Move the cursor to column 15 in line 5.

Step 4. Enter Ec&d @ (this ends the enhancements). The field should be white.

Step 5. Enter the word TERMINAL beginning in column 9 of line 5. It should appear as shown below. The characters "ERMIN" should be in inverse video and blinking.



You may want to enter some frequently used enhancements into the user keys for ease in entering the enhancements onto the display.

Example: Enter Underline, Blinking Inverse, Inverse, and End Enhancement escape sequences into the f1 through f4 user keys. (Figure 4-7).

Step 1. Press SHIFT USER KEYS to display the menu.

Step 2. Press "NEXT CHOICE" f2 until "L" (for local operation), is displayed in the attribute field of f1.

Step 3. Press TAB to position the cursor to the label field; then, type a meaningful label to represent the enhancement (e.g., "UNDER" in the first label and "LINE" in the second label field).

Step 4. Press TAB to position the cursor to the definition field; then, press the "DISPLY FUNCTN" key. Now, enter the escape sequence for the enhancement; then, turn off display functions. (Display functions is turned on to enter the "Ec", escape, character.)

Step 5. Tab to the next key fields, and enter the appropriate data in a similar manner.

Step 6. When you have finished entering the data into the menu, press USER KEYS to return to the normal display with the user key labels.

Step 7. You many now turn on any enhancement specified in the user keys by positioning the cursor to where you want the enhancement to begin; then, press the appropriate user key. To turn off the enhancement, press the user key containing the end enhancement escape sequence.

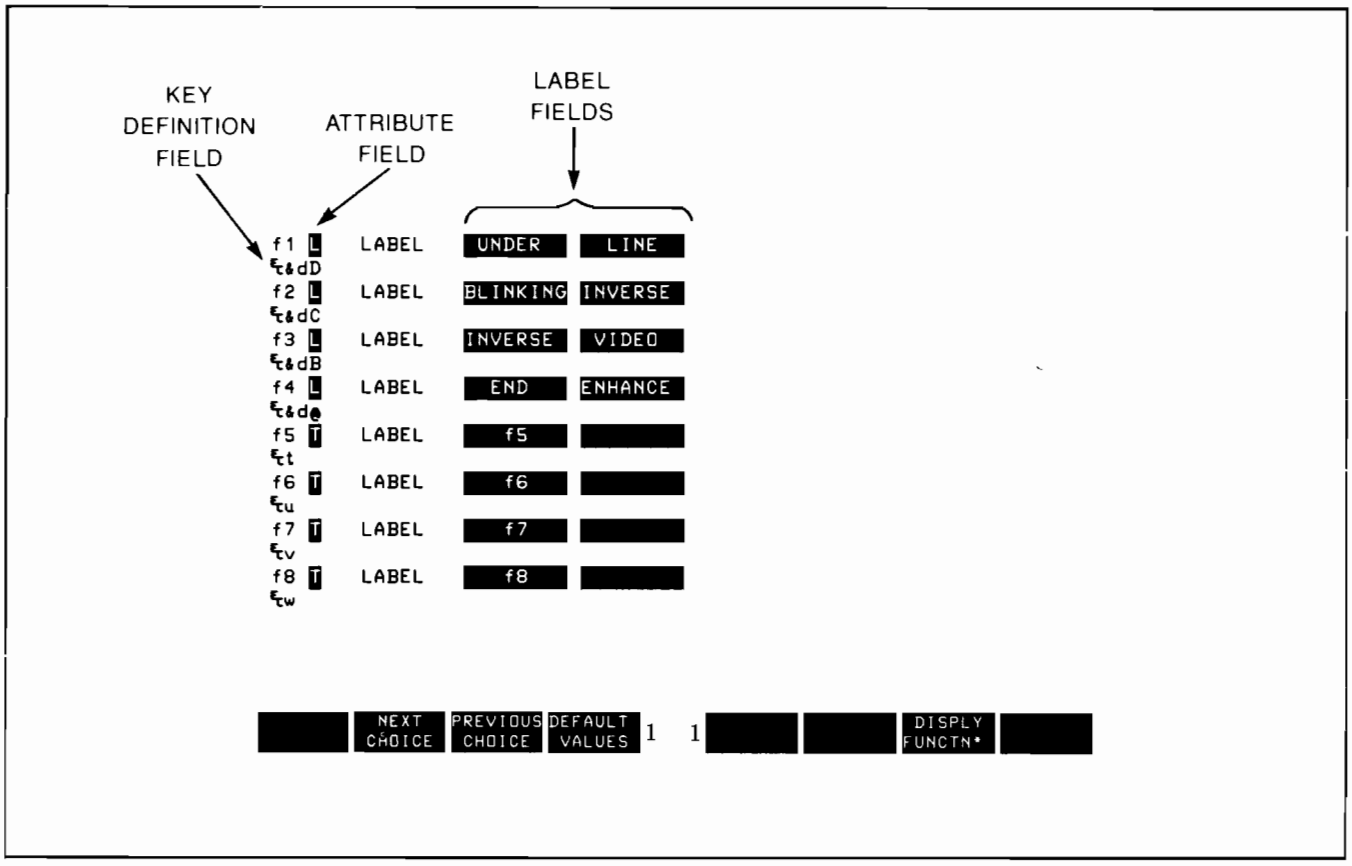


Figure 4-7. Menu After Entering Four Enhancement Escape Sequences

SELECTING COLOR PAIRS

The terminal has eight programmable color pairs that are available as alphanumeric color enhancements on a per character basis. Each color pair consists of a foreground and a background color. Each foreground and background color can be programmed through escape sequences to select any of the basic eight colors. A total of 64 color pairs are possible, but only eight can be displayed at any one time. The eight default alphanumeric color pairs are as follows:

Color Pair	Foreground Color	Background Color
0	White	Black
1	Red	Black
2	Green	Black
3	Yellow	Black
4	Blue	Black
5	Magenta	Black
6	Cyan	Black
7	Black	Yellow

When no other color enhancement is selected, color pair "0" is the default. Thus, at power on, the display appears as white characters on a black screen. Also at power on time, the softkey labels are displayed in color pair "7", which means black alphanumeric characters are displayed with a yellow background. The half-bright enhancement is displayed in color pair "3".

From a host computer, or from the keyboard, you can define alphanumeric color pairs using the following escape sequence:

Select Color Pair: Esc & v <parameters>

where the <parameters> consist of one or more of the color commands:

Parameter	Description
<0/1>m	Color specification method (RGB/HSL)
<decimal>a	Red (or hue) color value for foreground
<decimal>b	Green (or saturation) color value for foreground
<decimal>c	Blue (or luminosity) color value for foreground
<decimal>x	Red (or hue) color value for background
<decimal>y	Green (or saturation) color value for background
<decimal>z	Blue (or luminosity) color value for background
<0-7>i	Color pair # to be initialized
<0-7>s	Color pair # to be selected
<0-7>^	Color pair definition status

NOTES

1. The default value for all the above color parameters, except STATUS, is 0.
2. Color parameter values (a,b,c,x,y,z) are in the range 0 to 1 and are truncated to hundredths (.01). All values outside the range will be ignored.
3. Color terminology (RGB/HSL) is discussed in Appendix A, "Color Technology".
4. Color pair status is discussed in Section VIII, "Status". The "^" terminator is interpreted as a "capital letter" and must always be placed at the end of the escape sequence when it is combined with other parameters.
5. Multiple color pairs can be defined at one time in the same escape sequence. Different parameters (m, i, s, and ^) can co-exist in one escape sequence.
6. The different parameters (m, i, s, and ^) are independent of each other. An escape sequence can contain any one or all. When the "i" parameter is present, it must immediately follow the foreground (a, b, c) and background (x, y, z) color values. Any missing foreground/background parameter value is assumed to be a "zero" value.

You can define color pairs using the following guide:

- Select the color method
- Initialize a color pair
- Select a color pair

Color Method

Alpha colors are specified using either RGB or HSL color notation. The color method is set with the following color parameter:

<0/1>m where 0=RGB, 1=HSL

The color method parameter tells the terminal how it should interpret subsequent color selection parameters (a,b,c,x,y,z). If no method parameter is present in the escape sequence containing a color specification, the terminal will use the last method entered. (The method selected will remain in effect until changed, a hard reset is performed, or the terminal is powered off then on. The terminal sets the RGB method as the default at power on.

NOTE

Group #1 Group #2

The Esc & v 0m . . . 1m . . . sequence interprets group #1 using the RGB method and group #2 using the HSL method.

Example: Set the color method to RGB.

Esc & v 0M

The Esc & v 0M sequence only sets the method to RGB. It does nothing more.

Initializing Color Pairs

Colors are assigned to specific color pairs by sending color commands followed by a color pair # assignment.

Esc & v <>a<>b<>c <>x<>y<>z <color pair #> I

Foregound Background

The foreground and background color parameters must immediately precede their associated color pair value. Any missing parameters are assumed to be zero.

For example (with the RGB color method):

```
Et&v1a1x1y1z2i1a4i6I
```

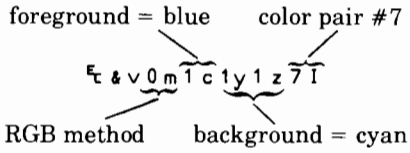
initializes color pair 2 as red characters on a white background; color pair 4 as red characters on a black background (all background parameters are zero); and color pair 6 as black characters on a black background! (All parameters for both foreground and background are zero).

Colors are specified using RGB or HSL methods. The R, G, B values control the red, green, and blue elements of the terminal's display. When the HSL method is used, the H value selects a particular hue, the S value controls the saturation of the color, and the L value controls the luminosity of the color. The RGB and HSL parameters < > do not have to be integers, but can have values in the range of 0 to 1 (inclusively) in .01 increments.

The user-definable softkey labels are normally displayed in color pair "7"; however, they can be specified in other color pairs. Refer to Section III, "Keyboard Control".

Example: Using the RGB method, specify a foreground color of blue and a background color of cyan. Assign the colors to color pair "7".

Blue = blue(Foreground)
Cyan = green + blue(Background)



NOTE

Alphanumeric characters may overlay or mask out the graphics display. To allow graphics data to be seen, select black background colors for the alphanumeric color pairs. The current color assignments can be displayed by executing a terminal self-test. The character set display will be shown using the eight alpha color pairs (see Section IX).

Color Pair Selection

Once colors have been assigned to the color pairs, you can select the color pairs to be used for alpha characters with the color pair selection command.

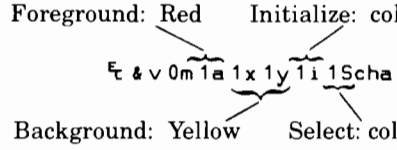
Select an Alpha Color Pair `Et&v<color pair#>S`

Example: Creating a Chameleon

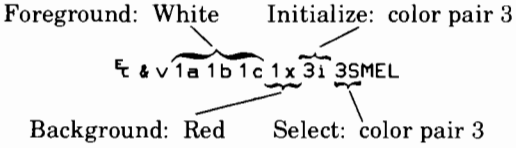
Using the RGB color method (the power-on value), you can create a chameleon in four steps:

(1) print "cha" as red letters on a yellow background.
(2) print "MEL" as white letters on a red background.
(3) print "eon" as cyan letters on a magenta background.
(4) change color pair assignment of "MEL" to obscure the chameleon's middle segment.

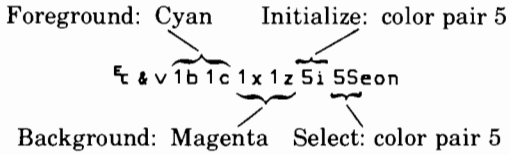
Step 1. The display is normally set to color pair "0". Assign the red on yellow to another color pair, such as color pair "1". Next, explicitly select color pair "1" then print "cha".



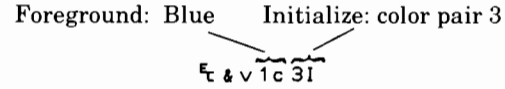
Step 2. Assign white on red to color pair "3" and print "MEL".



Step 3. Assign cyan on magenta to color pair "5" and print "eon".



Step 4. Reassign color pair "3" to be blue on black. As soon as this assignment takes effect, MEL fades away.



Background: Black (all parameters are zero)

Table 4-1 gives sample parameter values for both RGB and HSL color definitions.

Table 4-1. Sample RGB/HSL Color Definition Values

R	G	B	Color	H	S	L
0	0	0	Black	X	X	0
0	0	1	Blue	.66	1	1
0	1	0	Green	.33	1	1
0	1	1	Cyan	.5	1	1
1	0	0	Red	1	1	1
1	0	1	Magenta	.83	1	1
1	1	0	Yellow	.16	1	1
1	1	1	White	X	0	1

X = don't care (may be any value between 0 and 1)

These listed values map directly to the colors supported by the 2627A. Therefore, you should use these values when programming. Since the escape sequences allow for many possible color definitions, the 2627A interprets other values as shown by the following tables.

Table 4-2. HSL Definition Algorithm

COLOR SELECTED	parm. 1 H RANGE	parm. 2 S RANGE	parm. 3 L RANGE
BLACK	don't care	don't care	< 0.25
WHITE	don't care	don't care	>= 0.25
RED	.00- .08	>= 0.25	>=0.25
YELLOW	.09- .24	>= 0.25	>= 0.25
GREEN	.25- .41	>= 0.25	>= 0.25
CYAN	.42- .58	>= 0.25	>= 0.25
BLUE	.59- .74	>= 0.25	>= 0.25
MAGENTA	.75- .91	>= 0.25	>= 0.25
RED	.92-1.00	>= 0.25	>= 0.25

In the RGB color method, when N represents the largest-valued (most intense) color value of the three color specifications, colors are selected as follows:

Table 4-3. RGB Definition Algorithm

COLOR SELECTED	parm.1 RED RANGE	parm. 2 GREEN RANGE	parm. 3 BLUE RANGE
BLACK	<.25 or <N/2	<.25 or <N/2	<.25 or <N/2
WHITE	>=.25 and >=N/2	>=.25 and >=N/2	>=.25 and >=N/2
YELLOW	>=.25 and >=N/2	>=.25 and >=N/2	<.25 or <N/2
GREEN	<.25 or <N/2	>=.25 and >=N/2	< .25 or <N/2
CYAN	<.25 or <N/2	>=.25 and >=N/2	>= .25 and >=N/2
BLUE	<.25 or <N/2	<.25 or <N/2	>=.25 and >=N/2
MAGENTA	>=.25 and >=N/2	<.25 or <N/2	>=.25 and >=N/2
RED	>=.25 and >=N/2	<.25 or <N/2	<.25 or <N/2

DESIGNING AND USING FORMS

With the terminal, you can design elaborate data entry forms constructed of varying line types from the line drawing set and containing alphanumeric annotations and protected and unprotected fields.

When format mode is enabled, the cursor automatically moves to the start of the first unprotected field in the form. Henceforth, the terminal operator can only enter data into those portions of the display screen which lie within unprotected fields; the remainder of the screen is protected. When the operator enters a character into the last position of a field, the cursor advances to the start of the next unprotected field. In addition, the SHIFT TAB and TAB keys can be used to move the cursor to the start of the preceding or next unprotected field, respectively. If the cursor is within a protected field, it automatically advances to the start of the next unprotected field when the operator attempts to type a data character.

You enable and disable format mode programmatically by using the following escape sequences:

ENABLE: ESC W
DISABLE: ESC X

These sequences may be entered through the keyboard, executed from within a user key definition, or issued from a program in a host computer.

There are three major steps to creating data entry forms:

1. Create the linear structure of the form on the screen using the line drawing set.
2. Define the various unprotected fields within the form.
3. Programmatically read the completed form and store it in the host computer for future use.

DRAWING FORMS

The terminal is equipped with two character sets: the standard USASCII/Roman Extension character set and the Line Drawing character set. The extended USASCII set is defined as the Base set, and the Line Drawing set is defined as the alternate character set. The elements of the Line Drawing set and their relationship to the terminal's standard USASCII keyboard are illustrated in figure 4-8.

Note that the relationship of some line drawing set elements to the keys varies if the terminal is configured to another national character set. (Refer to the Terminal Configuration Menu discussed in Section II for the terminal's present configuration.) Appendix C provides a description of the national keyboard options and a code chart which cross-references the keyboard character to the line drawing set character.

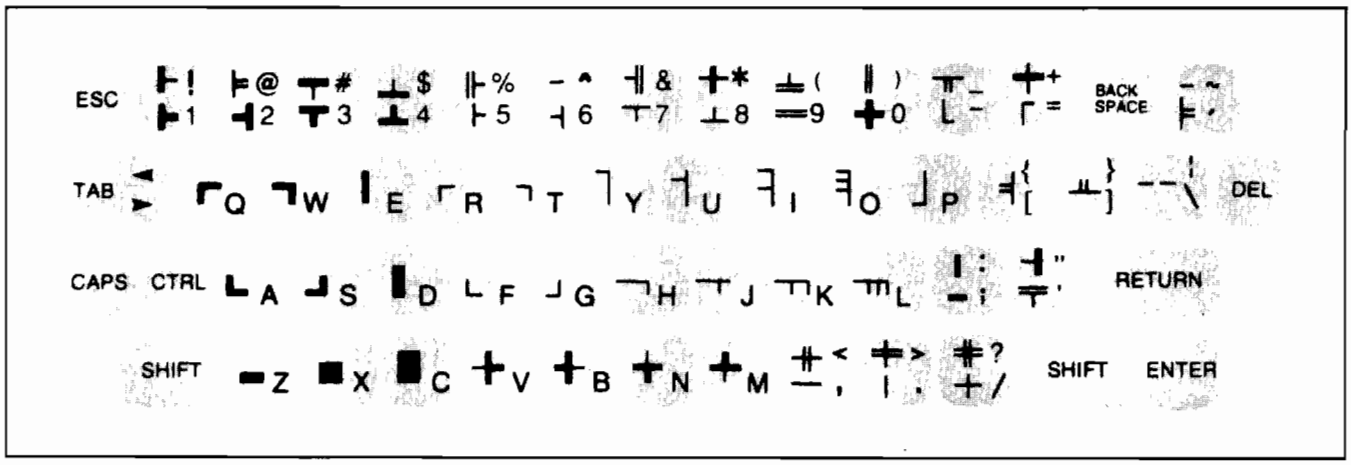


Figure 4-8. Line Drawing Set Elements

The first step in generating a data entry form is to create the linear structure of the form on the screen along with any constant alphanumeric annotations such as the form's title and the row/column headings. You do this using the Line Drawing and Base character sets.

When you are designing a form through the keyboard, you switch from text characters to the Line Drawing set by issuing an ASCII $^{5}_{6}$ code (control-N) and you switch from the Line Drawing set back to the Base set by issuing an ASCII$^{5}_{1}$ code (control-O). Note that the $^{5}_{6}$ and $^{5}_{1}$ code affect only those characters from the current cursor position through the next $^{5}_{6}$ or $^{5}_{1}$ code or through the end of the line or the start of a video enhancement field, whichever occurs first. Consequently, if the Line Drawing set is enabled at the end of one line on the screen and you also want it enabled at the start of the next line, you will have to explicitly issue another $^{5}_{6}$ code at the start of the second line.

Figure 4-9 illustrates a sample form and identifies the keystrokes used for generating the various different types of line segments. Figure 4-10 shows the same form as it actually appears on the terminal's screen. Figure 4-11 shows the Base set equivalent characters for the entire form structure.

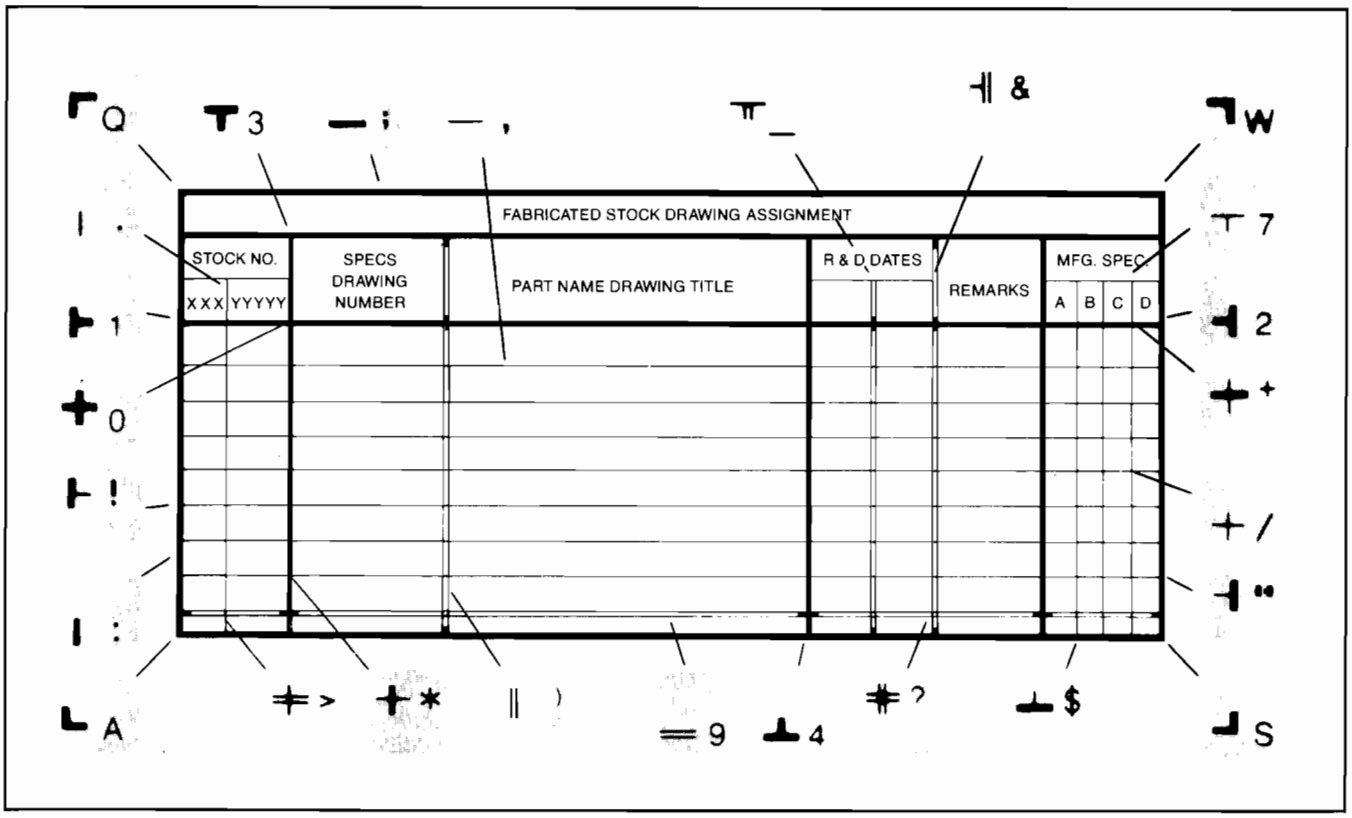


Figure 4-9. Sample Data Entry Form

FABRICATED STOCK DRAWING ASSIGNMENT

STOCK NO.		SPECS. DRAWING NUMBER	PART NAME DRAWING TITLE	R&D DATES		REMARKS	MFG. SPEC			
XXX	YYYYY						A	B	C	D

Figure 4-10. Completed Data Entry Form

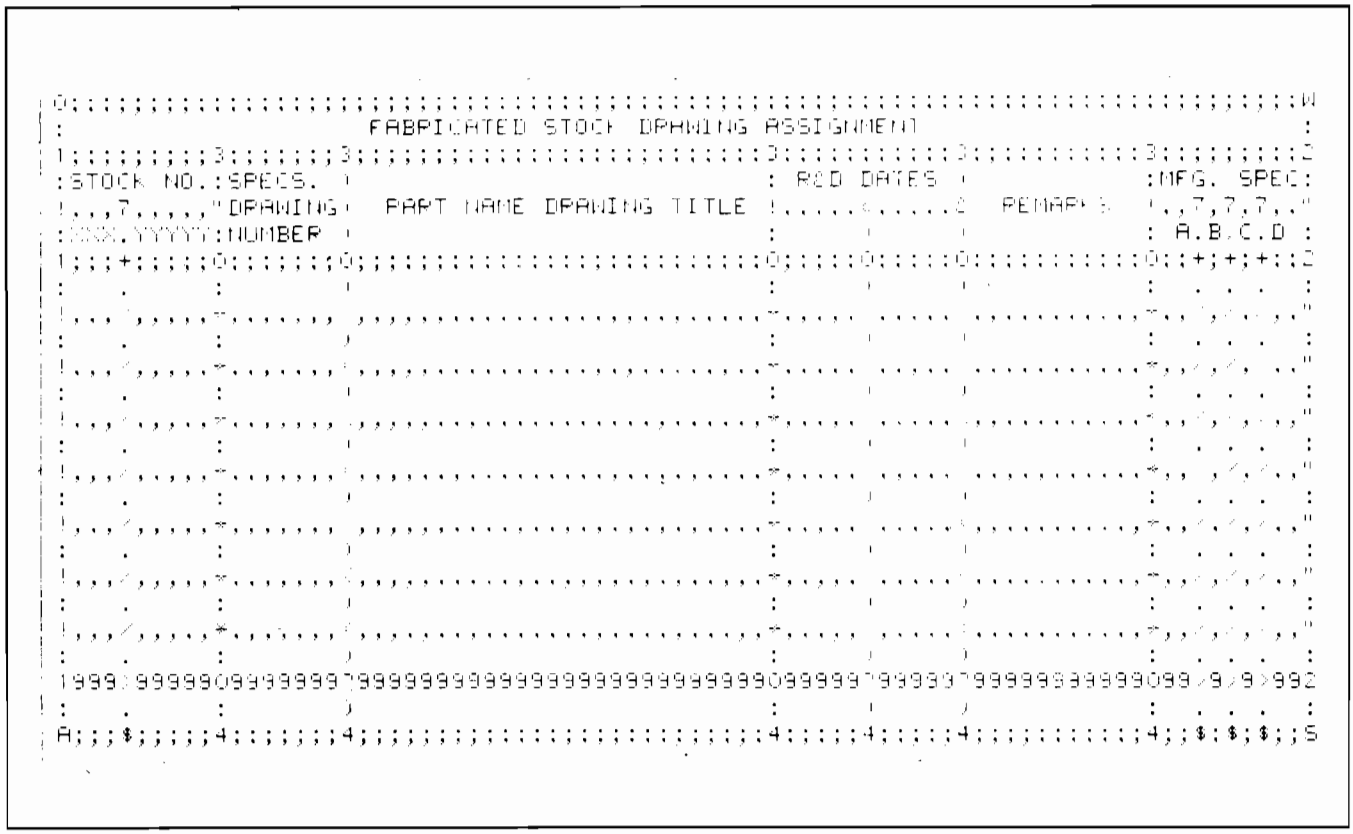


Figure 4-11. Base Set Equivalents for Data Entry Form Illustrated in Figure 4-10

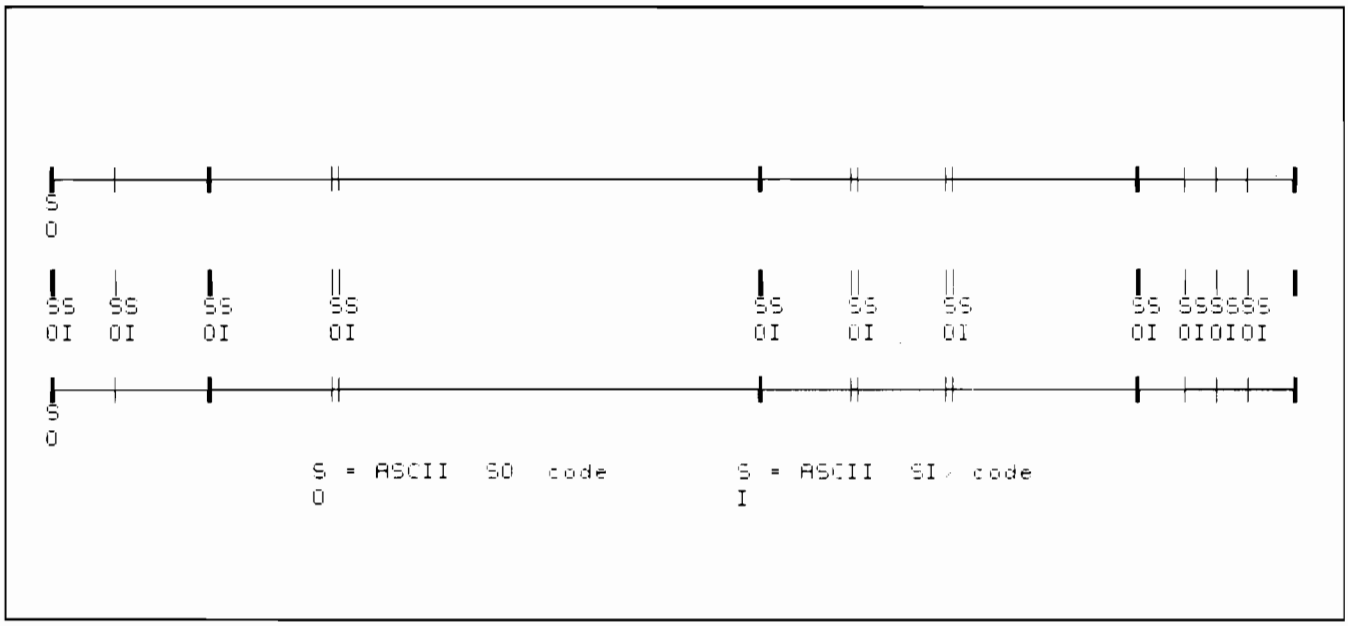


Figure 4-12. Use of Shift-In and Shift-Out Codes

One approach to generating a form structure through the keyboard is to load two of the user keys with the ^N and ^O codes (control-N and control-O, respectively, with display functions mode enabled), define both as Local keys, define their Labels as "`Line Draw`" and "`Base Set`", respectively, and enable them by pressing USER KEYS. Then draw the form structure and alphanumeric annotations using the Base set as illustrated in figure 4-11. As the form is evolving, use the cursor control keys and the two user keys to switch the linear structure portions of the form to the Line Drawing character set. When doing this, however, be sure that those portions of the form that will be used for data fields are set to the Base set (figure 4-12 shows three lines of the sample form and the various points at which you would use the ^N and ^O codes). You may also, if you wish, load some of the more repetitive line defnitions (such as the second and third lines in figure 4-12) into user keys to speed up the drawing of the main body of the form.

Video and color enhancements can be incorporated into the form using the user keys. Define the keys as "`local`", the labels as "`blinking/inverse`", "`color/pair #1`", "`under/line`", "`enhance/off`", etc. The escape sequences are given previously in this section under "DISPLAY ENHANCEMENTS" and "SELECTING COLOR PAIRS". To begin an enhancement, position the cursor on the screen; then, press the user key containing the desired enhancement. Next, to end the enhancement, position the cursor to where you want the enhancement to end (must be in the same line); then, press the key containing the escape sequence for ending the enhancement.

A program running in the host computer can also be used to draw forms on the terminal display screen.

The program must first define the line drawing character set as the alternate character set by sending an ^[)B to the terminal.

The program then must go through the same process of shifting out of the base character set (^N = control-N = decimal 14) to draw the linear portions of the form, and shift back into the base character set (^O = control-O = decimal 15) to structure the alphanumeric portions of the form.

FORMS MODE (FORMAT MODE)

In Forms Mode, the terminal prevents you from overwriting or transmitting data in protected fields. Forms Mode is normally entered under control of the computer. Forms Mode is turned on by sending ^[W (the cursor is homed to the beginning of the first unprotected field). Forms Mode is turned off (return to normal operation) with ^[X (the cursor remains in its present position).

Protected Fields

Fields can be protected so that displayed data cannot be overwritten or sent to a computer. When the terminal is placed in "Forms Mode" (Format Mode) all character positions on the screen are protected except those fields that have been specifically defined as "unprotected".

Unprotected Fields

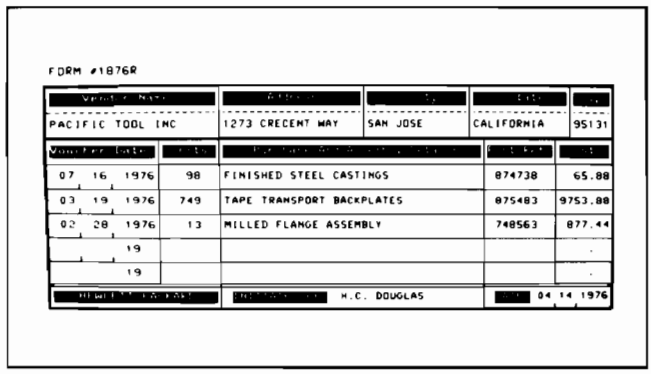
Data can be written into unprotected fields in the normal manner. After reaching the end of an unprotected field, the cursor moves to the beginning of the next unprotected field.

The tab functions can be used to move from one unprotected field to the beginning of the next unprotected field. Back tabbing causes the cursor to be positioned at the beginning of the previous unprotected field. Fields are defined as "unprotected" by using ^[[at the start of the field. ^[] or the end of the line is used to end the field.

In the following figure, only the fields shown in white are unprotected. Even if the operator moves the cursor to a protected field and types a character, the cursor will move to the nearest unprotected field before displaying the character.

NOTE

Although the terminal does not support "transmit only" fields, if the "transmit only" escape sequence (ESC {) is sent from the computer, it is redefined as an unprotected field.



Graphics Display

SECTION V

INTRODUCTION

This section contains a description of the terminal's graphics functions and how they are used. The information and the examples are intended for use in developing programs to control the graphics functions. Additional information on how to use the graphics features from the keyboard is contained in the User's Manual.

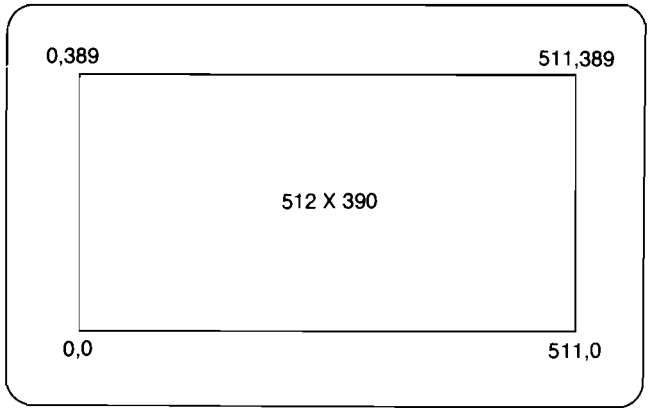
GRAPHICS DISPLAY

You can display graphics data by addressing points in a 512 by 390 array.

The graphics and alphanumeric data are displayed in the same area on the screen but are stored in separate RAM memories. This allows you to read or modify graphics and alphanumeric data separately.

NOTE

Alphanumeric characters overlay or mask out the graphics display. To allow graphics data to be seen, select black (clear) background colors for the alphanumeric pens.



KEYBOARD GRAPHICS FUNCTIONS

All of the graphics functions commands can be entered from the terminal keyboard by the operator. Some of the functions are available through a special set of graphics control keys located to the right of the normal ASCII character set (see figure 5-1). Table 5-1 contains a list of the keys and a description of their functions. These keys can be used in both local and remote operation. This allows a combination of operator and program control of graphics functions

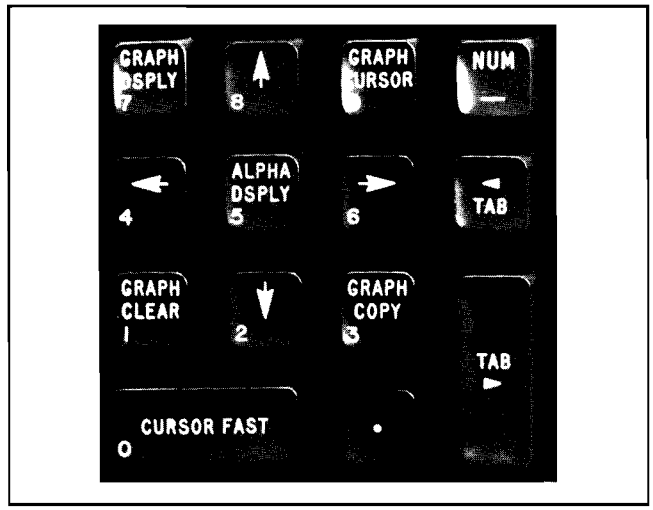


Figure 5-1. Location of Graphics Keys

to be used to make maximum use of the terminal's capabilities. Detailed information for using the graphics control keys is contained in the User's Manual.

Each key (except the two TAB keys and the decimal point key) performs two functions. The NUM key controls whether or not the "10-key" pad performs the graphics functions labeled on the keys or the numeric functions labeled in the lower-left corner of each key. When the terminal is turned on initially, or after a hard reset SHIFT CTRL RESET, the keypad is in graphics mode. This means that the graphics functions labeled on the keys are in effect. To change the numeric functions, press SHIFT NUM. To change back to graphics functions, press SHIFT NUM again. As you can see, SHIFT NUM toggles the keypad functions between graphics and numerics. The only graphics keys that repeat when held depressed are those that control the graphics cursor.

When the keypad is in graphics mode, the minus function on the NUM, and the TAB◄, TAB►, and . keys are disabled.

Table 5-1. Graphics Control Keys

KEY	DESCRIPTION
GRAPH CURSOR	Toggles the graphics cursor on and off.
▲ ◄ ► ▼	Move the graphics cursor. More than one can be pressed for diagonal motion.
CURSOR FAST	Speeds up the graphics cursor if pressed in conjunction with the cursor keys. The rate returns to normal when released.
GRAPH DSPLY	Toggles the graphics display, to inhibit the graphics image without erasing.

Table 5-1. Graphics Control Keys (Continued)

KEY	DESCRIPTION
ALPHA DSPLY	Toggles the alphanumeric displays.
GRAPH CLEAR	Erases the graphics image memory.
GRAPH COPY	Copies graphics memory to the specified "to" devices.
NUM —	Toggles the function of the keypad between graphics and numerics when SHIFT is pressed simultaneously. Unshifted in numeric mode, the key is used to display a dash (—) character.

PROGRAMMABLE GRAPHICS FUNCTIONS

Graphics functions are controlled by parameterized escape sequences. All graphics escape sequences begin with Esc *. The third character, always lower case, selects the type of graphics sequence. Table 5-2 lists the types of graphics sequences. For example, Esc * p specifies a plotting sequence.

Subsequent characters in the control sequence are read as either parameters or commands, depending on the location of the character in the ASCII table.

Table 5-2. Summary of Graphics Sequence Types

ESCAPE SEQUENCE	DESCRIPTION
Esc * d	Display Control
Esc * e	Image Control
Esc * l	Labeling
Esc * m	Drawing Mode
Esc * n	Graphics Text
Esc * p	Vector Plotting
Esc * s	Graphics Status
Esc * t	Compatibility Mode
Esc * w	Graphics Initialization

Control Codes

Control codes are generally ignored, with the exception of the ESCAPE character (Esc). If an Esc character is detected and the previous graphics control sequence has not been properly terminated with a "Z" *or some other valid upper case character*, the Esc will cause the execution of the previous sequence to be terminated. The new escape sequence will then be executed.

Commands

Graphics commands come from columns 4–7 of the ASCII table, the upper and lower case letters (A–Z and a–z). Both upper and lower case commands execute the same function. *Upper case letters terminate the sequence and cause it to be executed. You can use more than one command in a sequence.*

BIT 7 6 5 / 4 3 2 1	0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1
0000	NUL	DLE	SP	0	@	P	`	p
0001	SOH	DC1	!	1	A	Q	a	q
0010	STX	DC2	"	2	B	R	b	r
0011	ETX	DC3	#	3	C	S	c	s
0100	EOT	DC4	$	4	D	T	d	t
0101	ENQ	NAK	%	5	E	U	e	u
0110	ACK	SYN	&	6	F	V	f	v
0111	BEL	ETB	'	7	G	W	g	w
1000	BS	CAN	(	8	H	X	h	x
1001	HT	EM	)	9	I	Y	i	y
1010	LF	SUB	*	:	J	Z	j	z
1011	VT	ESC	+	;	K	[	k	{
1100	FF	FS	,	<	L	\	l	|
1101	CR	GS	-	=	M	]	m	}
1110	SO	RS	.	>	N	^	n	~
1111	SI	US	/	?	O	_	o	DEL

Parameters Commands

BIT 7 6 5 4 3 2 1	
0 0	Control Code
0 1	Parameter
1 0	Command and Terminate Sequence
1 1	Command and Continue Sequence

Graphics sequences can be any length. (The terminal ignores CR and LF characters in the middle of graphics sequences.) For example, to plot a figure containing 100 points the escape sequence could appear as follows:

```
Esc * p a <x1,y1> . . . <x100,y100>Z
```

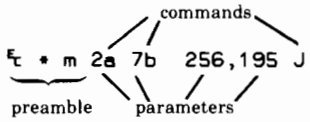
This could cause problems if an error occurs and the system tries to report it in the middle of a long sequence. Since most systems use upper case characters for messages, the first character of the message would end any graphics sequence that might be in progress. Letters that have not been assigned a function for a particular graphics sequence are treated as NOPs and if they are lower case, are ignored. If upper case, they will end the sequence. The letter z has been defined as a NOP in all sequences so that a capital Z can always be used to end a graphics escape sequence.

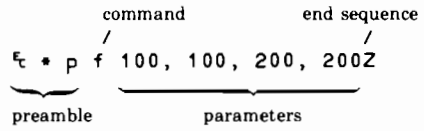
Parameters

Parameters come from columns 2 and 3 of the ASCII table (SPACE through ?). Most parameters are simply the ASCII numeric characters used to represent data coordinates or to select one of several settings. Binary formatted data is generated by appending the bits 0 1 to five bits of binary data. Note that in binary formats, spaces are treated as data and are not ignored or used as delimiters. Both ASCII and binary data formats are described later in this section.

Parameters precede their associated commands (postfix notation). The most frequently used parameters are vector data. Refer to the discussion of Vectors for additional information on parameters used to define vector operations.

Examples:





The terminal's graphics functions are organized into the following groups:

- Display Control
- Cursor Control
- Plotting Sequences
- Vector Data Formats
- Relocatable Origin
- Pens
- Line Types
- Area Fills
- Drawing Modes
- Graphics Text
- Graphics Defaults

Table 5-3 lists the escape code sequences used to control the graphics functions.

Table 5-3. Graphics Functions

Display Control

Ec*da	Clears graphics display memory
Ec*db	Sets graphics display memory
Ec*dc	Turns on graphics display
Ec*dd	Turns off graphics display
Ec*de	Turns on alphanumeric display
Ec*df	Turns off alphanumeric display
Ec*eb	Sets background pen and clears graphics display to background pen color

Cursor Control

Ec*dk	Turns on graphics cursor
Ec*dl	Turns off graphics cursor
Ec*do	Moves graphics cursor to absolute coordinates
Ec*dp	Moves graphics cursor to incremental coordinates
Ec*dq	Turns on alphanumeric cursor
Ec*dr	Turns off alphanumeric cursor

Plotting Sequences

Ec*dm	Turns on rubberband mode
Ec*dn	Turns off rubberband mode
Ec*pa	Lift pen
Ec*pb	Lower pen
Ec*pc	Use graphics cursor as next data point
Ec*pd	Draw a point at the current pen position

Vector Data Formats

Ec*pf	Use ASCII absolute format
Ec*pg	Use ASCII incremental format
Ec*ph	Use ASCII relocatable format
Ec*pi	Use binary absolute format
Ec*pj	Use binary short incremental format
Ec*pk	Use binary long incremental format
Ec*pl	Use binary relocatable format

Relocatable Origin

Ec*mj	Set relocatable origin to absolute coordinates
Ec*mk	Set relocatable origin to current pen position
Ec*ml	Set relocatable origin to graphics cursor position

Pens

Ec*mx	Select primary pen
Ec*my	Select secondary pen

Line Types

Ec*mb	Select line type
Ec*mc	Define user line pattern

Area Fills

Ec*md	Define user area fill pattern
Ec*me	Absolute rectangular area fill
Ec*mf	Relocatable rectangular area fill
Ec*mg	Select area fill pattern
Ec*mh	Select area boundary pen
Ec*mw	Select predefined dither pattern
Ec*mv	Define user dither pattern
Ec*ps	Begin polygon area fill
Ec*pt	Close polygon area fill
Ec*pu	Lift boundary pen
Ec*pv	Lower boundary pen

Drawing Modes

Ec*ma	Select drawing mode

Graphics Text

Ec*ds	Turn graphics text on
Ec*dt	Turn graphics text off
Ec*l	Graphics label
Ec*mm	Set graphics text size
Ec*mn	Set graphics text direction
Ec*mo	Turn on graphics text slant
Ec*mp	Turn off graphics text slant
Ec*mq	Set graphics justification/origin
Ec*nx	Set graphics text pen

Table 5-3. Graphics Functions (Continued)

Drawing Defaults

Esc*mr	Set graphics defaults
Esc*wr	Performs graphics hard reset

DISPLAY CONTROL

Graphics display control is made up of the functions used to control the graphics cursor or the state of the graphics memory.

Graphics Display Set/Clear

All the points in graphics display memory can be set to a selected color programmatically:

Clear Graphics Memory: Esc*d<pen#>a (Default is pen 0)
Set Graphics Memory: Esc*d<pen#>b (Default is pen 7)

where <pen#> may be in the range −32768 to 32767. The terminal uses the lowest three binary bits of the selected pen# to determine the color of the pen#(0-7). If the optional pen# is not selected, the Esc*d<pen#>a sequence defaults to (pen 0, black) and the Esc*d<pen#>b sequence defaults to (pen 7, white).

Example: Clear the screen and set the background to blue:

Esc*d4A

You may also clear graphics data on the screen from the keyboard by pressing [key] key on the graphics/numeric keypad. This will clear the screen to black.

Graphics Display On/Off

You can turn the graphics part of the screen display on or off programmatically:

Turn On Graphics Display: Esc*dc

Turn Off Graphics Display: Esc*dd

These functions enable or disable the routing of data from raster memory to the terminal display. The alphanumeric display and raster memory data are unaffected.

From the graphics/numeric keypad, pressing [key] toggles the graphics display on and off:

Alphanumeric Display On/Off

You can turn the alphanumeric screen display on or off.

Turn On Alphanumeric Display: Esc*de

Turn Off Alphanumeric Display: Esc*df

These functions enable or disable the routing of data from alphanumeric memory to the terminal display. The graphics display and alphanumeric data are unaffected.

Image Control

Set background pen number: Esc*e<pen#>b

where <pen#> may be in the range −32768 to 32767 (default is 0, black).

The terminal uses the lowest three binary bits of the selected pen# to determine the background color. Executing the escape sequence causes the display to be cleared (set) to the specified color. The background pen is set to black at power on, after a hard reset, or if a Set Graphics Default command is issued.

CURSOR CONTROL

A separate graphics cursor is available for use in locating points in the graphics display. The graphics cursor is used by the terminal operator to input position data or to interact with a graphics application program.

Graphics Cursor On/Off

The graphics cursor is initially off (power on or full reset). Turning the cursor on or off does not effect the data in graphics memory.

The graphics cursor may be toggled on and off by pressing the [key] key on the graphics/numeric pad.

Programmatically, you can toggle the cursor:

Graphics Cursor On: Esc*dk

Graphics Cursor Off: Esc*dl

Graphics Cursor Positioning

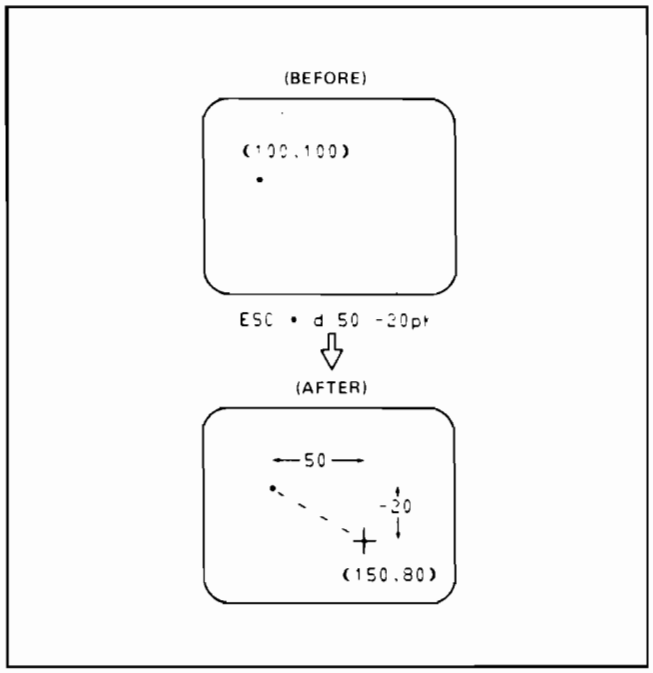
The graphics cursor is initially at position (0,0) after power on or a full reset. The cursor can be positioned (even if it is not turned on) using either absolute or relative coordinates. In the following sequences X and Y give the new cursor position. Refer to Vector Data Formats for a discussion of absolute and relative coordinates.

Position Graphics Cursor Absolute: Esc*d<X, Y>o

Position Graphics Cursor Relative: Esc*d<X, Y>p

You may position the graphics cursor from the graphics/numeric keypad by pressing [◄], [►], [▲], or [▼]. Pressing two keys simultaneously will cause diagonal movement. The [key] key may be pressed simultaneously to speed up cursor positioning.

Example: The cursor is currently at position 100,100 and off. Move it 50 units to the right and 20 units down from its current position and turn it on.



Alphanumeric Cursor On/Off

You can turn the alphanumeric cursor on and off.

Turn Alphanumeric Cursor On: Esc*dq

Turn Alphanumeric Cursor Off: Esc*dr

PLOTTING SEQUENCES

Graphic data is made up of vectors (line segments). There is no explicit "draw vector" command. Instead, the terminal uses the concept of a "pen" in drawing vector data.

The general format for a plotting sequence is:

Esc*p <pen state> <y1> <x2> <y2> . . . Z (or any capital letter)

where <pen state> indicates whether the pen is up or down, and the values are delimited by spaces or commas. The capital "Z" (a non-operative) terminates the sequence.

When enough parameters have been received to specify a data point, the pen is moved from its current position to the new end point. If the pen is down, a vector will be drawn. If the pen is up, the pen is moved to the new point (without drawing a vector) and lowered. In either case, the new point becomes the *current pen position*.

Plotting sequences can extend indefinitely. In general, longer sequences are preferred as they minimize the overhead needed for a plot sequence. As the sequence length decreases, the percentage of prefix characters increases, and the drawing rate goes down. The worst possible case would be to send each vector in its own sequence; approximately 50% of the characters sent would be overhead, reducing vector speed by a factor of 2.

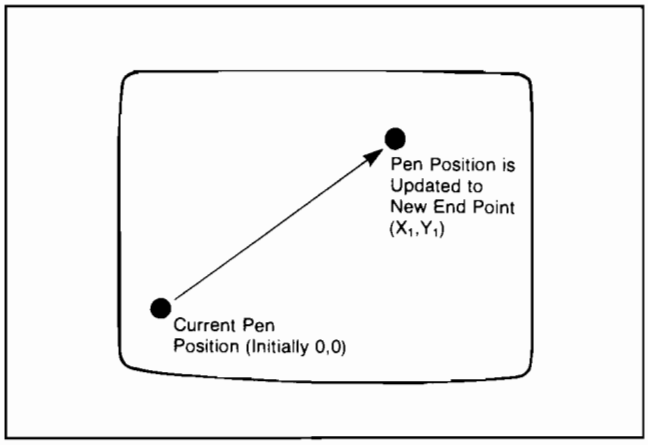


Figure 5-2 Current Pen Position and New End Point

When a graphics hard reset is performed (Esc*wr), the pen is in the down state and positioned at coordinates 0,0.

Lift Pen

Esc*pa

This command causes the imaginary plotting pen to be lifted from the drawing surface. Movement of the pen from the current position will not draw a line. The pen must be lowered (by supplying a coordinate or a lower pen command—Esc*pb) before a line can be drawn.

Example: Lift the pen, move it to 100,100, draw a vector to 200,200 and then to 50,300.

Esc*pa 100,100 200,200 50,300Z

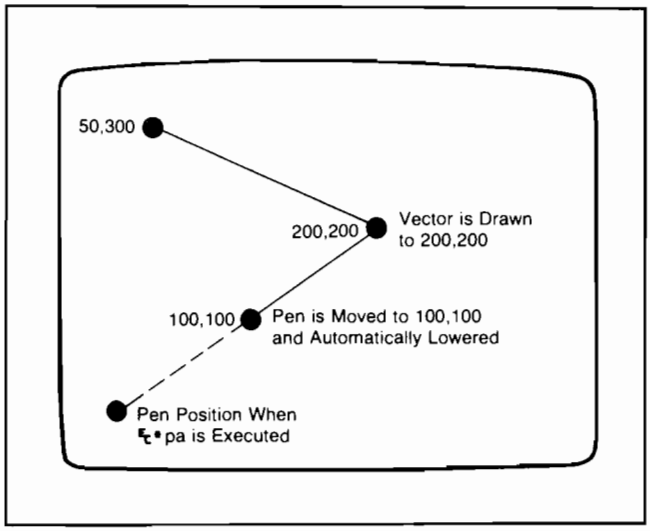


Figure 5-3. Esc*pa Example

NOTE

This command can also be used in conjunction with polygonal area fill commands. See "Polygonal Area Fills" in this section.

Lower Pen

$^{E}_{t}$*pb

This command causes the imaginary plotting pen to be lowered to the drawing surface. Movement of the pen from the current position will draw a line.

Example: Draw a vector from the current pen position to 194,250.

$^{E}_{t}$*pb 194 250Z

Use Cursor As Next Data Point

$^{E}_{t}$*pc

This command causes the position of the graphics cursor to be used as the next data point.

Rubberband Line Mode

Turn Rubber Band Line On: $^{E}_{t}$*dm

Turn Rubber Band Line Off: $^{E}_{t}$*dn

"Rubberband Line" mode causes the terminal to display a temporary line connecting the graphics cursor to the current pen position. As the cursor is moved (using the cursor control keys or move cursor commands), the temporary line will move, stretch, or contract as required to maintain the connection. The temporary line is "set" when the cursor position is entered as a new point (by executing the $^{E}_{t}$*pc command). The origin of the temporary "rubberband" line is then updated to the new point and the process may be repeated.

NOTE

If the graphics cursor is not already on, activating the rubberband line function will turn on the graphics cursor.

Draw A Point At The Current Pen Position

$^{E}_{t}$*pd

This command draws a point at the current pen position. The pen is left in the "down" position. This command is ignored when encountered during an area fill sequence.

VECTOR DATA FORMATS

Graphic data is made up of vectors. Each vector is specified by the current graphic starting point and an end point. The current graphic starting point is one of the following:

0,0 Initial starting point

Last point defined by the graphics cursor ($^{E}_{t}$*pc).

Last point defined by data in a draw or move command ($^{E}_{t}$*pf/g/h/i/j/k/l).

Graphic points are specified in one of following formats:

- ASCII Absolute
- ASCII Incremental
- ASCII Relocatable
- Binary Absolute
- Binary Incremental
- Binary Short Incremental
- Binary Relocatable

If no format is specified in the graphic command, ASCII absolute format is assumed. More than one point can be given in a command. This minimizes communications overhead. Tables 5-4, 5-5 and 5-6 provide a reference for computer data bytes used in the various vector formats.

ASCII Formats

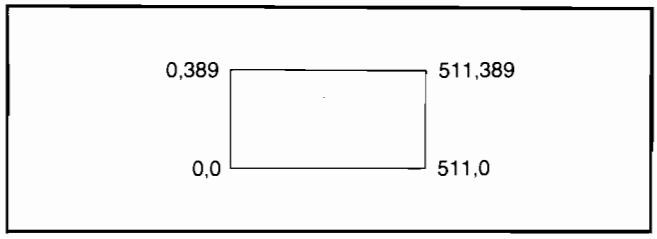
In the ASCII formats, coordinates are specified with ASCII characters 0 through 9. This means that numeric characters generated by a simple print statement can be used to specify X,Y pairs. The first value is used as the X coordinate, and the second as the Y coordinate.

Spaces or commas must be used to delimit the X and Y values. Excess delimiters are ignored. Digits following a decimal point are ignored (i.e. 123.456 is read as 123).

Exponential notation cannot be used. Consequently, the values must be in integer form. The number of bytes necessary to specify a single end point depends on the magnitude of the values.

ASCII ABSOLUTE FORMAT. The values used in the ASCII absolute format can range between −16384 and 16383. Note that only points where X is the range 0 to 511 and Y is the range 0 to 389 will be visible on the screen. The following example draws vectors around the perimeter of the screen:

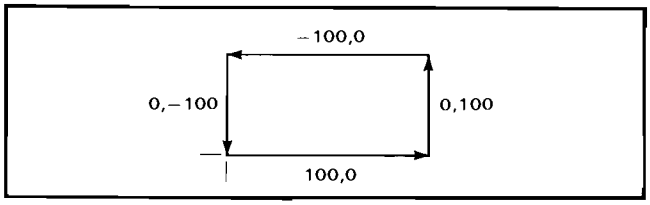
$^{E}_{t}$ * p a 0,0 511,0 511,389,0,389,0,0Z



Since no format is indicated, ASCII absolute is assumed. The "a" raises the pen, which is moved to (0,0) and lowered. Vectors are then drawn to (511,0), (511,389), (0,389), and back to (0,0). (Note that the values are delimited by spaces or commas. The upper case Z [a nop] terminates the sequence. Imbedded carriage return and line feed characters are ignored.)

ASCII INCREMENTAL FORMAT. In the ASCII incremental format you can specify a delta X and a delta Y. These values are added to the current pen position to obtain a new end point. The first value is read as delta X and the second as delta Y. For example to draw a square 100 units on a side, the following sequence could be used:

```
ESC * p g 100 0 0 100 -100 0 0 -100 Z
```

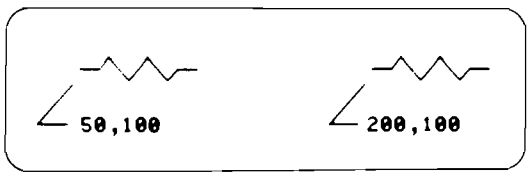


Beginning at the current pen position, a series of vectors is drawn by moving the pen 100 units to the right, up 100 units, left 100 units, and finally down 100 units. The same figure could have been drawn at any screen location by first positioning the pen to the desired starting point before sending the drawing sequence.

ASCII RELOCATABLE FORMAT. The ASCII relocatable format allows you to use a relocatable origin to be added to the incoming X and Y coordinate values. The resultant values are then treated as absolute coordinates by the terminal. The relocatable format allows you to use absolute data as if it were incremental by merely changing the relocatable origin. For example, symbol elements specified in absolute coordinates can be drawn in different locations as shown in the following example.

Example: Draw a resistor symbol stored in absolute coordinates at screen locations 50,100 and 200,100.

```
Resistor Data = 0,10
                10,10
                15,15      0,20    60,0
                25,5
                35,15
                45,5
                50,10      0,0     60,0
                60,10
ESC*a50,100J
ESC*pah0,10 10,10 15,15 25,5 35,15
                45,5 50,10 60,10Z
ESC*a200,100J
ESC*pah0,10 10,10 15,15 25,5 35,15
                45,5 50,10 60,10Z
```



Binary Format

In binary format all points are sent in a packed binary format. The coordinate values are sent using the bit patterns of the ASCII characters listed in table 5-7. The number of characters required to specify a coordinate depends on the format used. The values for X and Y coordinates can be from −16384 to 16383.

BINARY ABSOLUTE FORMAT. Binary absolute data is plotted with respect to an origin at 0,0. Four bytes are required to specify a single end point. A 10 bit coordinate in the range 0-1023, is sent for both x and y.

The bytes are ordered as follows:

BIT	7	6	5	4	3	2	1	
BYTE 1	0	1	X9	X8	X7	X6	X5	HI X
BYTE 2	0	1	X4	X3	X2	X1	X0	LOW X
BYTE 3	0	1	Y9	Y8	Y7	Y6	Y5	HI Y
BYTE 4	0	1	Y4	Y3	Y2	Y1	Y0	LOW Y

Although it is possible to send coordinates in the range 0 to 1023, only points in the range 0-511 for X, and 0-389 for Y are visible on the screen. Vectors going off the screen are clipped. If the data requires scaling, this must be done before the data is sent to the terminal.

The following example shows how the 4 data bytes are computed. The numbers are converted to the 10 bit binary equivalent. Bits 7 and 6 are set to 01 to indicate a parameter.

```
X = 0 = 00000 00000          Y = 0 00000 00000
        HI X  LOW X                  HI Y  LOW Y

        BYTE 1 = 01 00000 = SPACE HI X
        BYTE 2 = 01 00000 = SPACE LOW X
        BYTE 3 = 01 00000 = SPACE HI Y
        BYTE 4 = 01 00000 = SPACE LOW Y


X = 360 = 01011 01000       Y = 180 = 00101 10100
          HI X  LOW X                 HI Y  LOW Y

          BYTE 1 = 01 01011 = +  HI X
          BYTE 2 = 01 01000 = (  LOW X
          BYTE 3 = 01 00101 = %  HI Y
          BYTE 4 = 01 10100 = 4  LOW Y
```

An escape sequence to draw a vector from 0,0 to 360,180 is as follows:

```
ESC * p i a SP  SP  SP  SP  +  (  %  4  Z
            \X=0/   \Y=0/ \X=360/ \Y=180/
```

ESC*p selects a plotting sequence. The "i" specifies binary format. The "a" raises the pen up. The first 5 bytes (all spaces) move the raised pen to 0,0 where it is lowered. The next 4 bytes specify the point 360,180. After the 4th byte is received, the pen is moved to that point, drawing a vector. The upper case "Z" terminates the escape sequence. Note that if spaces are used in the data sequence they are interpreted as data that could result in an improper plot.

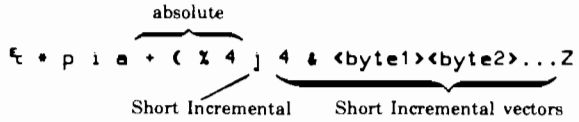
BINARY SHORT INCREMENTAL FORMAT. The short incremental format uses two bytes to specify a delta X and a delta Y in the range −16 to +15. The five least significant bits are interpreted as a signed, two's complement number. This number is added to the current pen position to obtain the new end point. The data bytes are ordered as follows:

```
BIT     7   6   5   4   3   2   1
BYTE 1  0   1   <     DELTA X     >
BYTE 2  0   1   <     DELTA Y     >
```

The following example illustrates the computation and use of the short incremental format:

```
DELTA X = -12 = 10100    DELTA Y = 6 = 00110
BYTE1 = 01 10100 = 4   DELTA X
BYTE2 = 01 00110 = &   DELTA Y
```

The following sequence moves the pen to 360,180 in absolute format, then draws a vector to X = 360−12 = 348, y = 180+6 = 186.



BINARY INCREMENTAL FORMAT. Incremental is similar to short incremental, but with a larger range. Using six bytes, delta X and Y can range from −16384 to +16383.

```
   BIT 7 6  5    4    3    2    1
BYTE1  0 1 DX14 DX13 DX12 DX11 DX10 HI  DELTA X
BYTE2  0 1 DX9  DX8  DX7  DX6  DX5  MID DELTA X
BYTE3  0 1 DX4  DX3  DX2  DX1  DX0  LOW DELTA X

BYTE4  0 1 DY14 DY13 DY12 DY11 DY10 HI  DELTA Y
BYTE5  0 1 DY9  DY8  DY7  DY6  DY5  MID DELTA Y
BYTE6  0 1 DY4  DY3  DY2  DY1  DY0  LOW DELTA Y
```

The following example shows how incremental data bytes are generated.

```
DELTA X = -400 = 11111  10011   10000
                 HI DX  MID DX  LO DX

DELTA Y = 100 = 00000  00011   00100
                HI DY  MID DY  LO DY

BYTE 1 = 01 11111 = ?      HI  DELTA X
BYTE 2 = 01 10011 = 3      MID DELTA X
BYTE 3 = 01 10000 = 0      LO  DELTA X

BYTE 4 = 01 00000 = space  HI  DELTA Y
BYTE 5 = 01 00011 = #      MID DELTA Y
BYTE 6 = 01 00100 = $      LO  DELTA Y
```

Table 5-4. Characters Used in Packed Data Formats

ASCII Character	Bit Pattern	ASCII Character	Bit Pattern
SP	01 0 0000	0	01 1 0000
!	01 0 0001	1	01 1 0001
"	01 0 0010	2	01 1 0010
#	01 0 0011	3	01 1 0011
$	01 0 0100	4	01 1 0100
%	01 0 0101	5	01 1 0101
&	01 0 0110	6	01 1 0110
'	01 0 0111	7	01 1 0111
(	01 0 1000	8	01 1 1000
)	01 0 1001	9	01 1 1001
*	01 0 1010	:	01 1 1010
+	01 0 1011	;	01 1 1011
,	01 0 1100	<	01 1 1100
-	01 0 1101	=	01 1 1101
.	01 0 1110	>	01 1 1110
/	01 0 1111	?	01 1 1111

BINARY RELOCATABLE FORMAT. Binary relocatable format specifies absolute X and Y coordinates in the range −16384 to +16383 using 6 bytes. The value specified in the relocatable origin command is taken to be the 0,0 point. The actual screen address is computed by the terminal by adding the relocatable origin to the X,Y pair.

```
   BIT  7  6   5   4   3   2   1
BYTE 1  0  1  X14 X13 X12 X11 X10  HI  X
BYTE 2  0  1  X9  X8  X7  X6  X5   MID X
BYTE 3  0  1  X4  X3  X2  X1  X0   LOW X

BYTE 4  0  1  Y14 Y13 Y12 Y11 Y10  HI  Y
BYTE 5  0  1  Y9  Y8  Y7  Y6  Y5   MID Y
BYTE 6  0  1  Y4  Y3  Y2  Y1  Y0   LOW Y
```

The following example shows how relocatable data bytes are computed.

```
RELOC X = -600 = 11111 01101 01000
                 HI X  MID X LOW X

RELOC Y =  200 = 00000 00110 01000
                 HI Y  MID Y LOW Y

BYTE 1 = 01 11111 = ?      HI  X
BYTE 2 = 01 01101 = -      MID X
BYTE 3 = 01 01000 = (      LOW X

BYTE 4 = 01 00000 = space  HI  Y
BYTE 5 = 01 00110 = &      MID Y
BYTE 6 = 01 01000 = (      LOW Y
```

Mixing Data Formats

There are no restrictions on mixing data formats. Simply specify the new format to be used, and follow it with data in the new format. Note that by restricting data values to binary values between 32 and 63, (the printing graphic characters and numbers), the plotting commands "a-z" can be intermixed in the binary data without confusion.

Example: Move the pen to 360,180 in ASCII absolute format, then draw a box 10 units wide by 5 units high using binary short incremental.

```
Et*p a f 360,180 j *SP SP %6 SP SP ; Z
         1       2  3  4  5
```

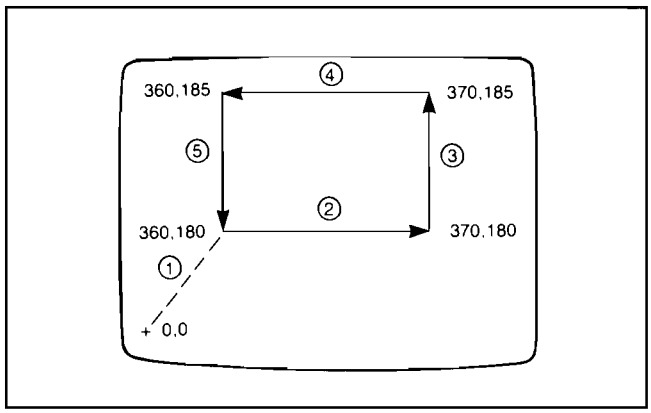


Figure 5-4. Example of Mixed Data Formats

Relocatable Origin

The relocatable origin allows you to use one set of data and drawing commands to display a figure at several different positions on the screen. (See the resistor example under ASCII Relocatable Format.)

You can also display portions of a figure that is too large to fit on the screen. You can create a "window" that can be positioned to display any 512 by 390 unit portion of the figure. The value of the relocatable origin is added to the relocatable data to obtain the coordinates used to draw the data. Figure 5-5 illustrates the effect of a Relocatable Origin on the display.

This technique eliminates the need to check boundary conditions or compute new data in order to display the desired portion of the figure. Simply set the relocatable origin to the proper value to display the desired portion of the figure and then send the unchanged figure data to the terminal. The terminal will then automatically select and adjust the "window" data.

Table 5-5. Absolute Format Addressing Bytes

	0	1	2	3	4	5	6	7	8	9
0	■■	■!	■"	■#	■$	■%	■&	■'	■(	■)
10	■*	■+	■,	■-	■.	■/	■0	■1	■2	■3
20	■4	■5	■6	■7	■8	■9	■:	■;	■<	■=
30	■>	■?	!■	!!	!"	!#	!$	!%	!&	!'
40	!(	!)	!*	!+	!,	!-	!.	!/	!0	!1
50	!2	!3	!4	!5	!6	!7	!8	!9	!:	!;
60	!<	!=	!>	!?	"■	"!	""	"#	"$	"%
70	"&	"'	"(	")	"*	"+	",	"-	".	"/
80	"0	"1	"2	"3	"4	"5	"6	"7	"8	"9
90	":	";	"<	"=	">	"?	#■	#!	#"	##
100	#$	#%	#&	#'	#(	#)	#*	#+	#,	#-
110	#.	#/	#0	#1	#2	#3	#4	#5	#6	#7
120	#8	#9	#:	#;	#<	#=	#>	#?	$■	$!
130	$"	$#	$$	$%	$&	$'	$(	$)	$*	$+
140	$,	$-	$.	$/	$0	$1	$2	$3	$4	$5
150	$6	$7	$8	$9	$:	$;	$<	$=	$>	$?
160	%■	%!	%"	%#	%$	%%	%&	%'	%(	%)
170	%*	%+	%,	%-	%.	%/	%0	%1	%2	%3
180	%4	%5	%6	%7	%8	%9	%:	%;	%<	%=
190	%>	%?	&■	&!	&"	&#	&$	&%	&&	&'
200	&(	&)	&*	&+	&,	&-	&.	&/	&0	&1
210	&2	&3	&4	&5	&6	&7	&8	&9	&:	&;
220	&<	&=	&>	&?	'■	'!	'"	'#	'$	'%
230	'&	''	'(	')	'*	'+	',	'-	'.	'/
240	'0	'1	'2	'3	'4	'5	'6	'7	'8	'9
250	':	';	'<	'=	'>	'?	(■	(!	("	(#
260	($	(%	(&	('	((	()	(*	(+	(,	(-
270	(.	(/	(0	(1	(2	(3	(4	(5	(6	(7
280	(8	(9	(:	(;	(<	(=	(>	(?	)■	)!
290	)"	)#	)$	)%	)&	)'	)(	))	)*	)+
300	),	)-	).	)/	)0	)1	)2	)3	)4	)5
310	)6	)7	)8	)9	):	);	)<	)=	)>	)?
320	*■	*!	*"	*#	*$	*%	*&	*'	*(	*)
330	**	*+	*,	*-	*.	*/	*0	*1	*2	*3
340	*4	*5	*6	*7	*8	*9	*:	*;	*<	*=

	0	1	2	3	4	5	6	7	8	9
350	+>	+?	+■	+!	+"	+#	+$	+%	+&	+'
360	+(	+)	+*	++	+,	+-	+.	+/	+0	+1
370	+2	+3	+4	+5	+6	+7	+8	+9	+:	+;
380	+<	+=	+>	+?	,■	,!	,"	,#	,$	,%
390	,&	,'	,(	,)	,*	,+	,,	,-	,.	,/
400	,0	,1	,2	,3	,4	,5	,6	,7	,8	,9
410	,:	,;	,<	,=	,>	,?	-■	-!	-"	-#
420	-$	-%	-&	-'	-(	-)	-*	-+	-,	--
430	-.	-/	-0	-1	-2	-3	-4	-5	-6	-7
440	-8	-9	-:	-;	-<	-=	->	-?	.■	.!
450	."	.#	.$	.%	.&	.'	.(	.)	.*	.+
460	.,	.-	..	./	.0	.1	.2	.3	.4	.5
470	.6	.7	.8	.9	.:	.;	.<	.=	.>	.?
480	/■	/!	/"	/#	/$	/%	/&	/'	/(	/)
490	/*	/+	/,	/-	/.	//	/0	/1	/2	/3
500	/4	/5	/6	/7	/8	/9	/:	/;	/<	/=
510	/>	/?	0■	0!	0"	0#	0$	0%	0&	0'
520	0(	0)	0*	0+	0,	0-	0.	0/	00	01
530	02	03	04	05	06	07	08	09	0:	0;
540	<■	0=	0>	0?	1■	1!	1"	1#	1$	1%
550	1&	1'	1(	1)	1*	1+	1,	1-	1.	1/
560	10	11	12	13	14	15	16	17	18	19
570	1:	1;	1<	1=	1>	1?	2■	2!	2"	2#
580	2$	2%	2&	2'	2(	2)	2*	2+	2,	2-
590	2.	2/	20	21	22	23	24	25	26	27
600	28	29	2:	2;	2<	2=	2>	2?	3■	3!
610	3"	3#	3$	3%	3&	3'	3(	3)	3*	3+
620	3,	3-	3.	3/	30	31	32	33	34	35
630	36	37	38	39	3:	3;	3<	3=	3>	3?
640	4■	4!	4"	4#	4$	4%	4&	4'	4(	4)
650	4*	4+	4,	4-	4.	4/	40	41	42	43
660	44	45	46	47	48	49	4:	4;	4<	4=
670	4>	4?	5■	5!	5"	5#	5$	5%	5&	5'
680	5(	5)	5*	5+	5,	5-	5.	5/	50	51
690	52	53	54	55	56	57	58	59	5:	5;
700	5<	5=	5>	5?	6■	6!	6"	6#	6$	6%
710	6&	6'	6(	6)	6*	6+	6,	6-	6.	6/

Note: ■ indicates a "space" character; every coordinate address must consist of the two characters shown in the table.

Table 5-6. Incremental (Short) Vector Bytes

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
-16	-15	-14	-13	-12	-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1
0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?

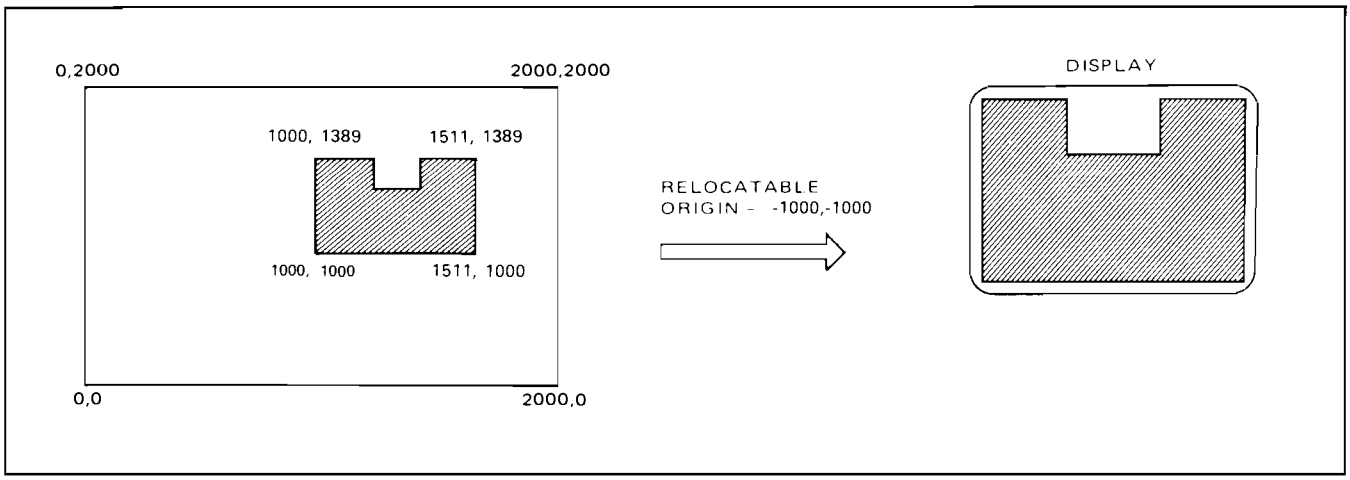


Figure 5-5. Relocatable Origin

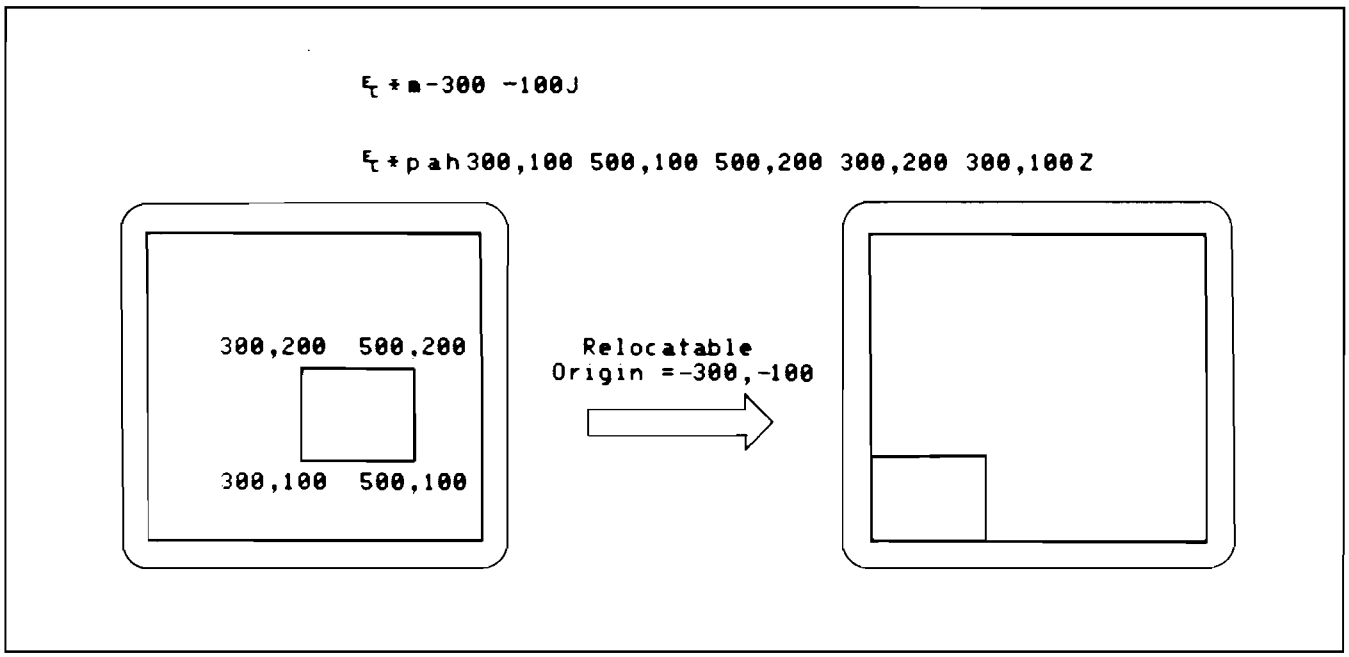
Set Relocatable Origin, Absolute

The relocatable origin can be set to any absolute coordinates using ASCII absolute format (−16384 to 16383).

Set Relocatable Origin Absolute: ESC * m <X,Y> j

where: <X,Y> are the x and y coordinates in ASCII absolute format.

Example: Set the relocatable origin to display the box in the figure so that the box is positioned at the lower left corner of the display.



Set Relocatable Origin to Current Pen Position

The relocatable origin can be set to the current pen position:

```
Et*mk
```

Set Relocatable Origin to Graphics Cursor Position

The relocatable origin can be set to the current graphics cursor position:

```
Et*ml
```

PENS

Each vector is drawn with one of eight pens (0–7). The primary and secondary drawing pen numbers are assigned using the escape sequences:

Set primary drawing pen number: `Et*m<pen#>x`

Set secondary drawing pen number: `Et*m<pen#>y`

where `<pen#>` may be in the range −32768 to 32767. The terminal uses the lowest three binary bits of the selected pen# to determine the pen#(0–7).

These assignments remain in effect until they are explicitly changed or the drawing defaults are reassigned.

The primary pen is used to draw most lines and area fills. The secondary pen is used in conjunction with specially defined line types and area fills (depending upon the drawing mode in use).

LINE TYPES

You can select the pattern to be used when drawing vectors. Patterns can be selected from a predefined set, or you can define your own pattern. This feature may be used for distinguishing between different groups of plotted data or for use in such applications as business trends, graphs, process diagrams, engineering drawings, or fabric patterns.

Selecting A Line Type

The terminal allows you to choose a line pattern for your vectors. You select a line type using the command:

```
Et*m <type> b
```

where `<type>` is shown in figure 5-6:

1 Solid line (default)	1 = ———
2 User defined line pattern	
3 Current area fill pattern	
4 Predefined pattern	4 = — · — · — · —
5 Predefined pattern	5 = — — — — —
6 Predefined pattern	6 = - - - - - -
7 Predefined pattern	7 = ...
8 Predefined pattern	8 = —·—·—·—·—
9 Predefined pattern	9 = ··· ··· ···
10 Predefined pattern	10 = —··—··—··—
11 Predefined pattern	11 = · (POINT PLOT)

Figure 5-6. Predefined Line Type Patterns

Once a line type has been selected, all subsequent vectors are drawn using that line type. You can select a vector line type to be a solid line, a user defined line pattern, the current area fill pattern, one of seven predefined dot patterns, or a point plot. Point plot causes a single point to be plotted at the vector end points. This line type is useful for generating "scattergram" type graphs. The current area fill pattern line type only applies to horizontal and vertical vectors. See "Using Area Fill Patterns as Line Types" in this section.

Example: Select line type 9, and draw a figure using the new line type.

```
Et*m 9B
Et*p a 0,0 200,0 200,100 0,100 0,0Z
```

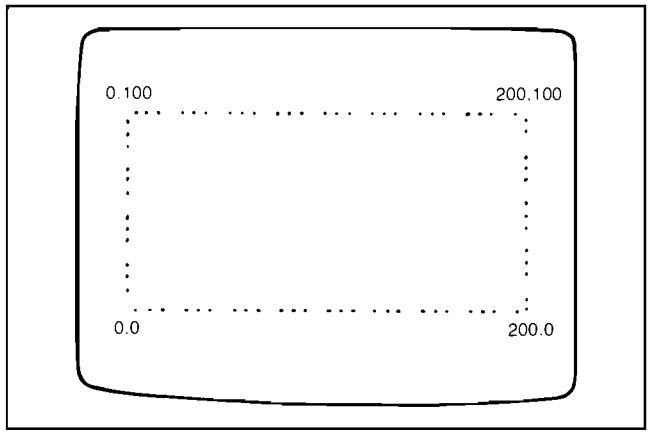


Figure 5-7. Plotting With Line Type 9

User Defined Line Types

The dot pattern used to draw vectors can be defined programmatically. Once a pattern is defined you must select "user defined line" as the line type (type=2) with the Select Line Type command.

A user defined line pattern is made up of a dot pattern and a scale factor. The dot pattern is a sequence of eight 1's and 0's. Using the default drawing mode, points indicated as a "1" in the patterns are drawn using the primary pen, and points indicated as a "0" are left unchanged. (See "Drawing Modes" in this section.)

The pattern is given as a decimal number between 0 and 255 that is the decimal equivalent of the 8-bit binary pattern. The default pattern is all "1"'s (255). For example, . . . = 10101010_B = 170. The actual number used for the pattern can be between −32768 and 32767. The least significant 8 bits of the number's 2's complement equivalent are used to determine the pattern.

The scale factor indicates how many times each bit in the pattern should be repeated. The default scale factor is 1. For example, a scale factor of 3 applied to the pattern defined above would result in a pattern of . . . . . or $1110001110001110001110000_B$.

The command for creating a user defined line type is:

Esc*m<pattern><scale>c

where <pattern> is an integer in the range (− 32768 to 32767) and <scale> is an integer in the range (1 to 255)

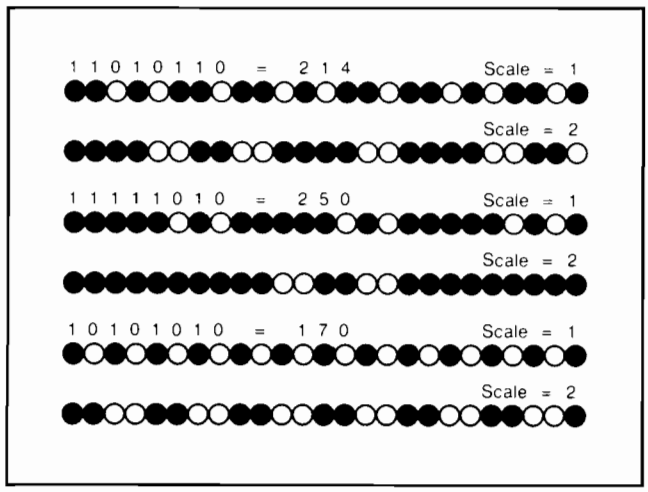


Figure 5-8 Examples of User Defined Line Patterns

Example. Define a pattern to generate the following vector:

11111111110011001111111111001100

pattern = 11111010 = 250
scale = 2

Esc*m 250 2 C

AREA FILLS

The terminal has two types of area fill specifications, rectangular and polygonal. An area can be filled with one of a variety of predefined patterns or with a user defined pattern. The pattern can also be used to provide line patterns for horizontal or vertical lines when the area pattern is selected as the line type. (Refer to "Using Area Fills As Line Types".)

When an area fill pattern is selected, the entire screen is divided up into 8x8 cells. Every location is mapped to the corresponding bit in the pattern. When an area fill operation is performed, the area fill pattern is replicated to fill the area. The pattern starts at screen coordinates 0,0 (see figure 5-9).

Selecting An Area Fill Pattern

The terminal uses two methods in filling an area: (1) area fill patterns and (2) dithering. Area fill patterns are drawn using the eight basic colors, according to the values of the primary, secondary, and background pens. Dithering is specified in densities for each of the three color planes (red, green, and blue) without regard to the primary or secondary pens. Area fill patterns and dithering are mutually exclusive. That is, when one is used to fill an area, the other method does not apply.

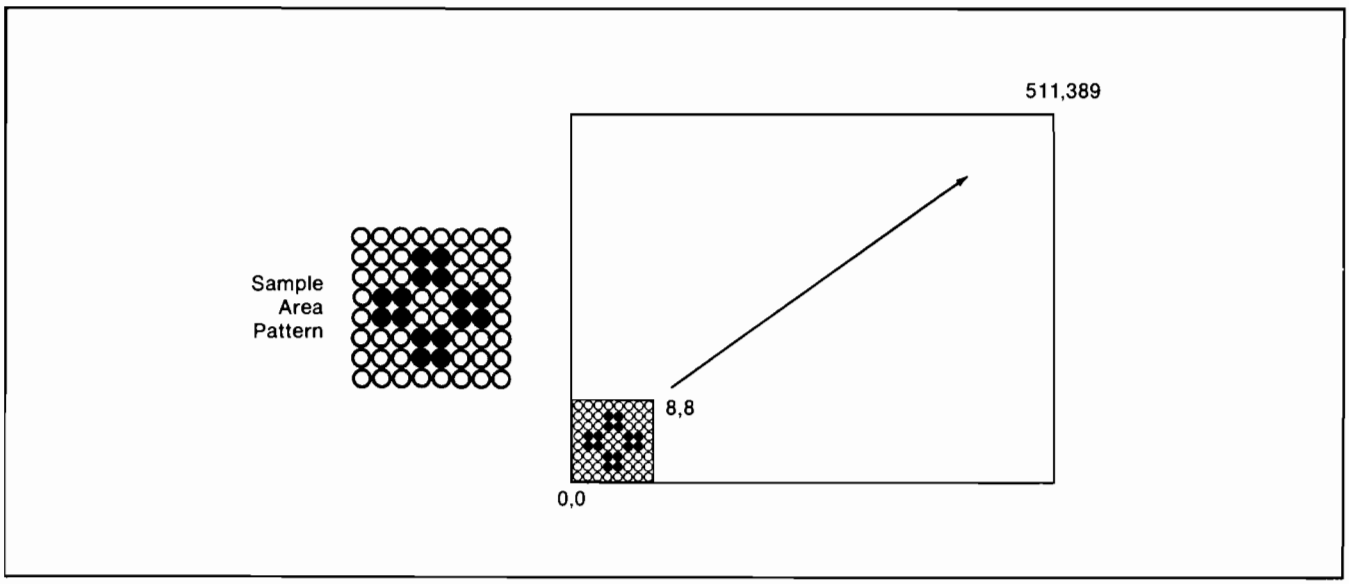


Figure 5-9. How the Area Fill Pattern is Mapped

The default pattern is a user defined area fill pattern. To select the type of area fill pattern to be used, use the following escape sequence:

```
Ec * m <area pattern> g
```

where <area pattern> is selected from:

0 Current dither pattern
1 Solid area fill
2 User defined area fill pattern (default)
3 Predefined area fill pattern #0 (short dashed hatching)
4 Predefined area fill pattern #1 (long dashed hatching)
5 Predefined area fill pattern #2 (hatching)
6 Predefined area fill pattern #3 (cross hatching)
7 Predefined area fill pattern #4 (fine cross hatching)
8 Predefined area fill pattern #5 (medium checkerboard)
9 Predefined area fill pattern #6 (fine checkerboard, 1:1 blend)
10 Predefined area fill pattern #7 (3:1 blend)

User Defined Area Fill Patterns

The user area pattern is defined using 8 parameters, one for every row of dots in the pattern. Each parameter is an integer in the range −32768 to 32767. The number is interpreted as a 2's complement number, and the least significant 8-bits are used to obtain a value between 0 and 255. The 8-bit number (0 to 255) represents an 8-bit binary pattern. Bits set to '1' are drawn using the primary pen, and bits set to '0' are not drawn (depending upon the drawing mode in use).

The command which defines a user area fill pattern is:

```
Ec*m <row 0> <row 1> . . . <row 7> d
```

where <row 0> is the 8-bit pattern for row 0

.
.
.

<row 7> is the 8-bit pattern for row 7

Example: Define a simple checkerboard pattern.

Row 0 = 10101010 = 170
Row 1 = 01010101 = 85
Row 2 = 10101010 = 170
Row 3 = 01010101 = 85
Row 4 = 10101010 = 170
Row 5 = 01010101 = 85
Row 6 = 10101010 = 170
Row 7 = 01010101 = 85

```
Ec*m 170 85 170 85 170 85 170 85 D
  row 0                          row 7
```

NOTE

The scale factor of an area fill pattern is always '1', unlike other Hewlett Packard terminals.

Other examples of user defined area patterns are shown in figure 5-10.

Using Area Fill Patterns as Line Types

If you select type 3 for the Line Type command, the line pattern is created from the current area fill pattern. Horizontal and vertical lines are drawn using the appropriate row or column from the area fill pattern. Diagonal lines are always drawn using a solid vector; they do not follow the area fill pattern. If a line is longer than 8 dots, the pattern is used over and over to complete the vector.

Example. Plot three vectors (2,3)-(7,3), (9,3)-(9,12), and (7,5)-(2,10) using a user defined area fill pattern of 51, 204, 51, 204, 51, 204, 51, 204 (figure 5-11).

Step 1. Create a user defined area fill pattern.

```
Ec*m51,204,51,204,51,204,51,204D
```

Step 2. Select the user defined area fill pattern as the current area fill pattern.

```
Ec*m 2G
```

Step 3. Select the current area fill pattern as the line type.

```
Ec*m3B
```

Step 4. Draw the vectors.

```
Ec*pa 2,3 7,3a 9,3 9, 12a 7,5 2,10 Z
```

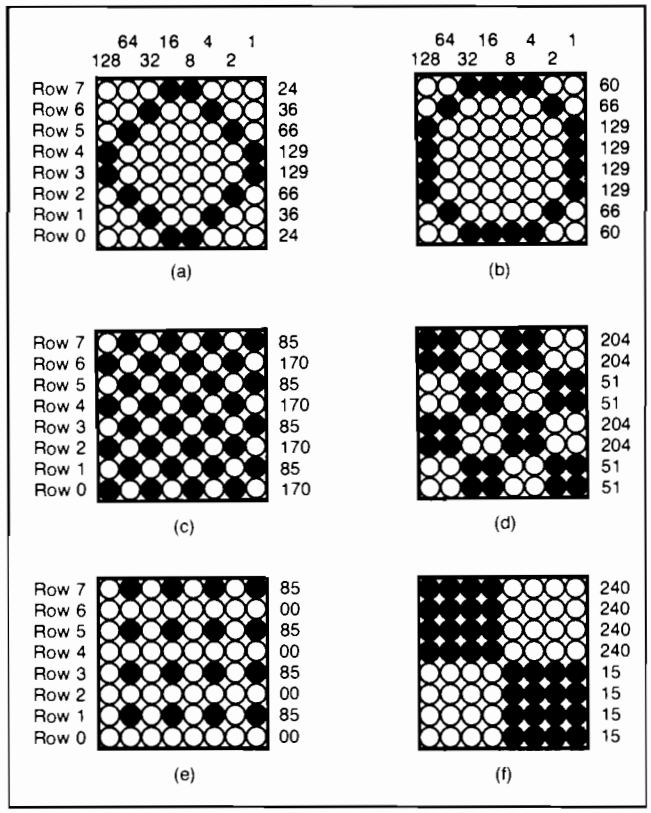


Figure 5-10. Examples of User Defined Area Fill Patterns

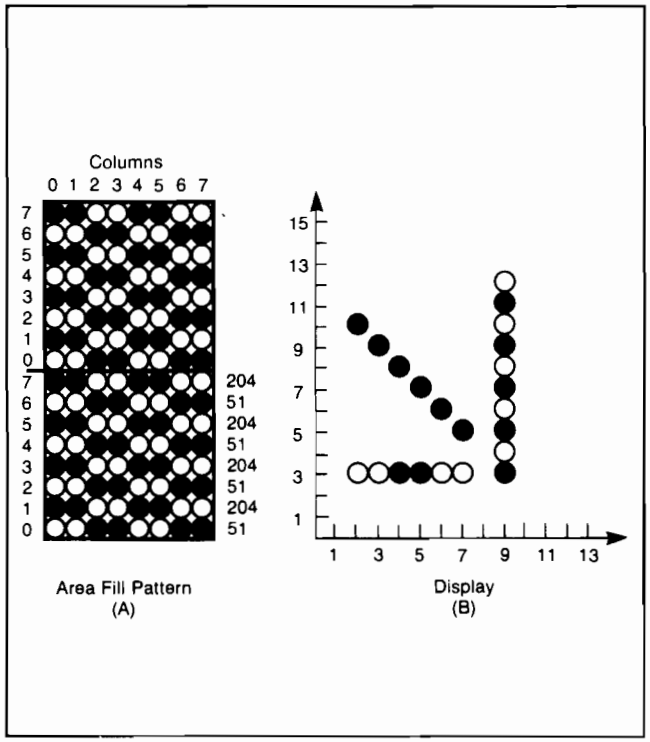


Figure 5-11. Using Area Fill Patterns As Line Types

Selecting Dither Patterns

The terminal has 11 predefined dithered color patterns that may be used to fill an area. The colors of the predefined dither patterns closely match Hewlett Packard's plotter pen colors. Select either the user defined dither pattern (default) or one of the predefined dither patterns to be the current dither pattern. The following escape sequence is used to select a dither pattern:

$^{E}t * m \langle dither\ pattern \rangle w$

where <dither pattern> is selected from:

1 User defined dither pattern (default)
2 Predefined dither pattern # 1 (violet)
3 Predefined dither pattern # 2 (brown)
4 Predefined dither pattern # 3 (burnt orange)
5 Predefined dither pattern # 4 (gold)
6 Predefined dither pattern # 5 (lime green)
7 Predefined dither pattern # 6 (turquoise)
8 Predefined dither pattern # 7 (red)
9 Predefined dither pattern # 8 (green)
10 Predefined dither pattern # 9 (blue)
11 Predefined dither pattern #10 (white)
12 Predefined dither pattern #11 (black)

NOTE

When doing area fills with predefined or user defined dither colors, there is some loss in the effective screen resolution. Therefore, single width vectors and graphics text superimposed on a dithered background may require extra width or size to be adequately seen.

User Defined Dither Patterns

The current dither pattern may be user defined using the following sequence:

$^{E}t * m \langle parm1 \rangle , \langle parm2 \rangle , \langle parm3 \rangle v$

where <parm1,2, or 3> (parameters) are selected as decimal fractions in the range 0.00 to 1.00, inclusively, to represent the density value for each color plane.

Example: Set up to fill an area with dithered orange:

Step 1. Use current dither pattern.

$^{E}t * m 0 G$

Step 2. Select user dither pattern.

$^{E}t * m 1 W$

Step 3. Define dither pattern

$^{E}t * m1 , .5 , 0V$

NOTE

The steps in this example can be combined into one escape sequence:

$^{E}t * m 0g 1w 1 , .5 , 0V$

Rectangular Area Fills

A rectangular area can be filled in with a pattern simply by sending the lower left and upper right coordinates in an escape sequence. The coordinates can be either in absolute or relocatable ASCII format. The pattern used for the area fill is determined by the Select Area Pattern command.

NOTE

The terminal can also fill irregular polygons. See "Polygonal Area Fills" in this section.

FILL RECTANGLE, ABSOLUTE

$^{E}t * m \langle x1 \rangle \langle y1 \rangle \langle x2 \rangle \langle y2 \rangle e$

where <x1> <y1> are the absolute coordinates of the lower left corner of the rectangular area to be filled (−16384 to 16383), and <x2> <y2> are the absolute coordinates of the upper right corner of the rectangular area to be filled (−16384 to 16383).

Example: Using area fill pattern 5, fill a rectangle defined by the diagonal 50,50 300,300.

$^{E}t * m 5g 50 , 50\ 300 , 300E$

FILL RECTANGLE, RELOCATABLE

$^{E}t * m \langle x1 \rangle \langle y1 \rangle \langle x2 \rangle \langle y2 \rangle f$

where <x1> <y1> are the relocatable coordinates of the lower left corner of the rectangular area to be filled (−32767 to 32767), and <x2> <y2> are the relocatable coordinates of the upper right corner of the rectangular area to be filled (−32767 to 32767).

This command is used in conjunction with the relocatable origin.

Example: Load a function key with the command:

```
ESC*ml 20,20 30,30 F
```

Use the cursor controls keys to move the graphics cursor while periodically pressing this function key.

Polygonal Area Fills

Begin Polygon Area Fill: ESC*ps
Close Polygon/Begin New Polygon: ESC*pa
Close Polygon Area Fill: ESC*pt

The terminal allows you to define any polygon up to 148 edges, and fill it with the current area fill pattern. The Begin Polygon Area Fill command causes subsequent coordinate pairs to be read as vertices of the polygon. When a lift pen command (ESC*pa) occurs in the middle of a polygon area fill sequence, a new polygon is started. (See example.) The Close Area Fill command (or any capital letter) causes the polygon to be filled using the current drawing mode, area pattern, area boundary color, and pen. Note that it is not necessary to specify a vector from the last point back to the first point; the polygon automatically closes itself at the end of the sequence.

If the polygon definition crosses over itself, the areas will be filled in an alternate order.

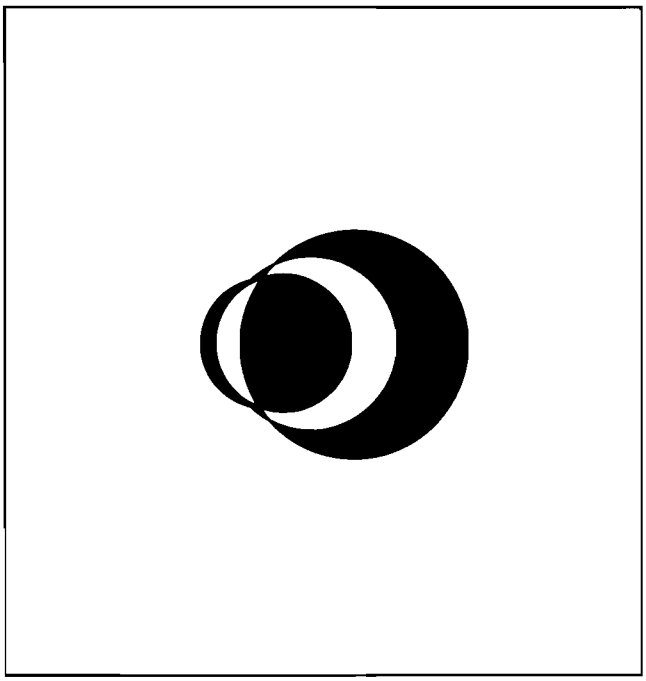


Figure 5-12. Overlapping Polygon Area Fills

Example: Move the pen to 33 0, and define and fill a pentagon 100 units on a side. Lift and move the pen to 40 10, and define another pentagon inside the first pentagon.

```
ESC*p a s 33,0 133,0 166,95 83,150 0,95
a 40,10 12,91 83,138 153,91 125,10T
```

NOTE

When using user-defined softkeys to define polygonal area fills, the entire polygon specification must be contained within one softkey, or some data may be lost.

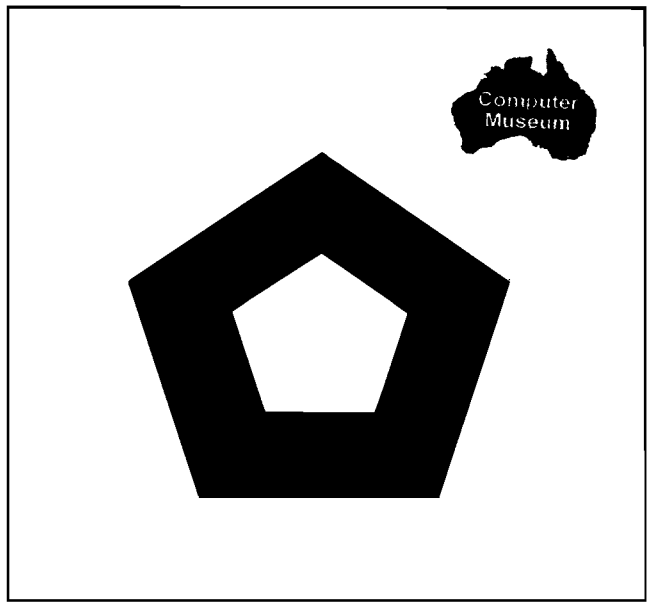


Figure 5-13. Polygon Area Fill Example

Area Boundary Pen

SELECT AREA BOUNDARY PEN. You can select the pen to be used to draw the boundary of an area fill. When a polygon area fill is performed, the vectors that are used to define the area are drawn using this pen. If the boundary pen is not defined, the edges of the boundary are the same color and pattern as the interior.

Set Area Boundary Pen: ESC*m<pen number>h

where <pen number> is an integer in the range (− 32768, . . ., 32767); the low three bits are used to select a pen in the range (0, . . ., 7).

Disable Boundary Pen: ESC*mh

LIFT/LOWER BOUNDARY PEN

Lift Boundary Pen: ESC*pu

Lower Boundary Pen: ESC*pv

If a boundary pen has been defined by the Ec*mh command (above), the pen may be "lifted" so that the edges of the polygon will be the same color and pattern as the interior. If the pen is "lifted", the boundaries of the fill will be drawn as a solid vector using the <pen number> supplied in the Ec*mh command.

NOTE

If the Ec*mh command was executed without supplying a <pen number>, the Ec*pv command has no effect.

DRAWING MODES

The terminal has seven drawing modes which affect the way in which raster pixels are set when a vector or area fill pattern is drawn. Using the default mode (JAM1), the following rules apply:

- If a bit in the pattern = 1, the corresponding raster pixel is drawn in the primary pen.
- If a bit in the pattern = 0, the corresponding raster pixel is not affected.

Drawing modes are set using the command:

Ec*m<mode>a

where <mode>	pattern value	effect
0 NOP	0,1	NOP
1 CLEAR1	0	NOP
	1	pixel ← bpen
2 JAM1	0	NOP
	1	pixel ← cpen
3 COMP1	0	NOP
	1	pixel ← NOT pixel
4 JAM2	0	pixel ← spen
	1	pixel ← cpen
5 OR	0	NOP
	1	pixel ← pixel OR cpen
6 COMP2	0	NOP
	1	pixel ← pixel XOR cpen XOR bpen
7 CLEAR2	0	NOP
	1	Pixel ← pixel AND NOT cpen

where cpen = current pen (primary drawing pen or text pen) or "dither pixel" (see note below).
spen = secondary drawing pen
bpen = background pen

NOTE

Where areas or lines are drawn with dithering, the color of an individual pixel within the dither pattern is the "CPEN" for that pixel (e.g., areas filled with dithered orange are drawn with a current pen that alternates between RED and YELLOW from pixel to pixel). The pattern value = 1 for all pixels in a dithered area.

The effects of the various drawing modes will be demonstrated using the following conditions:

- A portion of a raster line exists with pixel values:

0 0 0 0	0 0 0 0	Blue Plane
1 1 1 1	0 0 0 0	Green Plane
1 1 1 1	0 0 0 0	Red Plane
3 3 3 3	0 0 0 0	

- A dotted line will be drawn with line pattern 7 (Ec*m7B):

x - x - x - x -

"x" means pattern value = 1
"-" means pattern value = 0

- Primary pen = 001_B = 1 (Ec*mx) = Red
- Secondary pen = 010_B = 2 (Ec*my) = Green
- Background pen = 110_B = 6 (Ec*eb) = Cyan

Mode 0. NOP (Picture Protect no change to raster)

Mode 1. CLEAR1—
If the overlaying bit = - No change
If the overlaying bit = x Pixel is drawn in the current background color

CLEAR1 mode is used for simple erasures.

3 3 3 3 0 0 0 0	Pixel value of raster
x - x - x - x -	Pattern drawn in CLEAR1 mode
6 3 6 3 6 0 6 0	New pixel values

Mode 2. JAM1—
If the overlaying bit = - No change
If the overlaying bit = x Pixel = Primary pen

JAM1 is the default drawing mode.

3 3 3 3 0 0 0 0	Pixel value of raster
x - x - x - x -	Pattern drawn in JAM1 mode
1 3 1 3 1 0 1 0	New pixel values

Mode 3. COMP1—
If the overlaying bit = - No change
If the overlaying bit = x NOT pixel

COMP1 performs a contrast function, however, the vectors are not guaranteed to contrast with the background color. (See Mode 6 COMP2). If you redraw a line a second time in COMP1 mode, the original pixel values will be restored.

3= 0 1 1	0= 0 0 0
NOT 3= 1 0 0	NOT 0= 1 1 1

NOT 3 = 4, and
NOT 0 = 7

3 3 3 3 0 0 0 0	Pixel value of raster
x - x - x - x -	Pattern drawn in COMP1 mode
4 3 4 3 7 0 7 0	New pixel values

NOTE

If the background pen were 4 or 7, there would be no contrast between the background and the new line.

Mode 4 JAM2—
If the overlaying bit = - Pixel = Secondary pen
If the overlaying bit = x Pixel = Primary pen

JAM2 causes special area fill patterns and line types to be drawn with the secondary pen.

3 3 3 3 0 0 0 0	Pixel value of raster
x - x - x - x -	Pattern drawn in JAM2 mode
1 2 1 2 1 2 1 2	New pixel values

Mode 5. OR—
If the overlaying bit = - No change
If the overlaying bit = x pixel is OR'ed with primary pen

OR mode causes intersections of vectors and area fills to take on a new color.

When pixel value 3 = 0 1 1, is OR'ed with the
primary pen = 1 = 0 0 1, the result is
0 1 1 = 3.

When pixel value 0 = 0 0 0, is OR'ed with the
primary pen = 1 = 0 0 1, the result is
0 0 1 = 1.

3 3 3 3 0 0 0 0	Pixel value of raster
x - x - x - x -	Pattern drawn in OR mode
3 3 3 3 1 0 1 0	New pixel values

Mode 6. COMP2—
If the overlaying bit = - No change
If the overalying bit = x Pixel is exclusively OR'ed (XOR'ed) with primary pen (XOR'ed with background pen)

COMP2 mode performs a contrast function in which the new line is guaranteed not to be the background color.

When the primary pen 1 = 0 0 1 is XOR'ed with the
background pen 6 = 1 1 0, the result is
1 1 1 = 7 which is XOR'ed
with the pixel value 3 = 0 1 1, and the result is
1 0 0 = 4.

When "7" is XOR'ed with the pixel value '0', the result is "7".

NOTE

Exclusive OR can be thought of as binary addition without carry.

Therefore,

3 3 3 3 0 0 0 0	Pixel value of raster
x - x - x - x -	Pattern drawn in COMP2 mode
4 3 4 3 7 0 7 0	New pixel values

Redrawing the line a second time in COMP2 mode results in the restoration of the original raster pixel values.

(From above . . .)
Primary Pen XOR'ed with
Background Pen = (7) = 1 1 1
XOR'ed with new pixel (4) = 1 0 0
1 0 0 = 4

and, (7) = 1 1 1
XOR'ed with new pixel (7) = 1 1 1
0 0 0 B = 0

Therefore,

4 3 4 3 7 0 7 0	Pixel value of raster
x - x - x - x -	Pattern redrawn in COMP2 mode
3 3 3 3 0 0 0 0	Result

Mode 7. CLEAR2—
If the overlaying bit = - No change
If the overalying bit = x Pixel AND'ed with NOT primary pen

CLEAR2 mode has the effect of clearing the bits which correspond to the primary pen.

primary pen = 1 = 0 0 1

NOT primary pen = 6 = 1 1 0
AND pixel = 3 = 0 1 1
0 1 0 = 2

NOT primary pen = 6 = 1 1 0
AND pixel = 0 = 0 0 0
0 0 0 = 0

3 3 3 3 0 0 0 0	Pixel value of raster
x - x - x - x -	Pattern drawn in CLEAR2 mode
2 3 2 3 0 0 0 0	New pixel Values

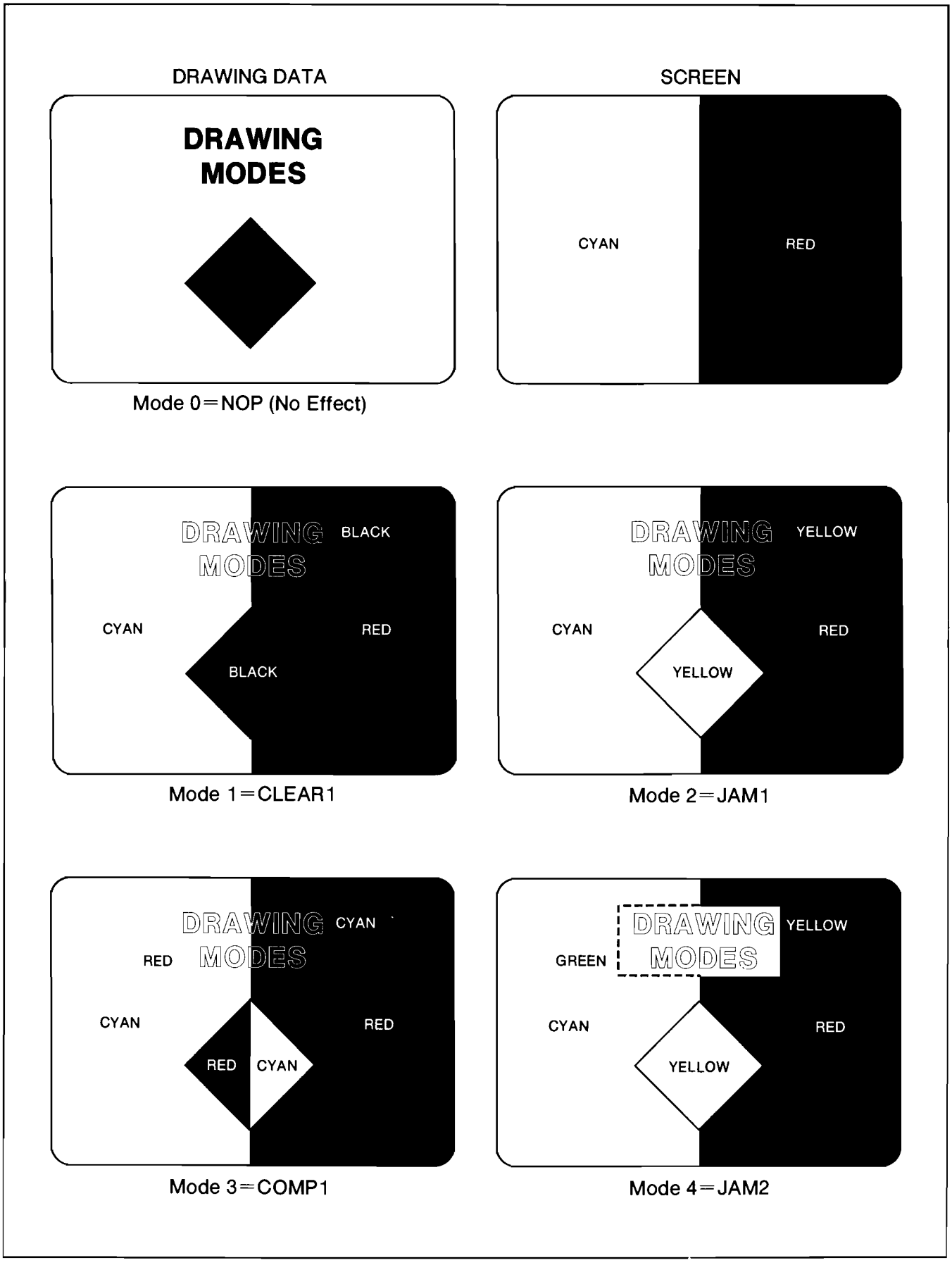
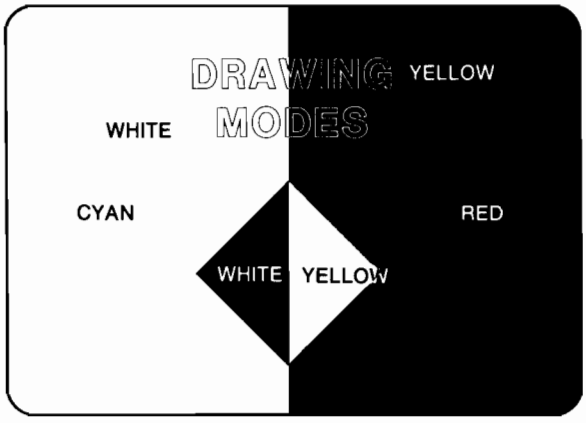
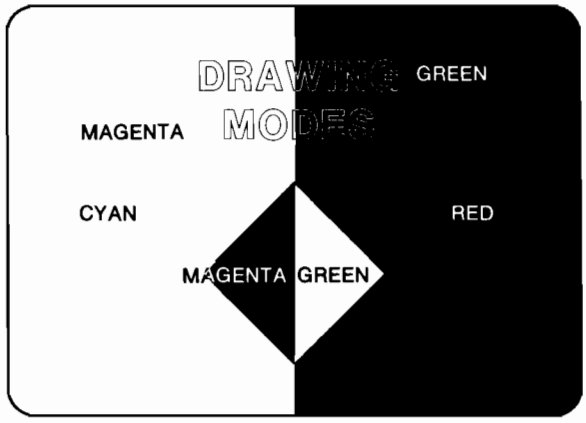


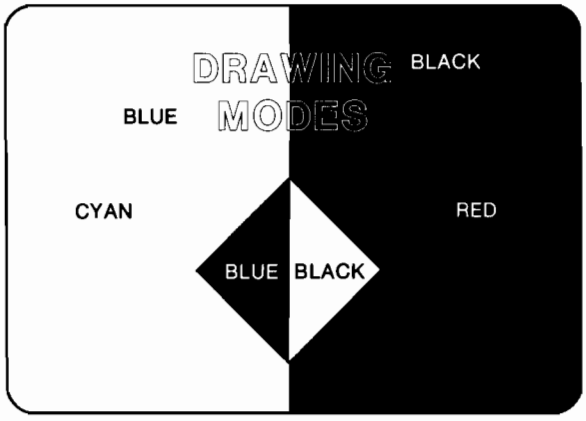
Figure 5-14. Examples of Drawing Modes



Mode 5=OR



Mode 6=COMP2



Mode 7=CLEAR2

PRIMARY PEN	= Yellow
TEXT PEN	= Yellow
SECONDARY PEN	= Green
BACKGROUND PEN	= Black

Figure 5-14. Examples of Drawing Modes (Continued)

GRAPHICS TEXT

Text strings can be written directly into the graphics image memory. An internal character generator converts the ASCII codes into a dot matrix representation which is drawn as vectors. The character set includes upper and lower case (95 characters) and the national characters shown in Appendix C. The national character sets are selected by the Terminal Configuration menu discussed in Section II. The characters will be drawn as a 5 by 7 matrix in a 7 by 10 cell, with descenders for lower case. This character set is in addition to the normal alphanumeric character set. While this character set may seem redundant, it offers the following advantages:

- Characters can be drawn at any dot position, rather than the 24 by 80 alphanumeric character positions.
- Characters can be rotated in multiples of 90 degrees.
- Characters can be scaled in size, from 1 to 8 times.
- Characters can be slanted 45 degrees for an italics-like effect.
- Lines of characters can be right, left, or center justified.

Figure 5-15 shows the graphics character set.

Keyboard Control of Graphics Text

Graphics text can be entered directly from the keyboard. The backspace, carriage return, and line feed functions work as expected (even on inverted text), making it easy to add or edit titles and labels. A summary of escape sequences and keyboard operations affecting Graphics Text Mode is given in table 5-7.

Table 5-7. Graphics Text Functions

Key	Description
ESC • d S	Selects the graphics image memory as the destination for all text. Characters entering from the keyboard or datacomm, are drawn as vectors in the graphics memory using the current text size and angle (see below). The graphics cursor indicates the position of the next character. Moving the graphics cursor will cause the next text line to begin at the new cursor position. The carriage return, line feed and backspace functions work normally.
ESC • d T	Terminates Text Mode.
ESC • m <size> M	Increases the character size from 1 to 8X. The smallest character is a 5 by 7 matrix in a 7 by 10 cell. Increasing the size makes the dots bigger while the character is still drawn as a 5 by 7 matrix.
ESC • m <orientation> N	Sets the character orientation (multiples of 90 degrees).
ESC • m 0	Turns slant on.
ESC • m P	Turns slant off.
ESC • n X	Sets graphics text pen number that selects color of pen.
TAB	Spaces one graphics text character to the right. (The actual direction of movement will depend on the text orientaiton.)
CTRL K	(Vertical Tab). Spaces one graphics text line up. (The actual direction of movement will depend on the text orientation.)

In addition, the following keys function in the same manner as for alphanumeric text characters:

RETURN, LINE FEED, BACK SPACE, ←, →, ↑, ↓

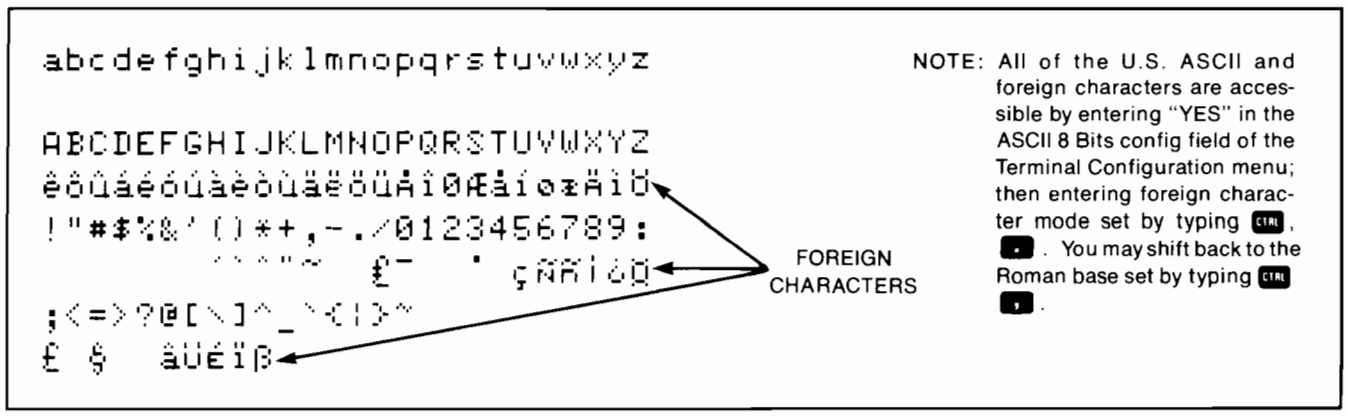


Figure 5-15. Graphics Text Characters

Program Control of Graphics Text

All of the parameters for graphics text can be set programmatically. Most commands are of the form: Esc * m <parameter>< command>. The command can be alone or part of another Esc * m sequence. Graphics text can also be drawn in color using the Esc *nX sequence.

SIZE. The ASCII characters 1 through 8 specify the character size for graphics text. A "1" indicates the smallest character, a 5 by 7 dot matrix character in a 7 by 10 cell. Increasing the size increases the size of the dots. If a text size of 1 is specified, each dot in the cell is one dot on the screen. A size of 2 uses 4 screen dots for each character dot (2 X 2), and so on (see figure 5-16). A size of "1" is the default.

Set Graphics Text Size: Esc * m <size> m

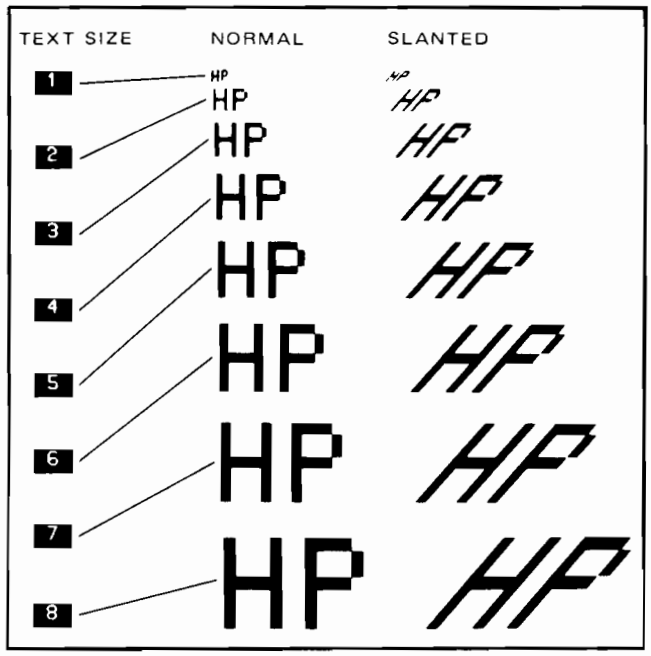


Figure 5-16. Graphics Text Sizes

TEXT DIRECTION. This command uses the ASCII characters 1 through 4 to specify the text orientation (see figure 5-17). This also changes the direction of line feed, carriage return, and backspace.

1 — Normal (upright, the default)
2 — Rotated 90 degrees counter clockwise
3 — Rotated 180 degrees counterclockwise (inverted)
4 — Rotated 270 degrees counter clockwise

Set Graphics Text Orientation: Esc * m <orientation> n

SLANT. The graphics text characters can be slanted 45 degrees for an italics effect.

Turn On Graphics Text Slant: Esc * m o

Turn Off Graphics Text Slant: Esc * m p

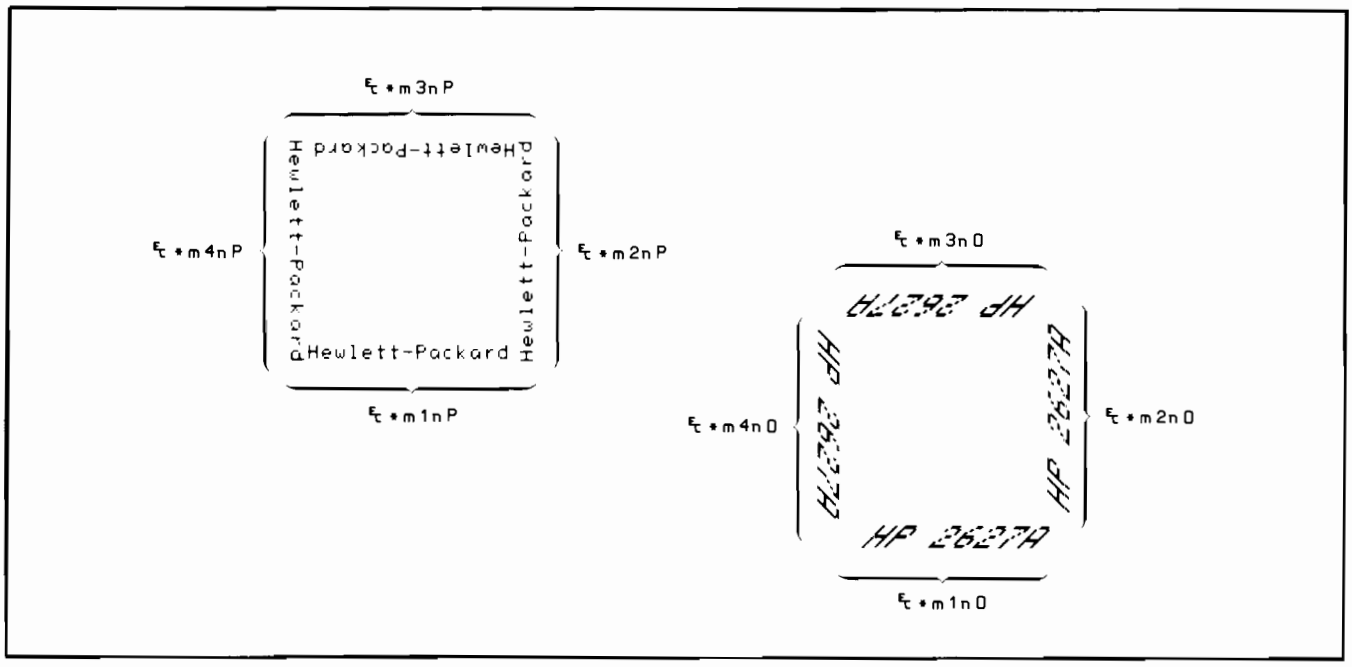
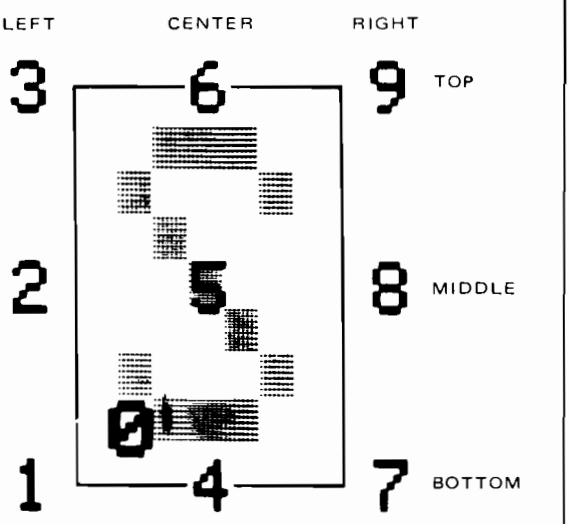


Figure 5-17. Graphics Text Direction

JUSTIFICATION/ORIGIN. Text strings can be automatially right or left justified, or centered about a specified point. An ASCII character 0 through 9 indicates the origin (justification and base line) for characters with respect to the current pen position. This function is useful when drawing labels. (Refer to the Label command.)

Set Graphics Text Justification: ESC * m <origin> q

If text is left justified, the current pen position is the left margin. Center causes the label to be centered on the pen position. Right justify selects the pen position as the right margin. Bottom, middle, and top select the base line for the line of text.



The numbers 1-9 represent the cursor position with respect to the character cell used for graphics text characters. The number 0 represents the cursor position with respect to the character (not this cell).

For example, if text was to be right justified and set with a base line on top of the normal character position, the number "9" would be used. Figure 5-18 illustrates the various text positions.

When centering or right justification is used, the text strings are buffered (stored) until all of the characters in the string have been received. The string end is detected by a CR or LF. The string is not displayed until the CR or LF is received. This may be confusing when entering text from the keyboard. The maximum length of a string when center or right justifying is 73 characters (not including the CR(LF)). In all cases, data written beyond the edge of the screen is lost. There is no automatic RETURN when the screen boundary is reached.

TEXT COLOR. The color of the graphics text can be set independently of the primary and secondary pens with the command:

ESC * n <pen number> x

where <pen number> is an integer in the range (– 32768, . . ., 32767); the low three bits are used to select a pen number in the range (0, . . ., 7).

Default: Primary Pen

If <pen number> is supplied, it is used for all subsequent text and labels until explicitly changed or graphics defaults are set.

Use primary pen for graphics text: ESC * n x

At power on, graphics hard reset, or set graphics default, graphics text is drawn with the primary pen. Until an ESC * n x sequence is received, the text pen "tracks" the primary pen.

TURNING GRAPHICS TEXT ON AND OFF. Graphics text mode can be turned on or off from a program. These two commands use the ESC * d sequence but are discussed here under graphics text for completeness.

On. This command will cause Graphics Text Mode to be turned on. All displayable characters will be stored in the graphics memory. The current drawing mode remains in effect until it is changed.

Text is drawn using the current text assignments for size and orientation. Graphics text mode accepts CR, LF, BS, HT, and VT as control characters. The ←, →, ↑, and ↓ keys can be used to position the graphics cursor in character increments.

Turn On Graphics Text Mode: ESC * d S

Turn Off Graphics Text Mode: ESC * d T

If the graphics cursor is moved, the graphics text margin is moved to the new cursor or pen position.

Characters are drawn using the current drawing mode. In most drawing modes, entering a character, backspacing, and entering a second character causes an overstrike. If JAM2 mode is used, the new character will replace the old character. JAM2 mode draws with two colors.

If a lower case "s" is used, additional escape parameters can be appended to the sequence. Otherwise the next characters will be routed to the graphics memory.

Example: Turn on graphics text mode, position cursor at 100,100, and type "this is graphics text".

```
ESC*dsk 100,100 0←position cursor at 100,100
        ↑
        turn on graphics cursor
```

Text Example: This is graphics text.

Off. This sequence turns off graphics text mode and restores normal alphanumeric operation.

Turn Off Graphics Text Mode: Esc * d t

NOTE

The ENTER key or modify mode do not work in graphics text mode.

GRAPHICS TEXT STATUS. You can check the current text settings with a graphics text status request. Refer to Secton VIII, Status, for additional information.

LABEL. This sequence is used to send a single record of graphics text to the terminal. The characters are stored in the graphics memory using the current text size, angle, slant, justification, and color. The label is drawn beginning at the current pen position.

Graphics Text Label: Esc * l <text string> % (F)

The record must end with a CR, LF, or both. Note that the actual directions moved following a CR or LF depend on the text orientation selected.

The maximum record length is 73 characters, not including the Esc * l preamble or the CR(LF).

Example: Esc*lThis is a sample label%F

SELECTING THE GRAPHICS DEFAULT PARAMETERS

Graphics parameters can be set to their default (power on or full reset) values (see table 5-8) by issuing the following sequence:

Esc*m <default flag> r

Some of the current graphics modes and settings can be obtained with graphics status requests. Graphics status requests are described in Section VIII, "Status". It may be desirable to reselect graphics settings before sending graphics data to the terminal.

GRAPHICS HARD RESET

Graphics hard reset performs exactly the same functions as if hard reset was initiated for graphics only. It sets all graphics parameters to their default values as specified for Esc*mr (see table 5-8) plus the following:

1. Clears raster memory buffer.
2. Primary drawing pen is positioned to location 0,0.

A graphics hard reset may only be performed programmatically:

Esc * w r

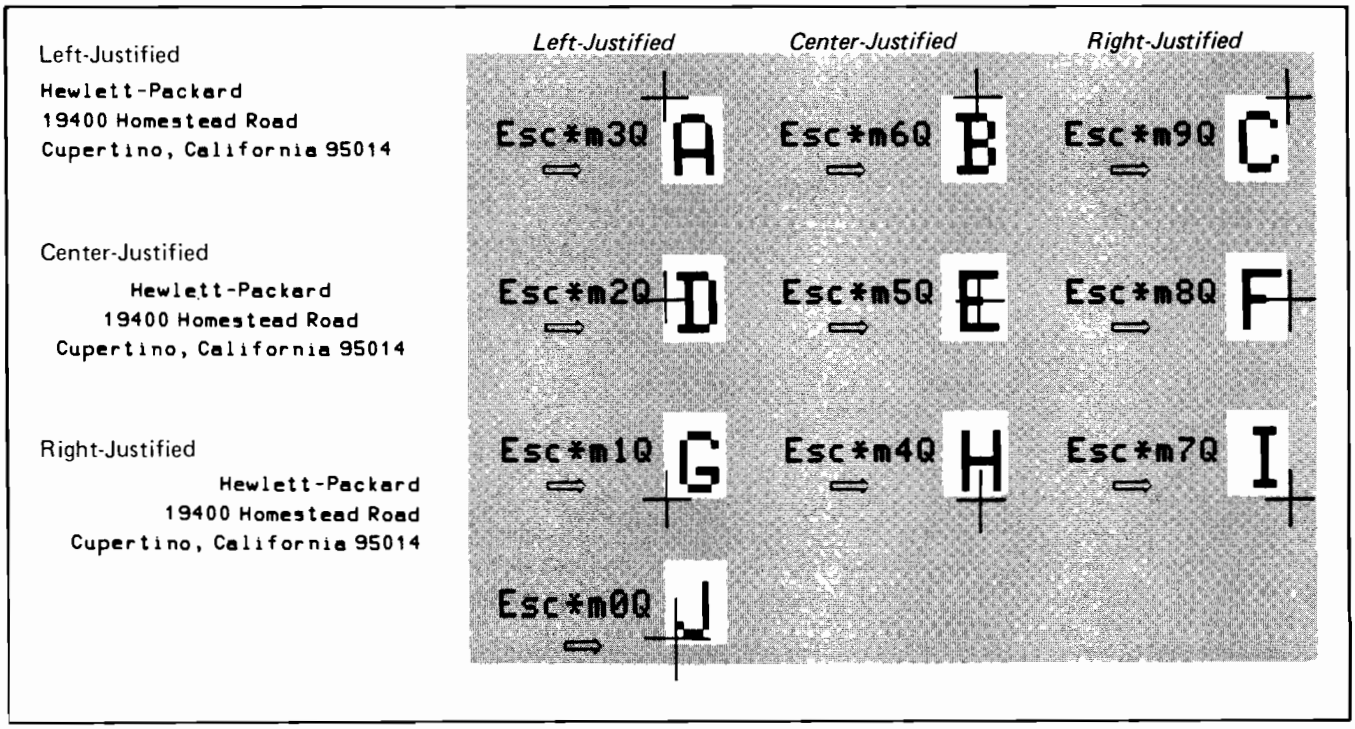


Figure 5-18. Graphics Text Justification

Table 5-8. Graphics Parameter Default Values

PARAMETER	DEFAULT VALUES
**Pen Condition	down
**Line Type	1(solid)
**Drawing Mode	2(JAM1)
**User Defined Line Pattern	255,1
**Area Fill Type	2(user defined pattern)
**User Defined Area Fill Pattern	255, 255, ..., (solid)
**User Defined Dither Pattern	1,1,1 (solid white)
**Current Dither Pattern	1(user defined)
**Background Pen	0(black)
**Primary Pen	7(white)
**Secondary Pen	0(black)
**Boundary Pen	off
**Graphics Text	off
**Text Size	1
**Text Direction	1
**Text Origin	1(left,bottom)
**Text Slant	off
**Text Color	primary pen
Relocatable Origin	0,0
Alpha Video	on
Graphics Video	on
Alpha Cursor	on
Graphics Cursor	off
Graphics Cursor Address	0,0
Rubberband Line	off
Compatibility Mode	
Page Full Straps	0(out)
GIN Strap	0(CR only)

**If a "1" is used for the <default flag> (Esc*m<1>r), only these parameters are defaulted. If no value is used (Esc*mr), all parameters are defaulted.

GRAPHICS FUNCTIONS DISPLAY FUNCTIONS MODE

The DISPLY FUNCTN key (at the Modes level) can be used to display the graphics escape sequences or the action of graphics control keys. The control sequences are entered into the alphanumeric display each time a command is executed. Table 5-9 lists the graphics control sequences that are generated when DISPLAY FUNCTION is on.

Table 5-9. Graphics Control Sequences With Display Functions On

Key	Sequence	Description
▲ ► ◄ ▼	none	Graphics cursor controls
CURSOR FAST	none	Graphics cursor fast
[GRAPH CURSOR]	Esc * dL	off
	Esc * dK	on
[GRAPH DSPLY]	Esc * dC	on
	Esc * dD	off
[GRAPH CLEAR]	Esc * dA	
[ALPHA DSPLY]	Esc * dF	off
	Esc * dE	on

Figure 5-19 shows the sequences generated when drawing a simple box. The graphics cursor is initially on and positioned at 0,0.

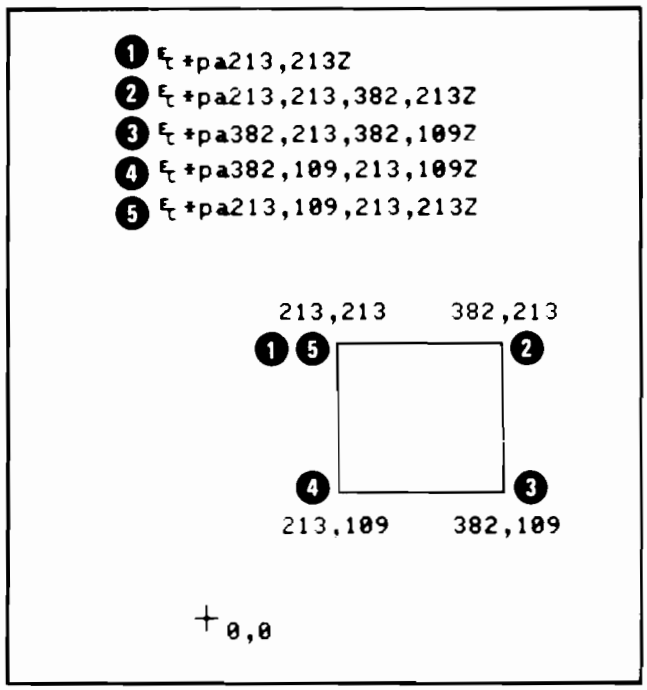


Figure 5-19. Displaying Graphics Sequences

GRAPHICS HARDCOPY OPERATIONS

There are two methods of obtaining a hardcopy of the contents of graphics memory. One method uses the function keys and graphics keys on the keyboard. The other method uses escape sequences which may be coded into a program running on a host computer. The hardcopy may be output from an external printer connected to the external printer port at the rear of the terminal.

The external printers that may be used for graphics hardcopy are the HP 2671G and HP 2673A.

Initiating a Transfer from the Keyboard

The GRAPH COPY key on graphics/numeric pad initiates the graphics transfer. The keypad must be in graphics mode for the GRAPH COPY function to be performed. SHIFT NUM toggles the function of the keypad between graphics mode and numeric mode.

The destination(s) may be selected by pressing AIDS, f1 (device control), f3 (to devices). You may then select f2 (TO EXT DEV), which selects the external printer port.

Using the Esc & p Escape Sequences

Coding a program to transfer graphics data to the external printer requires selecting the graphics memory as the source and the external printer as the destination.

alpha display as the source:	3s
graphics memory as the source:	7s
external printer as the destination:	4d

Example: Define graphics memory as source and the external printer as destination.

Esc & p 7s 4D

After the source and destination are defined, the transfer is initiated by either:

Esc & p F (copy file from source to destination)

or

Esc & p M (copy all from source to destination)

Note that an escape sequence is terminated with an uppercase character. Also, you may combine the source and destination assignments and the transfer initiation in one escape sequence:

Esc & p 7s 4d F

COMPATIBILITY MODE

Compatibility Mode allows the terminal to plot data intended for a terminal using a display with 1024 by 1024 addressable points. This mode makes it possible to use graphics programs developed for use with other graphics terminals with a minimum of reprogramming.

The terminal operates in two submodes while in Compatibility Mode. In Alphanumeric mode the terminal simply displays alphanumeric data on the screen as in normal operation. In Graphics mode the terminal responds to alphanumeric data as vector coordinates. Normally the terminal will be switched between these modes to display messages, plot graphics figures, and then display additional messages. These modes are controlled with several control sequences. (These sequences are ignored or acted on differently if the terminal is not set for Compatibility Mode.) Table 5-10 lists the terminal's responses to Compatibility Mode control sequences.

If delays are required, the baud rate can be lowered or fill characters added to prevent data loss when operating the terminal at high speeds. Refer to Section VI, Data Communications.

Vectors are drawn using the current line type, color pen, and line drawing mode. This gives you the capability of drawing dotted and dashed lines, etc. by changing the program to send the additional escape sequences.

Compatibility Mode is turned on by selecting either scaled or unscaled operation. Escape sequences controlling Compatibility Mode begin with Esc *t. This preamble is then followed with one or more commands. These commands are listed in table 5-11. As in all other escape sequences, a capital letter ends the sequence. Figure 5-20 contains examples of typical escape sequences.

Compatibility Mode Configuration

Compatibility Mode operation is controlled by Compat(P,Q) field in the Terminal Configuration menu. This field can be set manually or programmatically using the Esc & s . . ." sequence shown in table 5-11. The P and Q straps determine the terminal's mode of operation after being initialized (power up or full reset). The straps are interpreted as follows:

STRAPS		DESCRIPTION
P	Q	
0	0	Normal graphics operation
0	1	Unscaled Compatibility Mode
1	0	Scaled Compatibility Mode
1	1	Normal graphics operation

In addition, when in Compatibility Mode, you can select the following optional capabilities:

GRAPHIC INPUT TERMINATOR. You can select the terminator sent by the terminal following the input of cursor address information. The terminator can be a CR, CR and EOT, or no terminator.

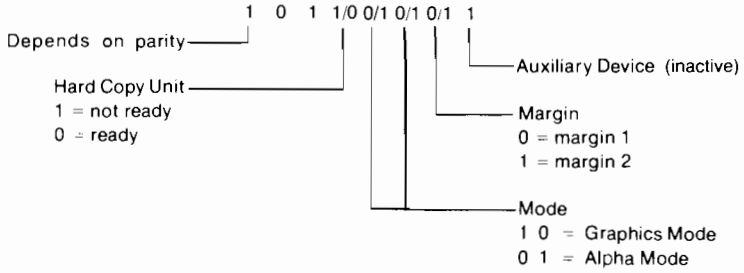
PAGE FULL BUSY. When this strap is in, the keyboard will be locked after the 35th line of text is received from the computer. The terminal can be cleared by pressing the GRAPH CLEAR key. This strap is ignored in Unscaled Mode.

PAGE FULL BREAK. When this strap is in, the terminal will send a 200ms break signal to the computer after the 35th line of text is displayed. The terminal may also be set to BUSY (see Page Full Busy). When out, the strap will cause the cursor to home and the next 35 lines of text to be set with a left margin at x = 259. This strap is ignored in Unscaled Mode.

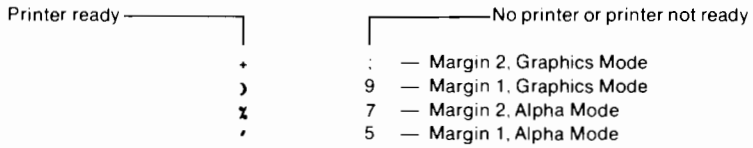
The commands to control these strap options are listed in table 5-11. Refer to the manual for the replaced graphics terminal for additional information on the operation of these straps and how they should be set.

Table 5-10. Compatibility Mode Control Sequences

CONTROL SEQUENCE	DESCRIPTION	RESPONSE
ESC ENQ	Read status and alpha cursor position	<status byte><HI X><LO X> <HI Y><LO Y><terminator>



The terminal will return one of the following characters as the status byte:



ESC ETB (20 ms delay) ESC ENQ	Read graphics cursor position	<HI X><LO X><HI Y><LO Y><terminator>
ESC ETB	Read graphics cursor position when key struck	<KEY><HI X><LO X><HI Y><LO Y><terminator>
ESC BS	Make hardcopy	
ESC FF	End graphics mode, clear screen, and home cursor	
GS	Go into graphics mode (draw vectors)	
US	Go into alpha mode	
BS	Backspace (H) Moves 1 space left (14 units)	
HT	Horizontal Tab (I) Moves 1 space right (14 units)	
CAN	End graphics mode	
LF	Line Feed (J) Moves 1 line down (22 units)	
VT	Vertical Tab (K) Moves 1 line up (22 units)	

NOTES

The terminal will normally respond with an ACK character when an ENQ character is received. Compatibility Mode disables the terminal's ENQ/ACK handshake. Compatibility Mode causes most control codes to be ignored.

The Read Status, alpha cursor position, and graphic cursor position cause block transfers to the computer system. If the computer system does not use the DC1/DC2/DC1 handshake, `InhHndShk(G)` and `InhDC2(H)` in the Terminal Configuration menu must be "YES" for these transfers to occur. (Refer to "Terminal Configuration Menu" in Section II).

Table 5-11. Commands for Selecting Compatibility Mode

COMMAND	CODE
TURN SCALED COMPATIBILITY MODE ON (P open)	Et & s 1 p 0 Q
TURN UNSCALED COMPATIBILITY MODE ON (Q open)	Et & s 0 p 1 Q
TURN COMPATIBILITY MODE OFF (P,Q closed)	Et & s 0 p 0 Q
The following commands simulate straps used on other graphics terminals:	
SET GRAPHICS INPUT TERMINATOR STRAP 0 — Carriage return only (Normal position) 1 — Carriage return and EOT 2 — No carriage return, no EOT	Et * t <byte1> a
SET PAGE FULL BREAK STRAP 0 — Out (Normal position) 1 — In	Et * t <byte1> b
SET PAGE FULL BUSY STRAP 0 — Out (Normal position) 1 — In	Et * t <byte1> c
NOP	z

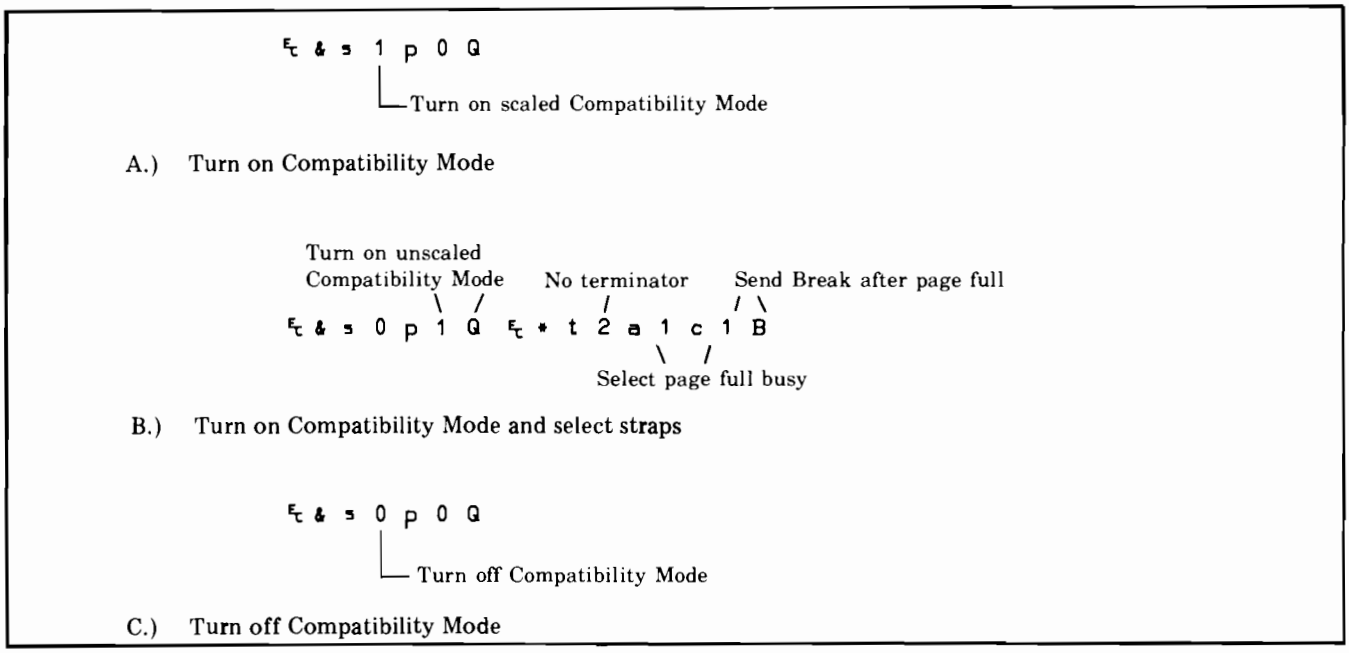


Figure 5-20. Turning On Compatibility Mode

Graphic Data

There are differences in display size (512 x 390 for the HP 2627A versus 1024 x 780 for other terminals) and line length (24 lines of 80 characters for the HP 2627A versus 35 lines of 74 characters for other terminals). See figure 5-21.

Graphic data can be drawn either scaled or unscaled. Scaling divides X and Y coordinates by 2. This maps the 1024 x 780 display into 512 by 390. This allows a program written for the 1024 x 780 terminal to run unchanged, and still display the entire picture (with some loss in resolution). See figure 5-22.

Unscaled mode shows a 512 by 390 subset of the 1024 × 780 picture. The area this covers can be changed by modifying the value of the relocatable origin (and redrawing the picture). The relocatable origin is subtracted from all incoming coordinates in unscaled mode. If this is set to 0,0 (the default) the range X = 0 to 511, Y = 0 to 389 will be displayed (see figure 5-23).

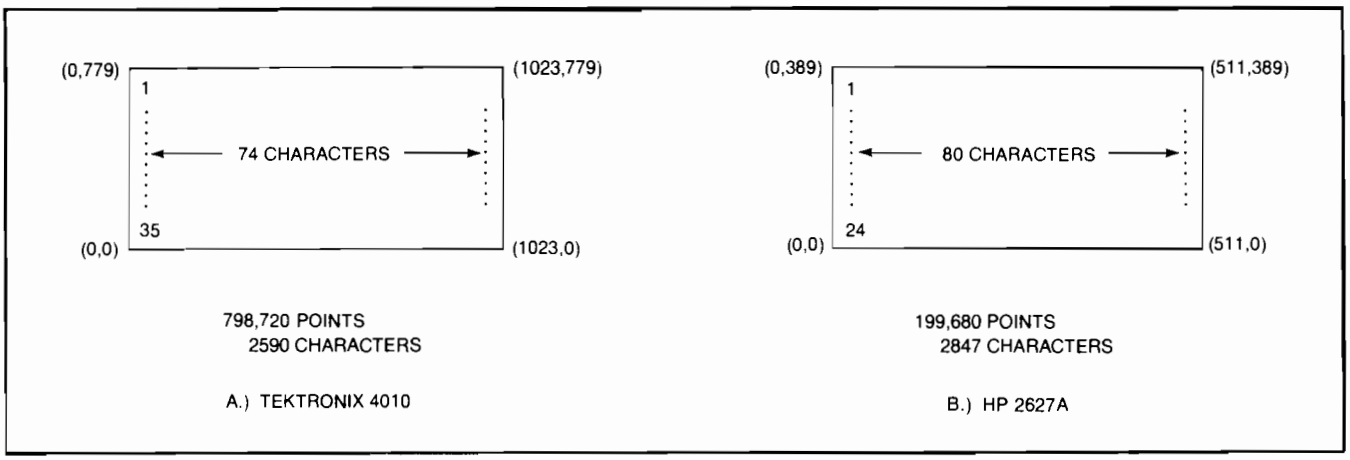


Figure 5-21. Comparison of a Terminal with 1024 x 780 Display and the HP 2627A

Setting the origin to 0,360 would cover the X = 0 to 511, Y = 360 to 749. To display an area larger than 512 × 390, you must change the scaling statements in the program.

Graphics Data Format

In Compatibility mode the graphics data is formatted as two-byte coordinate values. The lower five bits of each byte are used to make a 10 bit (0-1023) coordinate. Data sent to the terminal must have the "Y" coordinate sent first; <Upper Y> <Lower Y> <Upper X> <Lower X>.

When data is returned to the computer (cursor position, etc.), the X coordinate is returned first; <Upper X> <Lower X> <Upper Y> <Lower Y>.

Data bytes sent to the terminal use bits 6 and 7 to indicate the byte is an Upper byte, a lower Y, or a lower X. Bit 8 (parity) is not used.

Bits		
7	6	
0	1	Upper X or Y byte
1	0	Lower X byte
1	1	Lower Y byte

These identifying bits allow you to send only the changed portion of a four byte address. The following data bytes must always be sent:

- Lower X byte
- Any changed byte
- Lower Y byte if the Upper X byte has changed

Table 5-12 can be used to determine address bytes. For example, to plot the points A (0,0), B (0,31), C (256,31), D (256,0) the sequence shown in figure 5-24 would be used:

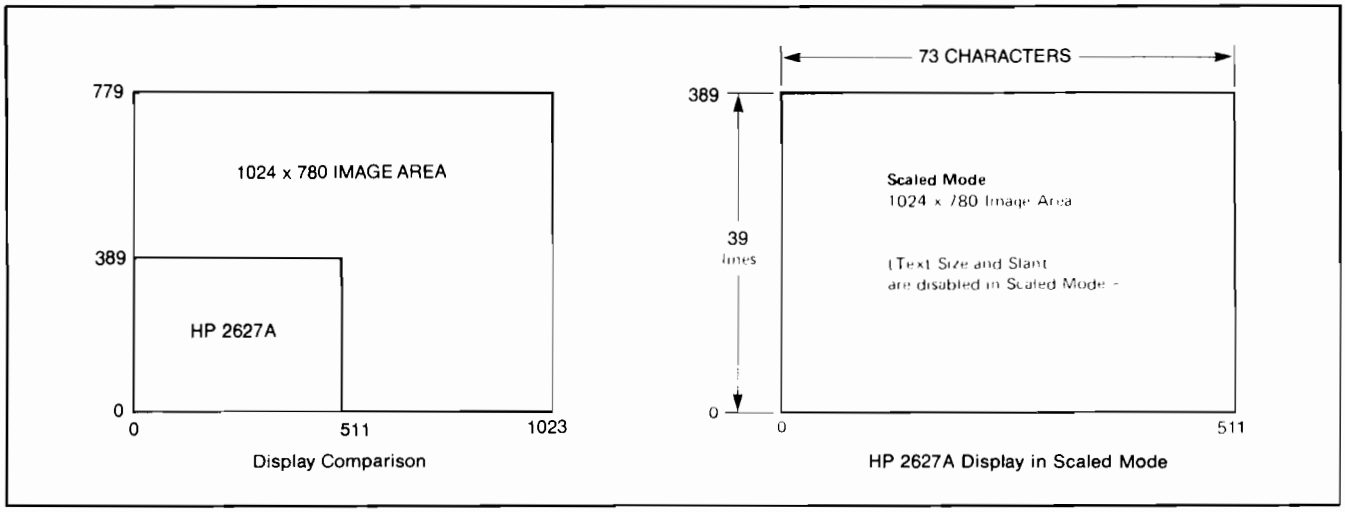
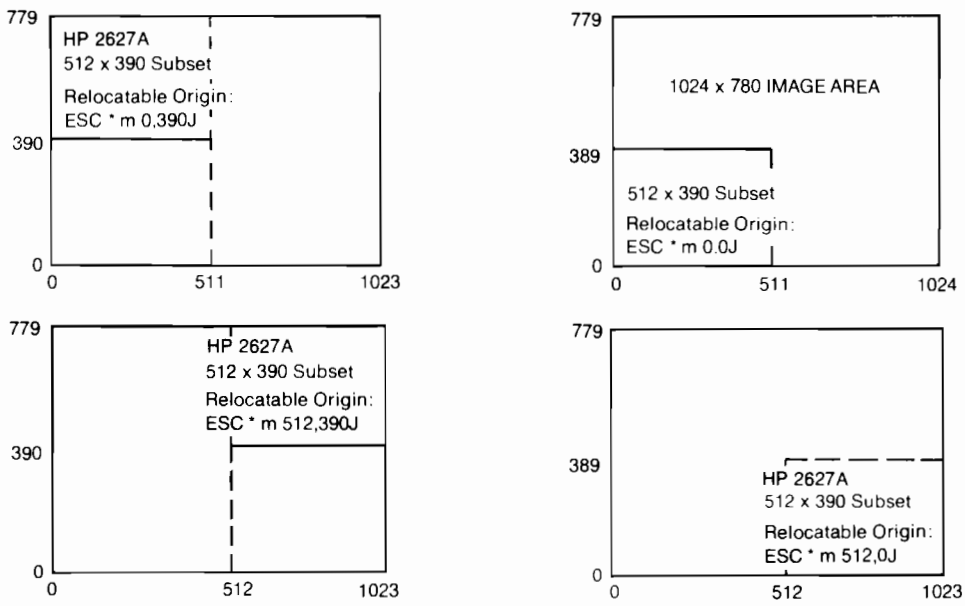


Figure 5-22 Scaled Data



Changing the Relocatable Origin (ESC * m x,y J) *on 2627's Display to Cover the Entire 1024 x 780 Display in Unscaled Mode*

Figure 5-23. Unscaled Data

Text

Text can be placed in either the alphanumeric memory or in the graphics memory. If the terminal is set for alphanumeric text, the text will be sent to the alphanumeric memory. This is generally the most useful, as text can be scrolled, edited, erased, etc. without affecting the graphics image. If you select graphics text (ESC * d 5), text will go into the graphics memory. Text to be written to the graphics memory can be scaled or rotated. (Refer to Graphics Text in this section for additional information.)

When text is written to the graphics memory, the graphics cursor is moved to indicate where the next character will be stored. (The alphanumeric cursor is only used when data is stored in the alphanumeric memory.) This differs from terminals that have only one mode for text and display the graphics cursor only when waiting for graphic input from the user.

SCALED MODE GRAPHICS TEXT. In Scaled Mode, text is initially written into the graphics memory, the size is fixed to allow for 35 lines of text. The text angle is set at 0 degrees and unslanted. The text origin is set to the left and bottom. These settings allow the "Page Full" feature to work properly and existing software to run without changes. If you do not require the Page Full feature, you can not change the text settings. You can redirect the text to the alphanumeric memory.

UNSCALED MODE GRAPHICS TEXT. In Unscaled Mode, the text size is unchanged and graphics text mode is not initially turned on. Text is stored in the alpha numeric memory unless the graphics text mode is specifically enabled.

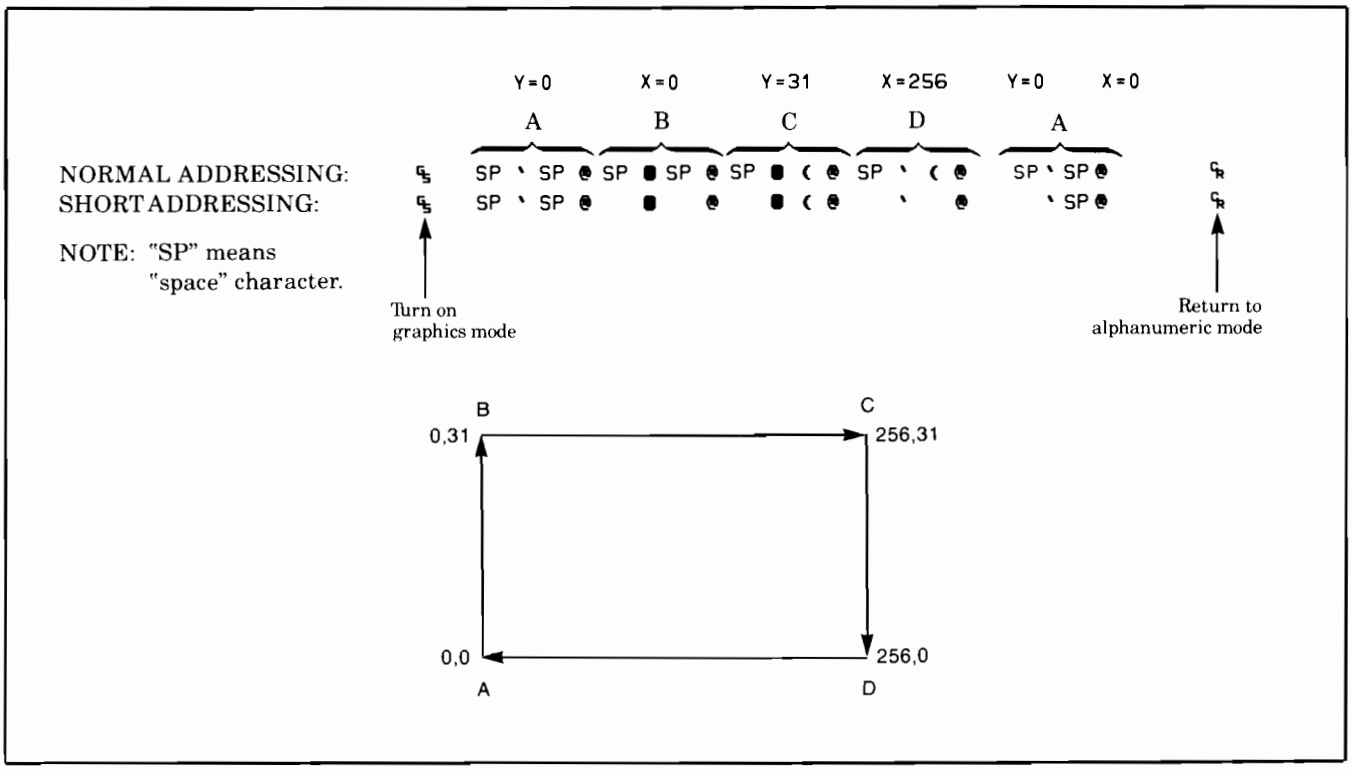


Figure 5-24 Graphics Data Format

Table 5-12. Coding of Compatibility Mode Graphics Data

X or Y Coordinate																Low Order Y		Low Order X	
																DEC.	ASCII	DEC.	ASCII
0	32	64	96	128	160	192	224	256	288	320	352	384	416	448	480	96	`	64	@
1	33	65	97	129	161	193	225	257	289	321	353	385	417	449	481	97	a	65	A
2	34	66	98	130	162	194	226	258	290	322	354	386	418	450	482	98	b	66	B
3	35	67	99	131	163	195	227	259	291	323	355	387	419	451	483	99	c	67	C
4	36	68	100	132	164	196	228	260	292	324	356	388	420	452	484	100	d	68	D
5	37	69	101	133	165	197	229	261	293	325	357	389	421	453	485	101	e	69	E
6	38	70	102	134	166	198	230	262	294	326	358	390	422	454	486	102	f	70	F
7	39	71	103	135	167	199	231	263	295	327	359	391	423	455	487	103	g	71	G
8	40	72	104	136	168	200	232	264	296	328	360	392	424	456	488	104	h	72	H
9	41	73	105	137	169	201	233	265	297	329	361	393	425	457	489	105	i	73	I
10	42	74	106	138	170	202	234	266	298	330	362	394	426	458	490	106	j	74	J
11	43	75	107	139	171	203	235	267	299	331	363	395	427	459	491	107	k	75	K
12	44	76	108	140	172	204	236	268	300	332	364	396	428	460	492	108	l	76	L
13	45	77	109	141	173	205	237	269	301	333	365	397	429	461	493	109	m	77	M
14	46	78	110	142	174	206	238	270	302	334	366	398	430	462	494	110	n	78	N
15	47	79	111	143	175	207	239	271	303	335	367	399	431	463	495	111	o	79	O
16	48	80	112	144	176	208	240	272	304	336	368	400	432	464	496	112	p	80	P
17	49	81	113	145	177	209	241	273	305	337	369	401	433	465	497	113	q	81	Q
18	50	82	114	146	178	210	242	274	306	338	370	402	434	466	498	114	r	82	R
19	51	83	115	147	179	211	243	275	307	339	371	403	435	467	499	115	s	83	S
20	52	84	116	148	180	212	244	276	308	340	372	404	436	468	500	116	t	84	T
21	53	85	117	149	181	21	245	277	309	341	373	405	437	469	501	117	u	85	U
22	54	86	118	150	182	214	246	278	310	342	374	406	438	470	502	118	v	86	V
23	55	87	119	151	183	215	247	279	311	343	375	407	439	471	503	119	w	87	W
24	56	88	120	152	184	216	248	280	312	344	376	408	440	472	504	120	x	88	X
25	57	89	121	153	185	217	249	281	313	345	377	409	441	473	505	121	y	89	Y
26	58	90	122	154	186	218	250	282	314	346	378	410	442	474	506	122	z	90	Z
27	59	91	123	155	187	219	251	283	315	347	379	411	443	475	507	123	{	91	[
28	60	92	124	156	188	220	252	284	316	348	380	412	444	476	508	124	|	92	\
29	61	93	125	157	189	221	253	285	317	349	381	413	445	477	509	125	}	93	]
30	62	94	126	158	190	222	254	286	318	350	382	414	446	478	510	126	~	94	^
31	63	95	127	159	191	223	255	287	319	351	383	415	447	479	511	127	■	95	_
32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	← DEC.			
SP	!	"	#	$	%	&	'	(	)	*	+	,	−	.	/	← ASCII			
High Order X & Y																			

X or Y Coordinate																Low Order Y		Low Order X	
																DEC.	ASCII	DEC.	ASCII
512	544	576	608	640	672	704	736	768	800	832	864	896	928	960	992	96	`	64	@
513	545	577	609	641	673	705	737	769	801	833	865	897	929	961	993	97	a	65	A
514	546	578	610	642	674	706	738	770	802	834	866	898	930	962	954	98	b	66	B
515	547	579	611	643	675	707	739	771	803	835	867	899	931	963	995	99	c	67	C
516	548	580	612	644	676	708	740	772	804	836	868	900	932	964	996	100	d	68	D
517	549	581	613	645	677	709	741	773	805	837	869	901	933	965	997	101	e	69	E
518	550	582	614	646	678	710	742	774	806	838	870	902	934	966	998	102	f	70	F
519	551	583	615	647	679	711	743	775	807	839	871	903	935	967	999	103	g	71	G
520	552	584	616	648	680	712	744	776	808	840	872	904	936	968	1000	104	h	72	H
521	553	585	617	649	681	713	745	777	809	841	873	905	937	969	1001	105	i	73	I
522	554	586	618	650	682	714	746	778	810	842	874	906	938	970	1002	106	j	74	J
523	555	587	619	651	683	715	747	779	811	843	875	907	939	971	1003	107	k	75	K
524	556	588	620	652	684	716	748	780	812	844	876	908	940	972	1004	108	l	76	L
525	557	589	621	653	685	717	749	781	813	845	877	909	941	973	1005	109	m	77	M
526	558	590	622	654	686	718	750	782	614	846	878	910	942	974	1006	110	n	78	N
527	559	591	623	655	687	719	751	783	815	847	879	911	943	975	1007	111	o	79	O
528	560	592	624	656	688	720	752	784	816	848	880	912	944	976	1008	112	p	80	P
529	561	593	625	657	689	721	753	785	817	849	881	913	945	977	1009	113	q	81	Q
530	562	594	626	658	690	722	754	786	818	850	882	914	946	978	1010	114	r	82	R
531	563	595	627	659	691	723	755	787	819	851	883	915	947	979	1011	115	s	83	S
532	564	596	628	660	692	724	756	788	820	852	884	916	948	980	1012	116	t	84	T
533	565	597	629	661	693	725	757	789	821	853	885	917	949	981	1013	117	u	85	U
534	566	598	630	662	694	726	758	790	822	854	886	918	950	982	1014	118	v	86	V
535	567	599	631	663	695	727	759	791	823	855	887	919	951	983	1015	119	w	87	W
536	568	600	632	664	696	728	760	792	824	856	888	920	952	984	1016	120	x	88	X
537	569	601	633	665	697	729	761	793	825	857	889	921	953	985	1017	121	y	89	Y
538	570	602	634	666	698	730	762	794	826	858	890	922	954	986	1018	122	z	90	Z
539	571	603	635	667	699	731	763	795	827	859	891	923	955	987	1019	123	{	91	[
540	572	604	636	668	700	732	764	796	828	860	892	924	956	988	1020	124	|	92	\
541	573	605	637	669	701	733	765	797	829	861	893	925	957	989	1021	125	}	93	]
542	574	606	638	670	702	734	766	798	830	862	894	926	958	990	1022	126	~	94	^
543	575	607	639	671	703	735	767	799	831	863	895	927	959	991	1023	127	■	95	_
48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	← DEC			
0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?	← ASCII			
High Order X & Y																			

Example: 340Y,70X is found as follows:

340Y = 42 (upper Y) 116 (Lower Y) 70X = 34 (Upper X) 70 (Lower X)

```
340Y,70X → * t " F
```

Data Communications

SECTION VI

INTRODUCTION

The term "data communications" (or "datacomm") refers to the transfer of data between the terminal and a host computer.

There are several ways to connect the terminal to a computer. To arrive at a particular way you must compare a number of factors and make a series of decisions. After selecting the necessary equipment and cables, you must then physically connect the terminal to the computer (or to the modem, if that is what you have chosen) and configure the terminal for use with the particular type of data communications link.

This section is divided into four parts:

1. The first is a general discussion that should help you decide what type of equipment and cabling you need for the communications link.
2. The second tells you how to physically install the terminal.
3. The third tells you how to configure the terminal to operate properly with the selected type of communications link.
4. The final part provides programming reference material for someone who is writing a datacomm driver or controller program to communicate with an the terminal in a point-to-point environment.

Before proceeding with the decision making process, it may help to briefly define the most important terminology as it pertains to data communications.

Communications Link: The means by which a terminal is connected to a host computer. This always includes some type of communications line (a coaxial cable, the public telephone network, or a leased telephone line), and it may also include a pair of modems (one at each end of the line).

Point-to-Point: A data communications configuration in which a single terminal is connected to a host computer over a communications link. The terminal is designed for use with a point-to-point communications link, such as a hardwired or modem connection.

Multipoint: A data communications configuration in which two or more terminals are "chained" together so as to share a communications link to a host computer. The HP 2627A does not support multipoint.

Asynchronous: A mode of transmission in which each data character is framed by a "start bit" and one or more "stop bits". The interval between successive data characters is random. The terminal is designed for use with an asynchronous communications link.

Full Duplex: A communications link in which data can be transmitted in both directions simultaneously.

Character Mode: When the terminal is operating in character mode, it sends data characters to the computer one at a time as they are typed into the keyboard.

Block Mode: When the terminal is operating in block mode, data characters typed into the keyboard are merely stored in display memory. When a block transfer is subsequently triggered (by the host computer or by pressing the ENTER key), a group of data characters is sent from the terminal to the computer as a block.

A point-to-point configuration is the standard form of data communications within the industry (it is sometimes referred to as a "Teletype-compatible" communications link). Point-to-point is supported by most computers. At any given time, it accomodates only one terminal per communications link; it may, however, operate in either character mode or block mode.

POINT-TO-POINT DECISIONS

Since the terminal only supports point-to-point configurations, you must now make a series of decisions illustrated in figure 6-1. Note that in the figure, the overall set of decisions is organized as a tree structure and that when you make a choice, you then follow the associated branch to the next set of alternatives.

For each desired point-to-point communications link you must decide whether you want a hardwired or modem connection. A hardwired connection, where feasible, is usually the lower cost alternative because it eliminates the use of modems and common carrier (telephone company) lines.

A major consideration in selecting which type of connection to use is the anticipated distance between the terminal and the computer. If the terminal is to be located in the vicinity of the computer system, you may use a hardwired connection. The Electrical Industry Association (EIA) Standard RS-232-C limits cable lengths to a maximum of 15 meters (50 feet). The alternate HP Direct Connect Type 422 allows the terminal to be connected to a host computer up to 1200 meters (4000 feet).

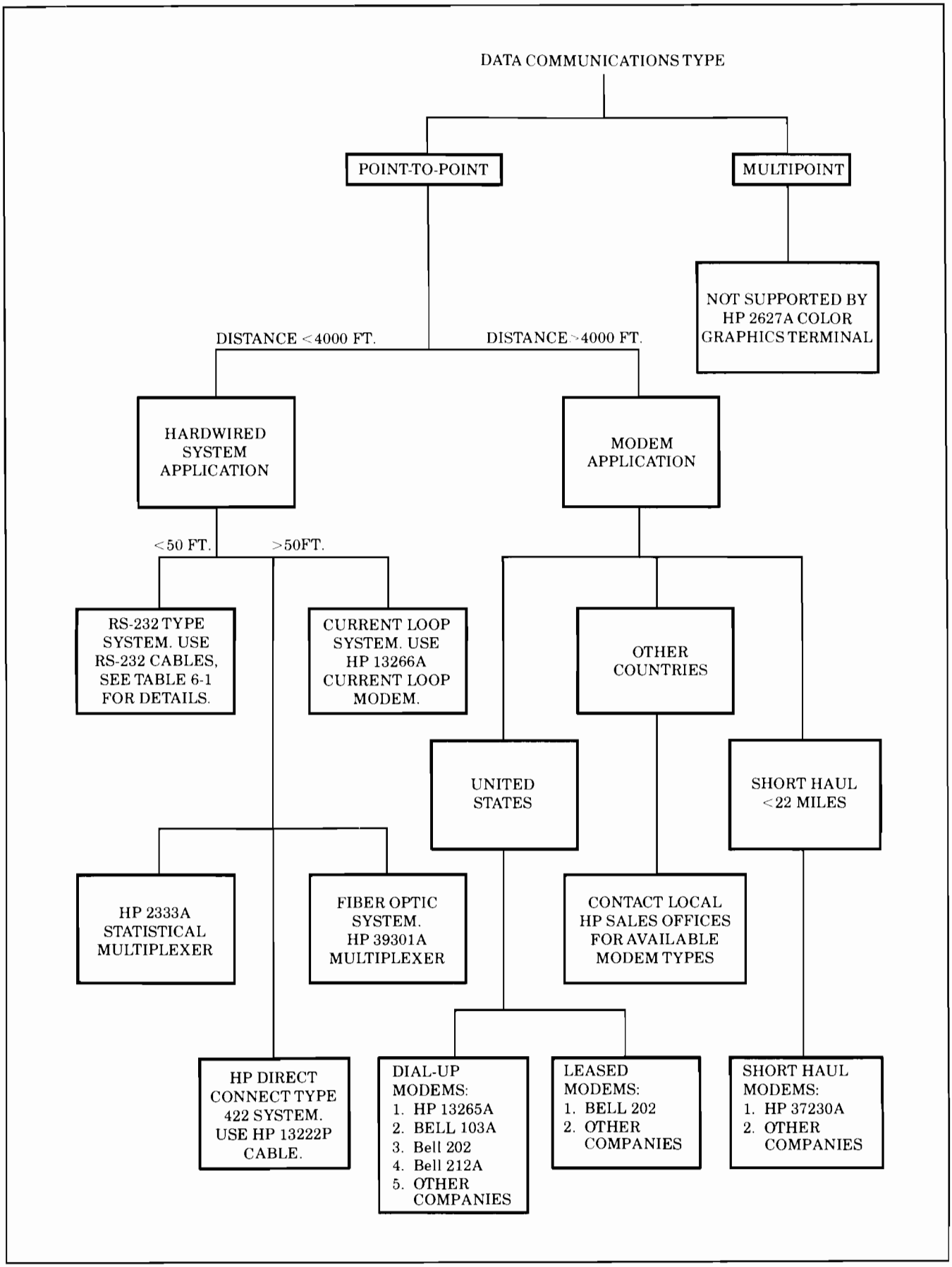


Figure 6-1. Data Communications Decision Tree

Table 6-1. Data Communications Cables

Cable No.	HP Part No.	Description
13222C	13222-60003	TERMINAL TO RS232 CABLE 50 PIN TO 25 PIN CABLE Female RS-232-C 25-pin connector. Length: 2 meters (6.6 feet)
13222M	13222-60002	EUROPEAN MODEM CABLE Male RS-232-C 25-pin connector for interfacing the terminal to the European telephone system via Bell 103 or 202C type European modems. Length: 5 meters (16.7 feet)
13222N	13222-60001	U.S. MODEM CABLE Male RS-232-C 25-pin connector for interfacing the terminal to an HP 1000, 2000, or 3000 Multiplexer; to a Bell 103A, 202C/D/S/T, 212A, or VADIC 3400 modem; or to an acoustic coupler (signal compatible only). Length: 5 meters (16.7 feet)
13222X	13222X	RS-232-C DIRECT CONNECT Male RS-232-C 3-pin connector for interfacing the terminal to an HP 3000 Computer System (64 or 44).
13222W	13222-60007	13222-60007 (W) Female RS-232-C 25-pin connector for interfacing the terminal to an HP 300 Computer System. Length: 5 meters (16.7 feet)
13222P	13222P	HP DIRECT CONNECT TYPE 422 Male HP Direct Connect Type 422 5-pin connector for interfacing the terminal to an HP 3000 Computer System (64 or 44).
13222Y	13222-60005	EMP PROTECT (MALE) Male RS-232-C 25-pin connector for interfacing the terminal to an HP 1000, 2000, or 3000 Multiplexer. Provides protection from lightning-induced transients. For use in hardwired configurations only. Length: 5 meters (16.7 feet)
13232U	5061-2403	Modem bypass cable with a female RS-232-C 25-pin connector on both ends. It crosses the signals so that two terminals can communicate with one another. Length: 1.5 meters (5 feet)

NOTE: Information on data communications test connectors is given in Section IX, "Error Messages and Self-Test."

Another consideration is the desired availability of the particular computer port. If you wish to have it available (at different times) to terminals in diverse and/or varying locations, then you should choose a modem connection with dial-up capability. In remote locations, where distances are greater than 4000 feet, you can select dial-up, leased, or short haul modems. Where distances are between 50 and 4000 feet, or if the system environment is electrically "noisy", an HP Direct Connect Type 422, a fiber optic multiplexer, or a current loop modem should be considered.

If the system consists of many terminals at a semi-remote site, or a cabling limitation is present, then a fiber optic or statistical multiplexer should be considered (figure 6-1).

Hardwired Connections

If you decide on a point-to-point hardwired connection, you must decide on the type of cable to be used, either RS-232-C or HP Direct Connect Type 422. The available cables are summarized in Table 6-1. Pin assignments for the cables are given in the *HP Cabling Manual*, part no. 5952-2047. Please note that a hardwired connection for your terminal is always full duplex (the terminal does not support half-duplex connections).

Modem Connections

If you decide on a point-to-point modem connection, you must now decide what type of modem to use. Note that

point-to-point as supported by the terminal always employs asynchronous transmission. You will therefore be limiting your choice of modem to the asynchronous variety. Refer to table 6-2 for help in selecting the proper modem.

Modem Considerations

If you are communicating with the host computer through a modem only with RS-232-C, it may be necessary for you to turn on a modem power switch or make modem parity setting changes. The modem's baud rate and parity settings should be the same as those configured in the terminal.

The terminal supports the modems listed in table 6-2. Modem connections are not supported by HP Direct Connect Type 422.

Whenever the line selected in the configuration menu is active, an asterisk appears between the fourth and fifth screen label at the bottom of the screen (see terminal configuration menu). If your facility requires the display of this "active modem" indicator, do not shut off the screen labels display.

The asterisk between the fourth and fifth screen label is controlled by an LED which tracks the Data Set Ready (DSR) input line to the terminal. When a modem is connected, the DSR line is low (active) and the modem indicator (asterisk) is on. When the DSR line is high, this signals a modem disconnect and the asterisk disappears from the screen.

INSTALLING A POINT-TO-POINT CONFIGURATION

The terminal's datacomm port may be connected to a computer via a 50-pin, female RS-232-C or HP Direct Connect Type 422 compatible connector provided on the back of the terminal (see figure 6-2).

CABLING

The HP 13222 cables listed in table 6-2 all have a male 50-pin connector on one end and either a male or female RS-232-C or HP Direct Connect Type 422 connector on the other. The 50-pin end is the wider of the two (approximately 7 cm or 2¾ inches wide) and you attach it to the connector on the rear panel of the terminal. The RS-232-C end attaches to the modem, computer multiplexer panel, external printer, or interface cable as illustrated in figure 6-3. The HP Direct Connect Type 422 can only be connected directly to a computer.

Cabling kits are available from Hewlett-Packard for customers desiring to make their own cables for custom installations. For details on how to order these kits, contact your local HP Sales Office.

You may also connect either an HP 13265A Modem or an HP 13266A Current Loop Converter to the connector as illustrated in figure 6-4.

NOTE: Cabling information for the external peripheral port and the optional video interface connectors is given in Section VII, "External Devices."

Table 6-2. Modems

MODEM	Data Rate (Bits/Sec)	Duplex Full/Half	Dialed/ Leased	Reverse Channel
HP 13265A (see note 3)	300	F	D	No
Bell 103A	300	F/H	D/L	No
Bell 202T Bell 202D	1200 (see note 2)	F/H	D/L	Option (see note 4)
Bell 212A (see note 1)	300 or 1200	F	D	No
Vadic VA3400 (see note 1)	1200	F	D	No

NOTES:
1. Can be configured for either asynchronous or synchronous operation. With the HP 2627A, however, it must be configured for asynchronous operation.
2. C2 line conditioning allows operation at 1800 bits per second.
3. HP 13265A is Bell 103A compatible.
4. Not supported on the HP 2627A.

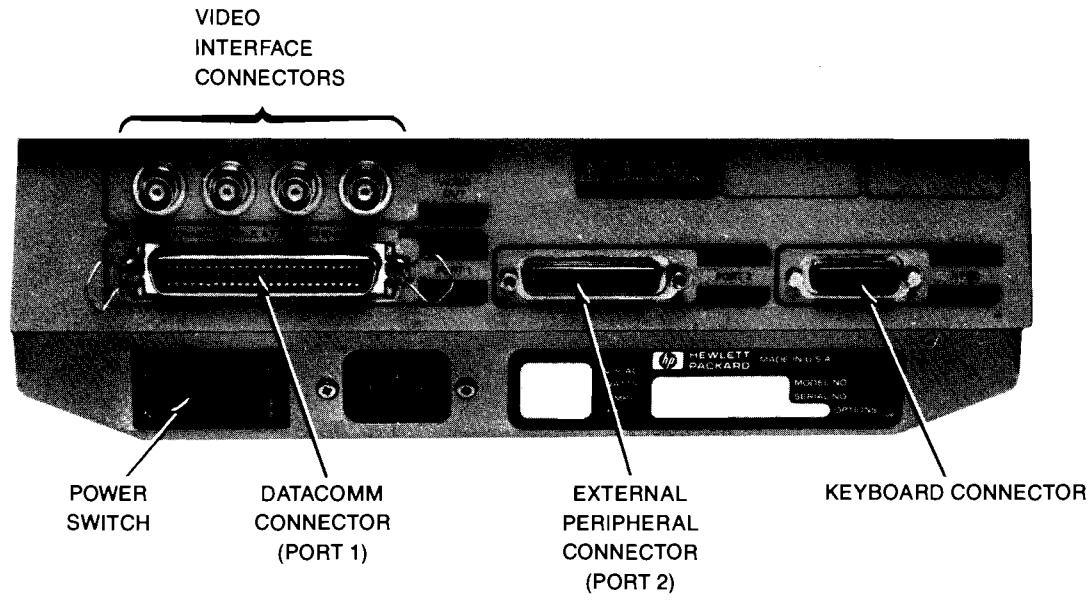
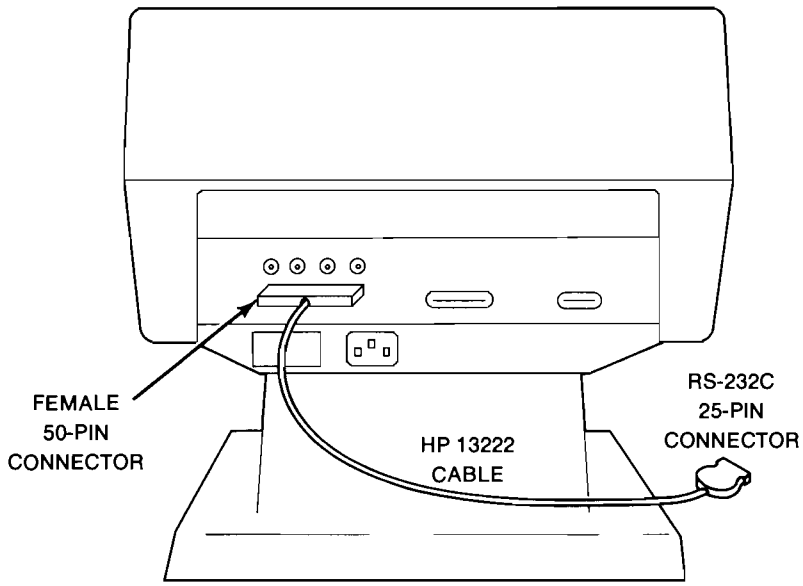


Figure 6-2. HP 2627A Display Terminal, Rear View



APPLICATIONS:

1. Interface cable to host computer.
2. Multiplexer panel on host computer
3. Modem
4. Modem bypass cable (HP 13232U)

Figure 6-3. Terminal Cabling (HP 13222 Cables)

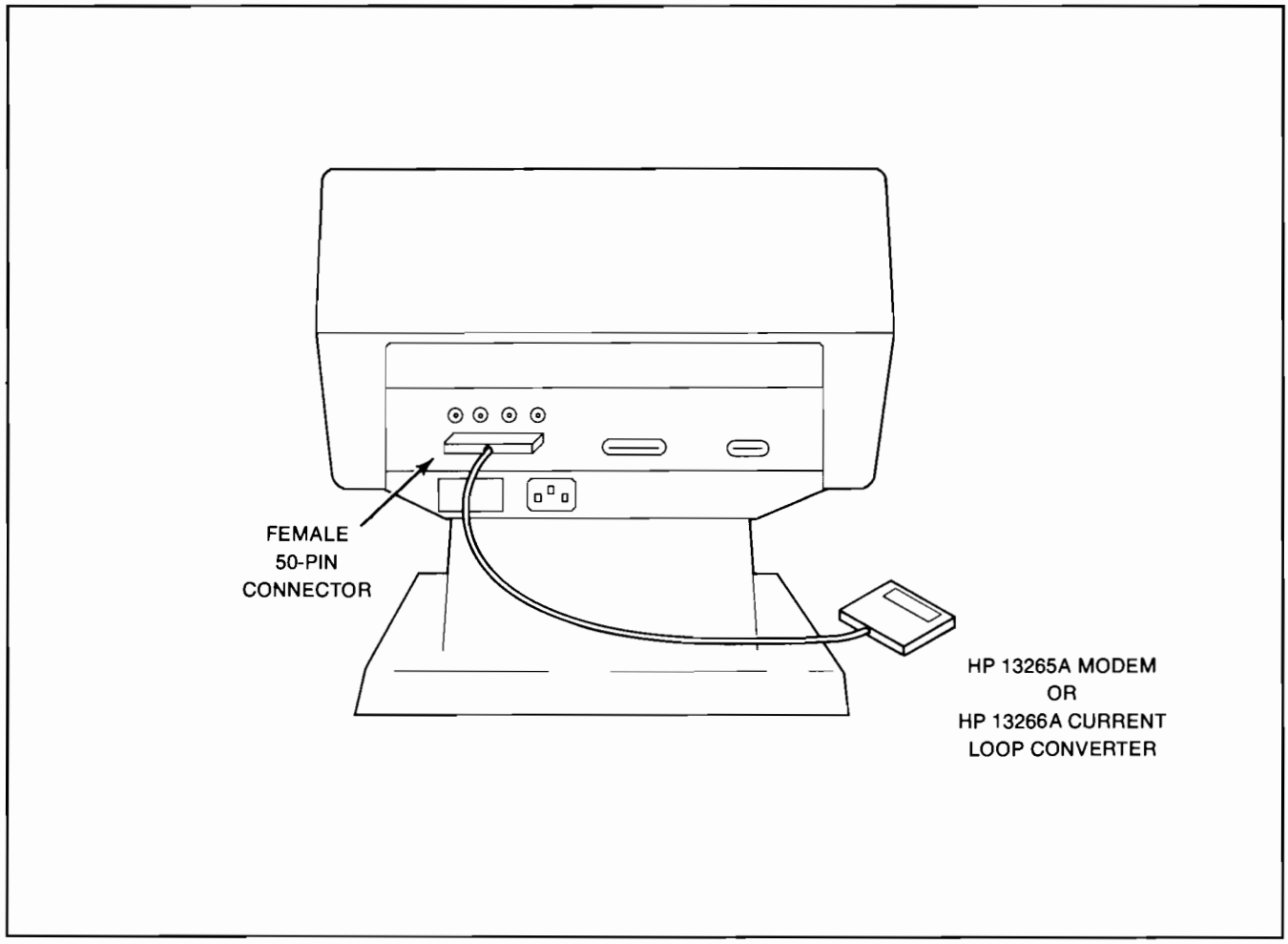


Figure 6-4. Terminal Cabling (HP 13265A Modem or HP 13266A Current Loop Converter)

DATA COMMUNICATIONS CONFIGURATION

Once the physical connections between the terminal and the computer or modem are complete, the terminal can be configured.

To configure the datacommunications portion of the terminal, first use the following keystroke sequence:

	[f8]		[f3]
[AIDS],	config keys	,	datacomm config

When you press the datacomm config (f3) function key, the datacomm configuration menu currently stored in non-volatile memory appears on the screen and the function key labels change to the following:

[f1]	[f2]	[f3]	[f4]
SAVE CONFIG	NEXT CHOICE	PREVIOUS CHOICE	DEFAULT VALUES
[f5]	[f6]	[f7]	[f8]
		DISPLY FUNCTN*	config keys

Note that if you have not previously stored a menu in non-volatile memory, the default values are displayed in the configuration menu (see figure 6-5).

The menu contains a set of unprotected fields that you access using the tab key. You select the desired parameters in these fields using the NEXT CHOICE (f2) and PREVIOUS CHOICE (f3) function keys.

The meanings of the various fields in datacomm menu are described in table 6-3.

When you have set all the fields to the desired values, you may then save them in non-volatile memory using the SAVE CONFIG (f1) function key. Note that when you do this, the particular datacomm configuration takes effect immediately.

While the datacomm configuration menu is displayed on the screen, the f4 and f8 keys have the effects described below:

[f4]	
DEFAULT VALUES	Pressing this key causes all fields in the menu to be filled with their default values.

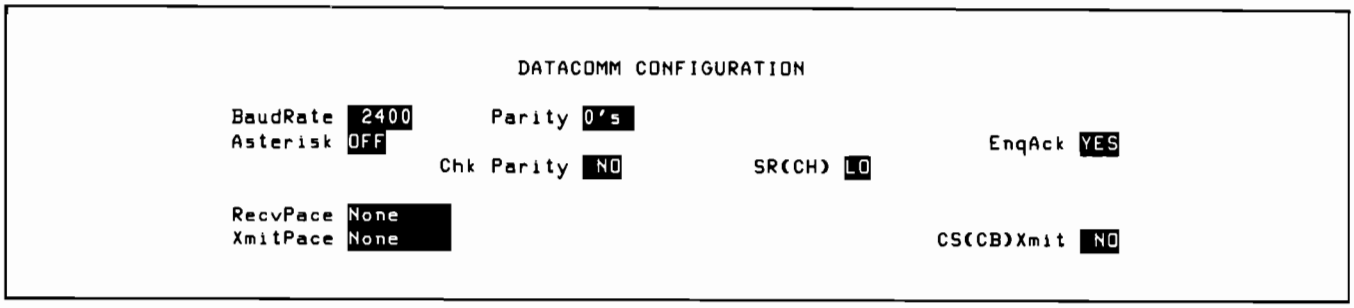


Figure 6-5. Datacomm Configuration Menu

Table 6-3. Datacomm Configuration Menu Fields

BaudRate	This field specifies at what speed you want the data transmission to take place (in bits per second). Values: 110 600 4800 134.5 1200 9600 150 1800 300 2400 (default) NOTE For 110 baud, the terminal is automatically configured to transmit 2 stop bits with the data to the computer. At 110 baud, the terminal also expects to receive 2 stop bits with the data received from the computer. For all other baud rates, 1 stop bit is tranmitted with data and expected to be received with data.
Parity	This field specifies what type of parity generation and checking you wish used with each data character. Values: NONE (no parity bit) 0'S (parity bit always zero) (default) ODD (odd parity) 1'S (parity bit always one) EVEN (even parity)
EnqAck	This field enables or disables the use of the Hewlett-Packard ENQ-ACK handshake. This type of handshaking is described under "Pacing Mechanisms" in the "Point-to-Point Programming Information" portion of this section. Values: YES (enable) (default) NO (disable)
Asterisk	This field indicates whether the selected datacomm line (CS, DM, or RR) is active or inactive (OFF). Typically, OFF is used for hard-wired applications; CS for US modems; DM for European modems; and RR for short haul modems. Values: OFF (default) CS (clear to send) DM (data mode) RR (receive ready)
Chk Parity	This field is used for enabling or disabling the parity check feature for data characters received over the datacomm line. Note that if the Parity field (above) is set to NONE, then this field is ignored. Values: YES (enable) NO (disable) (default)
SR(CH)	This field specifies the desired state of the RS-232-C SR line when the terminal's power is first turned on or when the terminal is reset. The SR line, RS-232-C pin number 23, is defined as the Data Signal Rate Selector (DTE Source). It is normally used on dual speed modems (Bell 212A) to select the appropriate speed (single speed modems merely ignore this line). Values: HI LO (default)

Table 6-3. Datacomm Configuration Menu Fields (Continued)

RecvPace	Receive pacing is a mechanism by which the terminal automatically controls (halts and resumes) the transmission of data from the remote device. There is one way of performing receive pacing: by using the XON and XOFF control codes. If this field is set to "XonXoff", the terminal will automatically perform receive pacing using XON (ASCII D_1) and XOFF (ASCII D_3) control codes. With this type of receive pacing, the terminal causes the remote device to halt transmission by sending an XOFF code and to resume transmission by sending an XON code. For this type of receive pacing to work, the remote device must of course be configured to start and stop transmission in response to XON and XOFF codes. Note that if the remote device recognizes XON and XOFF codes and your terminal is operating in character mode, you can issue the codes through the keyboard regardless of the setting of this field. The CTRL and Q keys (when pressed simultaneously) generate an XON code and the CTRL and S keys generate XOFF. Values: NONE (default) XON/XOFF
CS(CB)Xmit	This field specifies whether or not a true state (−12V) on the RS-232-C Clear to Send (CS/CB) control line is a required condition for transmitting data. For a modem configuration, it is recommended that you set this field to "YES". Values: YES NO (default)
XmitPace	Transmit pacing is a mechanism by which the remote device can control (stop and resume) the transmission of data from the terminal. If enabled, transmit pacing is performed using XON and XOFF control codes. When the terminal receives an XOFF code (ASCII D_3), it stops transmitting data. When the terminal subsequently receives an XON code (ASCII D_1), it resumes transmitting data. This should not be used with D_1/D_3 handshaking. Therefore, if XmitPace is enabled, InhHndShk (G) and Inh D_3 (H) in terminal configuration must both be set to "YES", which disables D_1/D_3 handshaking. If this field is set to "NONE", the terminal does NOT recognize the ASCII D_1 and D_3 codes as XON and XOFF. For another form of transmit pacing, refer to the description of the CS(CB)Xmit field above. Values: NONE XON/XOFF

[f8]
config keys

Pressing this key removes the menu from the screen (WITHOUT activating it or saving it in non-volatile memory) and returns the function key labels to the following:

POINT-TO-POINT PROGRAMMING INFORMATION

This topic discusses programming information of interest to someone who is writing a data communications driver or controller program to communicate with the terminal in an asynchronous point-to-point environment.

An asynchronous point-to-point data communications environment is characterized by a flow of characters that have been produced over random time intervals. In order to achieve hardware synchronization, each character is delimited by a "start bit" and one or more "stop bits".

Start And Stop Bits

These hardware-generated bits are used for synchronizing the transmit and receive devices in an asynchronous environment. A start bit is a "zero" line state (+12V) that lasts for 1.0 bit time; it is affixed to the beginning of a serial character bit stream (which may also include a parity bit). A stop bit is a mark or a "one" line state (−12V) that lasts for 1.0 bit time; it is appended to the end of each serial character bit stream. After the stop bit, the line remains in the mark state until the next character, signified by a start bit, is transmitted.

The start and stop bits are not configurable. For 110 baud, the terminal is automatically configured to transmit 2 stop

bits with the data to the computer. At 110 baud, the terminal also expects to receive 2 stop bits with the data received from the computer. For all other baud rates, 1 stop bit is transmitted with data and 1 stop bit is expected to be received with data.

Parity Checking

In an asynchronous point-to-point environment, the terminal provides a vertical redundancy check (VRC), which is a character-based error checking mechanism for non-binary data. With VRC, an additional bit is affixed to each character to provide an expected high-order bit state for each character. This type of parity generation and checking is a means of determining the validity of data transfer on a character-by-character basis.

Note that when 8-bit data is being exchanged, parity cannot be used and the "Parity" field in the datacomm configuration menu must be set to "NONE". Otherwise, one of the bits will be mistaken for parity.

The terminal offers the following four types of parity:

1. 0'S. The high-order bit is always a zero.
2. 1'S. The high-order bit is always a one.
3. ODD. The high-order bit is set to a zero or a one, whichever produces an odd number of one bits in the overall character representation (the seven data bits plus the eighth parity bit).
4. EVEN. The high-order bit is set to a zero or a one, whichever produces an even number of one bits in the overall character representation (the seven data bits plus the eighth parity bit).
5. NONE. Eight bits of data are transmitted and received. No parity bit is transmitted or received.

Receive Buffer

The terminal's receive buffer is a first in/first out (FIFO) storage area for accepting data from the remote device. When you are using any type of receive pacing, this buffer is partitioned into a working buffer and a 40-byte overrun area. For example, the specified buffer size is always 256 bytes, thus if receive pacing is being used, the working buffer is 216 bytes long and the overrun area is 40 bytes long. When the data being received exceeds the working buffer and intrudes on the overrun area, the terminal will exercise its receive pacing mechanism (send an XOFF, for example, if XON/XOFF receive pacing is enabled) at that time to temporarily halt the flow of data from the remote device. When enough data has been processed so that the receive buffer is only half full, the terminal then signals the remote device to resume transmission (by sending an XON, for example, if XON/XOFF receive pacing is enabled).

There is no equivalent overrun area for transmitting data from the terminal to the remote device.

Receive Errors

When receiving data from the remote device, the terminal can detect the following three types of error conditions (in addition to parity errors):

1. Character overruns—a character is received before the preceding character was processed by the terminal's datacomm firmware.
2. Framing errors—no stop bit was detected at the end of a character.
3. Buffer overflows—the entire allocated buffer space is filled (both the working buffer and the overrun area).

Receive errors, when detected, are reported to the remote device by way of byte 5 of the primary terminal status bytes. The remote device will not be able to determine which type of error occurred. If multiple receive errors occur simultaneously, only one will be reported.

When a datacomm receive error occurs, a delete (del) character (■) is placed in the datacomm queue and later it is displayed on the terminal screen.

NOTE

Because null and del characters are automatically stripped from datacomm, the only del characters appearing on the screen from datacomm are the result of datacomm errors.

Local/Remote Modes

The data communications portion of the terminal operates independently whether the terminal is in local or remote mode. If the terminal is switched from remote to local while data is being received from the remote device, the datacomm portion of the terminal continues receiving data (it does NOT halt the transmission). In such a case, the data received while the terminal is in local is discarded by the terminal's maincode firmware.

Full-Duplex Operation

In a full-duplex environment, the terminal is capable of transmitting and receiving data simultaneously. The ability to transmit may be inhibited temporarily, but it is never exclusive of the ability to receive. Two physical sets of data lines are required; control lines are needed only when hardware handshaking or a modem is used. Transitions on the control lines have no effect on the actual transmit/receive state of the terminal.

When the terminal is connected to the host computer via a modem, the following primary control lines are required:

Request to Send (RS/CA)

Clear to Send (CS/CB)

Data Terminal Ready (CD/TR)

If the terminal is hardwired directly to an HP 3000 computer system (no modem), only Transmit Data (SD/BA), Receive Data (RD/BB), and Signal Ground (AB) are required.

The HP Direct Connect Type 422 transmits data on SD(A), SD(B) and receives data on RD(A), RD(B) using two twisted pairs.

Pacing Mechanisms

In a full-duplex environment, the terminal can participate in either of the following forms of transmit pacing:

1. Hardware handshake. The host computer can temporarily restrain the terminal from transmitting by lowering the Clear to Send (CB) line. Note that this type of transmit pacing can only be used in a hardwired configuration where the Clear to Send (CB) line exists in the cabling.
2. XON-XOFF handshake. The host computer or external printer uses the ASCII control codes XON (D_1) and XOFF (D_3) to start and stop the terminal from transmitting. Note that a single XON code cancels any number of preceding XOFF codes.

In a full-duplex environment, the terminal can also participate in the XON/XOFF handshake form of receive pacing, in which the terminal uses the ASCII control codes XON (D_1) and XOFF (D_3) to start and stop the host computer from transmitting. Note that a single XON code cancels any number of XOFF codes.

The terminal can also participate in an ENQ/ACK handshake (which is a Hewlett-Packard handshaking mechanism). With this form of handshaking, the host computer transmits a block of data and then sends an ASCII <ENQ> control code. The terminal responds to the <ENQ> by sending back an ASCII <ACK> control code when it has processed all of the data preceding the <ENQ>. The general interpretation of these two control codes is as follows:

ENQ: "Have you processed the data up to this point?"
ACK: "Yes, I have."

There are some cases where the terminal may take quite some time before responding to the <ENQ>. Also, if the <ENQ> was not received properly by the terminal, the terminal will never respond with the <ACK>. To recover in these cases, the HP 3000 Computer System sets a timer whenever it sends an <ENQ>. The timer is about 10 seconds. If the HP 3000 receives an <ACK>, the timer is cleared and the next block of data (80 characters) is sent. However, if the timer ticks off (times out), the HP 3000 will assume that the terminal has received the data and processed it, and will still send the next block of data. This may cause problems for the terminal, which may not have the buffer space available to receive the data. Even if the terminal can receive all of the data, when it finally is able to send the <ACK> to the first <ENQ> (the one that timed out), the HP 3000 will interpret that as the < ACK> to a subsequent <ENQ> (if one has been issued). These problems do not normally occur, because the terminal can usually process data, including most escape sequences, fast enough to respond with an <ACK> before the HP 3000 times out.

NOTE

If the HP 3000 does not receive the <ACK> properly from the terminal, it will eventually time out and recover.

Some polygonal area filling requires more time to process the data. To ensure that data is not lost when doing extensive area filling, select the "XON/OFF" value for the "RecvPace" field in the datacomm menu (see figure 6-5 and table 6-3).

The above pacing mechanisms are responded to by the terminal in the following order of precedence:

1. Hardware handshaking pacing (highest priority)
2. XON/XOFF transmit pacing
3. XON/XOFF receive pacing
4. ENQ/ACK pacing (lowest priority)

External Devices

SECTION VII

INTRODUCTION

The HP 2627A Color Graphics Terminal includes an asynchronous RS-232C port for connecting the terminal to an external printer. As an option, a video interface is available for output to monitors and cameras.

With an external printer present, you may do any of the following:

- Initiate and terminate print operations programatically from a host computer.
- Print the line containing the cursor.
- Print all lines from the one containing the cursor through the bottom line on the screen.
- Print all lines from the one containing the cursor through the end of display memory .
- Print all of display memory.
- Copy a configuration menu from the screen to an external printer.
- Enable data logging (to occur either from the top or bottom of display memory, as designated by you when you enable it).
- Perform a line feed (advance the paper one line).
- Perform a form feed.
- Print the contents of graphics memory.

All of the above printer control functions can be initiated either locally by operator keystrokes or remotely by escape sequences sent from a host computer.

SELECTING PRINTER MODES

To enable or disable the various printer modes (record mode, or data logging), you must get to the "device modes" set of system function keys doing the following keystroke sequence:

This changes the function key labels to the following:

The use of f2 - f4 "device modes" keys are described in the next few topics below.

To copy graphics data from graphics memory to the external printer, you may either initiate the transfer from the graphics keypad or by an escape sequence. Pressing the GRAPH COPY key on the graphics/numeric pad initiates the transfer. Copying graphics data under program control (using the Ec&p sequence) is discussed later.

Record Mode

Record Mode copies data from the display or datacomm to the selected "to" device(s), depending upon whether or not the terminal is in Remote Mode.

- If in local mode (not Remote), the contents of display memory (alphanumeric display memory in the HP 2627A) is copied to the selected "to" device.
- If in Remote Mode, the data stream on the datacomm line is sent directly to the selected "to" device.

Record Mode may be initiated from either the keyboard or from an Ec&p escape sequence. To initiate record mode from the keyboard, press

An asterisk will appear in the softkey label to indicate that record mode is enabled. While in record mode, the keyboard is disabled except for the BREAK, RESET, and "RECORD MODE" keys. Pressing RESET (soft reset), or SHIFT, CTRL, RESET (hard reset), or "RECORD MODE" softkey will terminate record mode.

The "to" device(s) must be selected before turning on the record mode. To programmatically select the "to" device(s), use one of following sequences:

Ec&p3D (selects display)
or
Ec&p4D (selects external printer)

To initiate record mode from an escape sequence, send:

Turn on Record Mode: Esc&p <NUM>p 20C
(<NUM>p is optional)

The optional <NUM> parameter defines the character which may be used to programmatically turn off record mode. <NUM> is the decimal equivalent of an ASCII character that will turn off record mode if it is the first character in a record. The default is "0". If "<NUM>p" is omitted, or if "0p" is specified, no character will terminate record mode. Termination can only occur by pressing the "RECORD MODE" softkey, pressing RESET or sending Esc g (a soft reset), or pressing CTRL SHIFT RESET or sending Esc E (a hard reset) programmatically.

NOTE

If the keyboard is locked when record mode is on, an escape sequence (Esc) cannot be typed on the keyboard.

The termination character is valid only for the current activation of record mode (i.e., when record mode is terminated, the termination character returns to "0", the default character).

Selecting the "to" device(s) and turning on the record mode may be combined in one sequence. For example, select the external printer and turn on the record mode:

Esc&p4d20C

The terminal returns an "S", or "F" to the host computer if the escape sequence is received from datacomm. An "S" means that the terminal executed the escape sequence successfully; an "F" means that the terminal failed to execute the escape sequence.

When the status character is sent depends upon whether or not DC1/DC2 handshake is enabled. (Refer to the Terminal Configuration Menu discussion in Section II for handshake types.) If DC1/DC2 handshake is disabled, the character is sent immediately after the escape sequence is received from the host computer. If the handshake is enabled, the character is sent after record mode is turned off and a DC1 is received from the host computer. For any status transfer, the keyboard is locked until a DC1 is received.

A 256-character buffer is used to hold each record prior to sending it to the specified "to" device(s). If the record exceeds 256 characters, the terminal's handshake holds off any further transmission from the host until the buffer's contents is sent to the "to" device(s). Records shorter than 256 characters are indicated by a LF (linefeed) character. Again, the terminal's handshake holds off any further transmission from the host until the record in the buffer is sent to the "to" device(s). A CR (carriage return) follows every record sent to the "to" device(s).

If record mode is turned off, the contents of a partially filled buffer will be sent to the "to" device(s).

If the record mode termination character is the first character into the buffer, record mode is terminated; the termination character is not sent to the "to" device(s).

DATA LOGGING

The terminal includes a mechanism called "data logging" whereby data can be automatically routed to an external printer. There are two types of data logging: top and bottom.

Top Logging

When the display is filled and another line of data is entered through the keyboard or received over a datacomm line, the top line in the display is purged to make room for the new line. With top logging, each line that is purged from the top of the display is printed. Thus, while the line is "lost" from display memory, it is maintained in hard copy form.

Bottom Logging

With bottom logging, each time the cursor moves from one line to another as the result of an explicit line feed or an end-of-line-wraparound, the line from which the cursor moved is printed. This feature allows you to maintain a hard copy "trail" of all lines added to the display in the order in which they were entered and/or received.

When performing data logging in remote mode, the terminal and host computer must be using the ENQ-ACK or XON-XOFF handshakes or they must be using a baud rate that is equal to or less than the rate at which the slowest device (terminal or external printer) can function without handshaking.

From the keyboard, you enable and disable data logging using the LOG TOP (f4) and LOG BOTTOM (f3) keys. These keys alternately enable and disable top logging and bottom logging, respectively. When either is enabled, an asterisk appears in the associated key display.

From a program executing in a host computer, you enable and disable data logging using the following escape sequences:

ENABLE BOTTOM LOGGING: Esc&p 11C
ENABLE TOP LOGGING: Esc&p 12C
DISABLE LOGGING: Esc&p 13C

Both forms of data logging may NOT be enabled simultaneously.

Once either form of data logging is enabled, it remains enabled until explicitly disabled, until the other form of data logging is enabled, until a hard reset is performed, or until the power is turned off.

Note that the keyboard is temporarily locked while a line of data is being "logged". This may make it difficult to perform any keyboard operations if a large quantity of data is coming into the display over a datacomm line rapidly enough to result in continuous logging.

DISPLAY TO PRINTER ALPHANUMERIC DATA TRANSFERS

The alpha display is defined as the "from" device in alphanumeric data transfers to the external printer. If you define the external printer as the destination ("to") device, you can use the "device control" set of system function keys to print one or more lines of data from the display. Select the "device control" keys by doing the following keystroke sequence:

This changes the function key labels to the following:

For data transfers initiated through the keyboard, display memory is always the source ("from") device while the external printer is the destination ("to") device. To select an external printer as the "to" device, press the "TO EXT DEV" function key (f2). When a printer is selected as the current "to" device, an asterisk appears in the associated function key label.

Programmatically, you can define the "to" device by using a device control escape sequence (Esc&p):

Select external device: Esc&p 4D

Copy Line

When the printer is selected as the destination device, you can copy the line containing the cursor from the display to the printer. The entire line is copied. Block terminators are ignored. After the line is printed, the cursor moves to the leftmost column in the next lower line (column 0, NOT the left margin). If the cursor is at a line that is beyond the last displayable line, the printer does nothing.

From the keyboard, you copy one line of data using the COPYLINE key (f8) in the device control set of system function keys.

From a program executing in a host computer, you copy one line of data using one of the following escape sequences:

Esc&p B or Esc&p 0B

Copy Page

When the printer is selected as the destination device, you can copy all lines, starting with the line containing the cursor through the last line visible on the screen, to the printer. Block terminators are ignored. After each line is printed the cursor moves to the leftmost column in the next lower line (column 0, NOT the left margin). If the cursor is at a line that is beyond the last displayable line, the printer does nothing.

From the keyboard, you can copy a page of data using the "COPY PAGE" softkey (f7) in the device control set of system function keys.

From a program operating in a host computer, you copy a page of data using one of the following escape sequences:

Esc&p F or Esc&p 0F

Copy All

When the printer is selected as the destination device, you can copy all lines, starting with the line containing the cursor through the last line of display memory, to the printer. Block terminators are ignored. After each line is printed the cursor moves to the leftmost column in the next lower line (column 0, NOT the left margin). If the cursor is at a line that is beyond the last displayable line, the printer does nothing.

From the keyboard, you copy "all" using the COPY ALL (f6) key in the device control set of system function keys.

From a program executing in a host computer, you copy "all" using one of the following escape sequences:

Esc&p M or Esc&p 0M

Copy All of Display Memory

When the printer is selected as the destination ("to") device, you can copy all of display memory to it by using an Esc0 sequence. In response to this sequence, the terminal homes the cursor and then copies the entire content of display memory to the printer.

During the data transfer, block terminators and non-displaying terminators within display memory are ignored.

The Esc0 sequence may be entered through the keyboard, issued from a user-defined function key, or issued from a program executing in a host computer.

When the terminal is in local mode, pressing the ENTER key performs this same function.

Copy Menu

When the printer is selected as the destination ("to") device and a configuration menu or user softkey menu is currently being displayed on the screen, you can copy the menu to the printer by pressing the ENTER key.

Skip Line

When the printer is selected as the destination device, pressing the ADVANCE LINE key (f5) in the device control set of system function keys sends an ASCII CR LF control code sequence to the printer, thus causing the paper to be advanced by one line.

Programmatically, you can cause a line feed on the external printer by using the following device control escape sequence:

Esc&p 1c 4u 1P

The "p" parameter in the above escape sequences specifies how many line feeds you wish performed. To initiate four successive line feeds, for example, merely substitute "4P" for the "1P" sequence. The "4U" parameter causes a line feed on the external printer without affecting the "to" devices.

Skip Page

When the printer is selected as the destination device, pressing the ADVANCE PAGE key (f4) in the device control set of system function keys sends an ASCII FF control code to the printer, thus causing the paper to be advanced to the top of the next page.

Programmatically, you can cause a form feed on the printer by using the following device control escape sequence:

Esc&p 0c 4U

The "0c" parameter causes one form feed to occur. The values 2 through 10 may be used instead of the value "0". These values will also be interpreted to mean, "initiate one form feed".

The "4U" parameter causes the paper to be advanced to the top of the next page on the external printer without affecting the "to" devices.

The control code for a Form Feed (CTRL L) results in a top-of-page operation on the printer.

Device Control Completion Codes

After issuing a copy line, copy page, copy all, copy screen, skip line, or skip page Esc&p sequence, the remote program determines whether or not the operation was successfully performed by executing an INPUT or similar instruction that requests one ASCII character from the terminal. The terminal responds by sending an "S", "F", or "U". An "S" indicates successful completion, an "F" indicates that the operation failed, and a "U" indicates that the terminal operator interrupted the data transfer by pressing RETURN. Note that these completion codes cannot be suppressed by configuration parameters or any other means. They are always transmitted and your programs should include input commands explicitly for accepting them. The keyboard is disabled ("locked") until the status is sent.

Note that in either character or block line mode, the terminal sends a CR (or a CR LF if auto line feed mode is enabled) following the completion code. In block page mode, it sends a block terminator character (as defined in the Terminal Configuration menu described in Section II of this manual).

If a datacomm error occurs during the transmission of the data record, the device control completion code is unpredictable. Datacomm errors are reported by way of the terminal status bytes described in Section VIII.

GRAPHICS MEMORY TO PRINTER DATA TRANSFERS

Graphics memory to printer data transfers may be initiated either by the graphics/numeric keypad or by an escape sequence. The GRAPH COPY key initiates the transfer from the keypad. Note that the graphics functions of the keypad are enabled at power-on, hard reset, or when toggled by the SHIFT NUM keys. The escape sequence to initiate the transfer is:

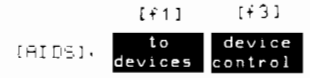
Initiate graphics transfer: Esc 7s F

NOTE

The 7s sets up the source as graphics memory. The default is alphanumeric memory which is selected with Esc&3s.

An external printer may be selected as the destination by the AIDS system softkeys or Esc&p escape sequence. The external printer is configured by the External Device Configuration Menu (discussed later in the section). When selecting the external printer as the destination for graphics data, parity must be set to None in the configuration menu.

To select the external printer by using the softkeys, press



This changes the function key labels to the following:

Press f2 for the external printer. An asterisk appears in the label when a device is selected.

To programmatically select the external printer as the destination, use the escape sequence:

Select external printer as the destination: Esc&p 4D

You may combine device selection and transfer initiation in one escape sequence:

Select external printer as destination and initiate graphics copy: Esc&p 4d F

COMPUTER TO TERMINAL DATA TRANSFERS

When the external printer is selected as a destination ("to") device, you can initiate a data transfer from a program executing in a host computer to the printer by using the following device control escape sequence:

Esc&p <character-count> W <record>

where:

<character-count>	is an integer within the range 1-256 specifying the number of binary bytes in <record>. This is an optional parameter. If present, then the record is terminated when the specified number of binary bytes have been transmitted. If this parameter is not present, ASCII transfers are initiated and the record is terminated when the 256th data byte after the "W" is transmitted or by the first ASCII LF code, whichever occurs first. If the record is terminated by an LF, the LF is also passed to the printer.
<record>	is the data record to be transmitted.

Example: Send the next 15 binary bytes from the computer to all selected "to" devices.

Esc&p 15W

This escape sequence is recognized only when received over a data comm line. It is ignored if entered through the keyboard.

You may include the desired destination device assignment(s) within the escape sequence by using the "d" command parameter. You may also, prior to issuing the above escape sequence, define the desired destination devices either locally through the keyboard or programmatically by way of a separate device control (Esc&p) sequence. In any case, the only destination devices that are recognized by this type of data transfer operation are the alpha display (3d) and an external printer (4d).

If no destination devices are specified within the above escape sequence, the the current "to" device assignments are used. If nothing is currently selected as a "to" device, the data record is accepted over the data comm port and then is discarded by the terminal (also an "F" is returned as the device control completion code).

Binary transfers are of the form Esc&p <character-count> W <record>. ASCII transfers are of the form Esc&p W <record>, where as ASCII LF or the 256th data byte terminates the record. In binary transfers, all eight bits received are passed to the printer. Parity checking and transmission is disabled. In ASCII transfers, seven bits will be passed to the printer.

If the escape sequence does NOT include a <character-count>, then the following applies:

- If EnqAck = YES in the active data communications configuration menu, the data comm firmware strips all <ENQ> codes from the incoming data and responds to each by transmitting an <ACK>.

If the escape sequence includes a <character-count>, then the following apply:

- If EnqAck=YES in the active data communications configuration menu, an <ENQ> code must immediately follow the "W" and precede the data record. It is treated as part of an Enq-Ack handshake (the data comm firmware strips the <ENQ> code from the incoming data and responds to it by sending an <ACK>).
- After the leading Enq-Ack handshake, if required, ALL characters received are treated as data (including <ENQ>, <ACK>, <NULL>, and <DEL>) regardless of the setting of the EnqAck and StripNulDel configuration fields.

When transferring a data record from the host computer to the printer using the above device control escape sequence, the remote program determines whether or not the operation was successfully performed by executing an INPUT or similar instruction that requests one ASCII character from the terminal. The terminal responds by sending an "S" or "F". An "S" indicates successful completion and an "F" indicates that the operation failed. Note that these completion codes cannot be suppressed by configuration parameters or any other means. They are always transmitted and your programs should include input commands explicitly for accepting them. The keyboard is disabled ("locked") until the status is sent.

Note that in either character or block line mode, the terminal sends a CR (or a CR LF if auto line feed mode is enabled) following the completion code; in block page mode, it sends a block terminator character (as defined in the Terminal Configuration menu).

If a data comm error occurs during the transmission of the data record, the device control completion code is unpredictable. Data comm errors are reported by way of the terminal status bytes described in Section VIII of this manual.

CONFIGURING THE EXTERNAL PRINTER

The terminal has an external printer port for interfacing RS-232C serial printers. Configuring an external printer consists of cabling it to the port and specifying parameters in the configuration menu.

Cabling

The terminal has an external device port, along with the data communications port, at the rear of the terminal (see figure 7-1).

The HP 13242 cables listed in table 7-1 both have a male RS-232C connector on one end and either a male or female RS-232C connector on the other. The male end attaches to the external device port on the rear of the terminal, and the other end attaches to the external printer as shown in figure 7-2. For the HP 13242G Cable, which has a male connector on each end, it makes no difference which end is attached to the terminal.

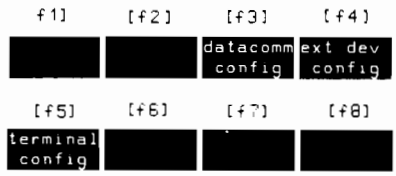
Since the external device connector on the terminal is a standard RS-232C female connector, you can use cables other than those listed in table 7-1 as long as they have a male RS-232C connector on one end and their pin-outs are compatible with those of the HP 13242 cables.

Filling-in the Configuration Menu

Now that you have made the physical connections between the terminal and the external printer, you are ready to configure the terminal's external device port.

To configure the port, first use the following keystroke sequence:

This changes the function key labels to the following:



The `ext dev config` function key, when pressed, causes the external device configuration menu to appear on the screen and redefines the function keys to a set of functions that will assist you in manipulating the various parameters within the menu (see figure 7-3). The configuration menu displays the currently stored menu parameters from non-volatile memory.

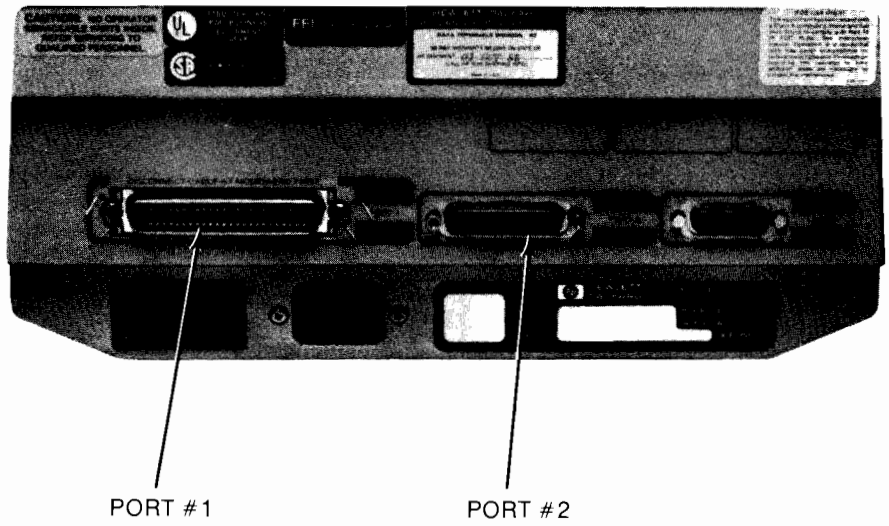


Figure 7-1. HP 2627A Color Graphics Terminal, Rear View

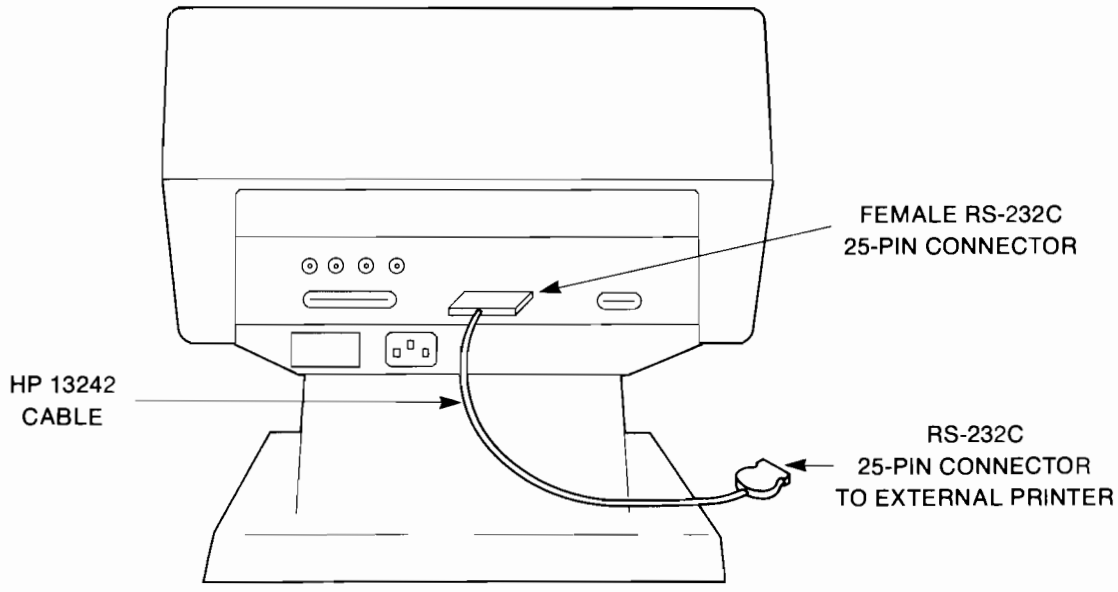


Figure 7-2. External Device Port Cabling (HP 13242 Cables)

Table 7-1. External Device Port Data Communications Cables

Cable No.	HP Part No.	Description
13242G	13242-600010	RS232 PRINTER CBL (MALE) Male RS-232C 25-pin connector for interfacing the terminal to RS-232C compatible printers such as the HP 2631 and HP 2635. Length: 15 feet (4.5 meters)
13242H	13242-600011	RS232 PRINTER CBL (FEMALE) Female RS-232C 25-pin connector for interfacing the terminal to RS-232C compatible printers. Length: 15 feet (4.5 meters)
13242X	13242X	RS232 PRINTER CBL (Direct Connect Type B) Male RS-232C 25-pin connector to 3-pin connector for interfacing the terminal to an external printer.

Whenever a configuration menu is displayed on the screen, the terminal is implicitly in format mode. The menu contains a set of unprotected fields that you access using the tab key. For most of the fields (the ones containing the underlined video enhancement) you select the desired parameters using the "NEXT CHOICE" (f2) and "PREVIOUS CHOICE" (f3) function keys.

The meanings of the various fields are described in table 7-2.

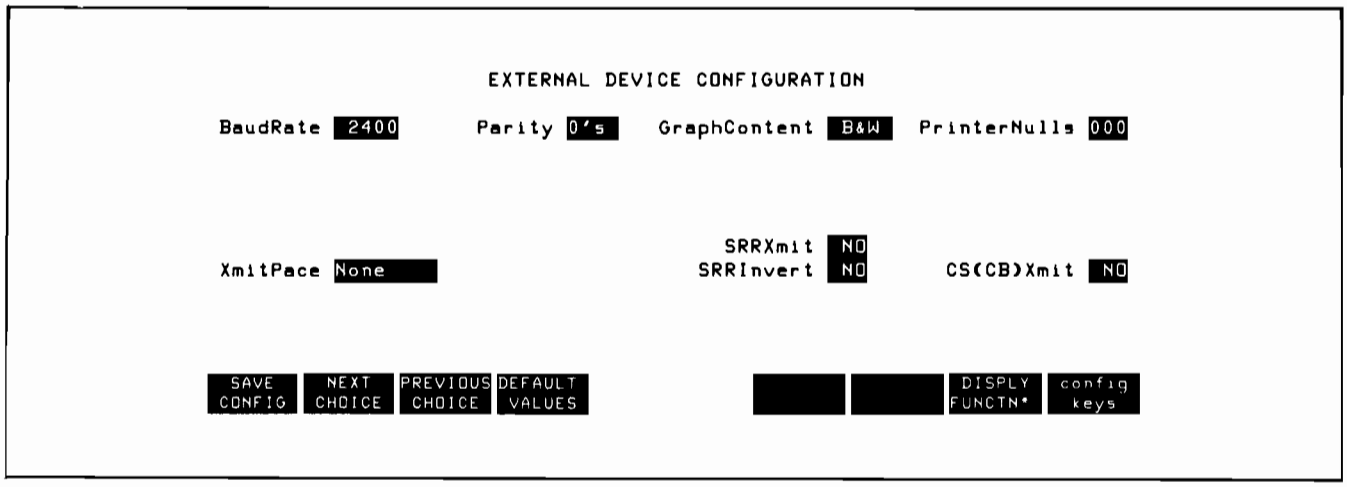


Figure 7-3. External Device Configuration Menu

Table 7-2. External Device Configuration Menu Fields

BaudRate	This field specifies at what speed you want the data transmission to take place (in bits per second). Values: 110, 134.5, 150, 300, 600, 1200, 1800, 2400, 4800, 9600
Parity	This field specifies what type of parity generation and checking you wish used with each data character. (Note that parity must be set to None for binary transfers.) Values: NONE (no parity bit); 0'S (parity bit always zero); ODD (odd parity); 1'S (parity bit always one); EVEN (even parity)

Table 7-2. External Device Configuration Menu Fields (Continued)

Graphics Content	The field specifies whether black and white or color graphics data is transmitted to the external printer. For black and white, all the bit planes are ORed together and then transmitted. For color, data is sent out in HP Raster Standard Format. Values: 0-255 Default: 0
PrinterNulls	This field specifies the number of ASCII null codes (0–255) to be transmitted to an external printer after each ASCII control code. Value: 0-255 Default: 0
XmitPace	Transmit pacing is a mechanism by which the remote device can control (stop and resume) the transmission of data from the terminal. If enabled, transmit pacing is performed using XON and XOFF control codes. When the terminal receives an XOFF code (ASCII <DC3>), it stops transmitting data. When the terminal subsequently receives an XON code (ASCII <DC1>), it resumes transmitting data. This should not be used with DC1/DC2 handshaking. If this field is set to "NONE", the terminal does NOT recognize the ASCII <DC1> and <DC3> codes as XON and XOFF. For other forms of transmit pacing, refer to the descriptions of the SRRXmit and CS(CB)Xmit fields below. Values: NONE Xon/Xoff Default: NONE
SRRXmit	This field specifies whether or not a true state (−12 V) on the RS-232C Secondary Receiver Ready (SRR) or Secondary Carrier Detect (SCF) control line is a required condition for transmitting data. This mechanism is primarily used in conjunction with printers which must be able to control the transmission of data from other devices. The SRR/SCF control line is connected to RS-232C pin number 12. Values: YES NO Default: NO
SRRInvert	This field applies only when the SRRXmit field is set to "YES". When both the SRRXmit and SRRInvert fields are set to "YES", the true state of the RS-232C Secondary Receiver Ready (SRR) or Secondary Carrier Detect (SCF) control line is inverted from −12 V to +12 V. Values: YES NO Default: NO
CS(CB)Xmit	This field specifies whether or not a true state (−12V) on the RS-232-C Clear to Send (CS/CB) control line is a required condition for transmitting data. For a modem configuration, it is set to "YES". Also, if the Asterisk field (in the datacomm configuration menu) is set to "CS", then CS(CB)Xmit should be set to "YES". Values: YES NO Default: NO

VIDEO INTERFACE HARDCOPIES

The optional external video interface provides an output from the display to a video monitor or camera. Figure 7-4 shows the location of the four video interface connectors on the terminal. Procedures for installing and operating the camera or monitor are provided in the appropriate document for the monitor or camera.

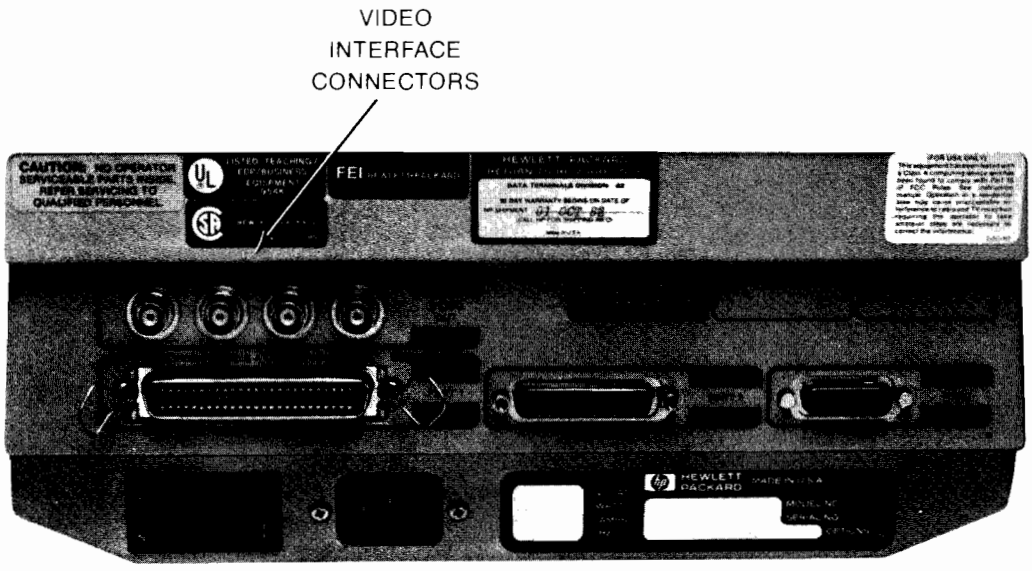


Figure 7-4. Video Interface Connectors

Status

SECTION VIII

INTRODUCTION

This section tells how a program executing in a host computer obtains and interprets status information from the terminal.

Status requests are issued in the form of escape sequences. There are six types of status requests:

1. Terminal ID Status. This request is the means by which your program verifies what kind of terminal it is communicating with.
2. Primary Terminal Status. This request returns seven bytes that report the status of some of the latching keys, various error and pending flags, and the following configuration menu fields (see Section II, table 2-1 for descriptions of the fields):

Xmit Fnctn(A)	Inh Hndshk(G)
SPOW(B)	Inh DC2(H)
Inh Eol Wrp(C)	Esc Xfer(N)
Line/Page(D)	Compat(P,Q)

3. Secondary Terminal Status. This request returns seven bytes that report the status of the memory lock, buffer memory, and I/O firmware.
4. Device Status. This request returns three bytes that report the status of the external device.
5. Graphics Status. There are 12 graphics status requests that can be made; they are listed in table 8-2.
6. Color Pair Status. This request returns the current color method and color pair definition.

The escape sequence used for each of the above requests and the format of the returned status information is presented in the following paragraphs.

All status requests are treated as block transfers. In response to a status request, the terminal transmits an escape sequence, followed by a series of bytes, followed by a terminator. The terminator is as follows:

Character Mode:	$^{C}_{R}$ or $^{C}_{R}$ $^{L}_{F}$
Block Line Mode:	$^{C}_{R}$ or $^{C}_{R}$ $^{L}_{F}$
Block Page Mode:	<BlkTermnator>

In either character mode or block line mode, the $^{C}_{R}$ $^{L}_{F}$ is used if auto line feed mode is enabled. In block page mode, the default block terminator is $^{E}_{S}$. However, <Blk Terminator> is user definable so you may change this setting.

The type of handshaking used is determined by the setting of the InhHndShk and InhDC2 fields of the configuration menu as follows:

1. InhHndShk(G) = YES Inh DC2(H) = YES		No handshake
2. InhHndShk(G) = NO Inh DC2(H) = YES or NO		D_1
3. InhHndShk(G) = YES Inh DC2(H) = NO		$D_1/D_2/D_1$

INTERPRETING STATUS

For primary, secondary, and device status requests, the terminal returns an escape sequence followed by a string of bytes. The status information is contained in the lower four bits of each byte. The upper four bits are set so that the byte translates into one of the 16 ASCII characters shown in table 8-1.

Terminal ID Status

You request the terminal ID status by issuing the following escape sequence:

$^{E}_{c}$ *s1 ^

The terminal responds by sending back the following five-character string:

2627A

Table 8-1. ASCII Status Characters

ASCII CHARACTER	BINARY
0	0011 0000
1	0011 0001
2	0011 0010
3	0011 0011
4	0011 0100
5	0011 0101
6	0011 0110
7	0011 0111
8	0011 1000
9	0011 1001
:	0011 1010
;	0011 1011
<	0011 1100
=	0011 1101
>	0011 1110
?	0011 1111

Terminal Status

Terminal status is made up of 14 status bytes (bytes 0–13) containing information such as display memory size, switch settings, configuration menu settings, and terminal errors. These 14 status bytes are displayed below the self-test screen pattern when the "TERMINAL TEST" (f5) key (in the "service keys" set of function keys) is pressed. There are two terminal status requests: primary and secondary. Each returns a set of 7 status bytes.

PRIMARY TERMINAL STATUS. You request the first set of terminal status bytes (bytes 0–6) by issuing the following escape sequence:

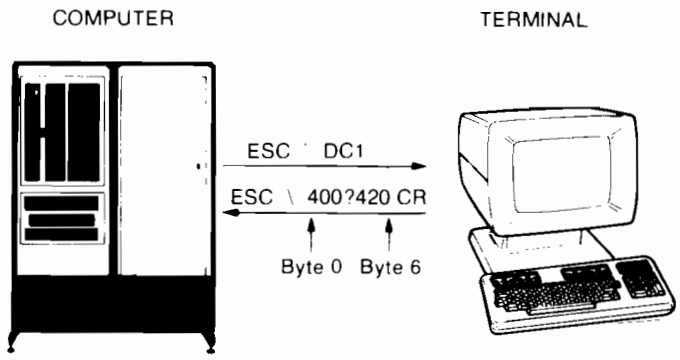
ESC ^

The terminal responds with an ESC \, and seven status bytes followed by a terminator. A typical primary terminal status request and response is illustrated in figure 8-1. The example assumes that the DC1 handshake is being used and that the appropriate terminator is a CR.

SECONDARY TERMINAL STATUS. You request the second set of terminal status bytes (bytes 7–13) by issuing the following escape sequence:

ESC ~

The terminal responds with an ESC |, and seven status bytes followed by a terminator. A typical secondary terminal status request and response is illustrated in figure 8-2. The example assumes that the DC1 handshake is being used and that the appropriate terminator is a CR.



BYTE	ASCII	BINARY	STATUS
0	4	0011 0100	4K bytes of display memory (2 pages of 24 lines)
1	0	0011 0000	Function key transmission disabled Space overwrite latch disabled Cursor wraparound disabled Strapped for line
2	0	0011 0000	Disable handshake Inhibit DC2 disabled
3	?	0011 1111	Upper case only Block mode Auto line feed Terminal sends secondary status
4	4	0011 0100	ENTER key has been hit
5	2	0011 0010	No data comm errors Last Self-Test ok
6	0	0011 0000	No device status pending No device operation status pending

Configuration Straps A–H

Figure 8-1. Primary Terminal Status Example

PRIMARY STATUS BYTES

BYTE 0 DISPLAY MEMORY SIZE

8	7	6	5	4	3	2	1
0	0	1	1	0	1	0	0

- 1K bytes
- 2K bytes
- 4K bytes
- 8K bytes

This byte specifies the amount of display memory available in the terminal. Note that the HP 2627A always returns 4K bytes.

BYTE 1 CONFIGURATION STRAPS A–D

8	7	6	5	4	3	2	1
0	0	1	1	1/0	1/0	1/0	1/0

- Strap D — Page/Line: 1 = Page, 0 = Line
- Strap C — (Inhibit End-of-Line Wraparound): 1 = yes (Enabled), 0 = no (Disabled)
- Strap B — (Space Overwrite Latch): 1 = yes (Enabled), 0 = no (Disabled)
- Strap A — (Function Key Transmission): 1 = yes (Transmitted), 0 = no (Not transmitted)

Refer to Section II for a detailed description of configuration straps A–D.

BYTE 2 CONFIGURATION STRAPS E–H

8	7	6	5	4	3	2	1
0	0	1	1	1/0	1/0	0	0

- Strap H (Inhibit DC2): 1 yes (Enabled), 0 no (Disabled)
- Strap G (DC2 Handshake): 1 yes (Enabled), 0 no (Disabled)
- Strap E
- Strap F

Refer to Section II for a detailed description of configuration straps G and H (straps E and F do not apply to the terminal).

BYTE 3 LATCHING KEYS

8	7	6	5	4	3	2	1
0	0	1	1	1	1/0	1/0	1/0

- Terminal sends secondary status
- AUTO LF Key: 1 = auto LF, 0 = no auto LF
- CAPS LOCK Key: 1 upper case only, 0 upper and lower case
- BLOCK MODE Key: 1 block mode, 0 character mode

BYTE 4 TRANSFER PENDING FLAGS

8	7	6	5	4	3	2	1
0	0	1	1	1/0	1/0	1/0	1/0

- Secondary Status Pending: 1 yes, 0 no
- ENTER Key Pending: 1 yes, 0 no
- Cursor Sense Pending: 1 yes, 0 no
- Function Key Pending: 1 yes, 0 no

BYTE 5 ERROR FLAGS

8	7	6	5	4	3	2	1
0	0	1	1	0	0	1/0	1/0

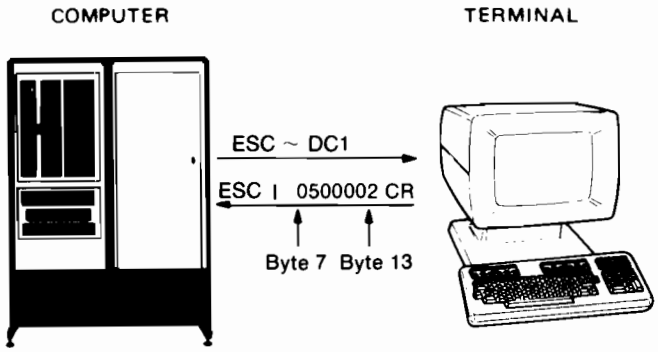
- Data Comm: 1 parity or buffer overflow error or framing overrun error; 0 no error
- Self-Test: 1 no error; 0 error

BYTE 6 DEVICE TRANSFER PENDING FLAGS

8	7	6	5	4	3	2	1
0	0	1	1	0	0	1/0	1/0

- Device Status Pending (Esc&p^): 1 yes, 0 no
- Device Operation Status Pending: 1 yes, 0 no

(tracks "S", "F", or "U" completion codes associated with Esc&p device control sequences.)



BYTE	ASCII	BINARY	STATUS
7	0	0011 0000	
8	5	0011 0101	I/O firmware installed Terminal identifies self
9	0	0011 0000	
10	0	0011 0000	Do not send escape codes to printer Compatibility Mode off
11	0	0011 0000	
12	0	0011 0000	
13	2	0011 0010	Memory locked

Figure 8-2. Secondary Terminal Status Example

SECONDARY STATUS BYTES

BYTE 7 BUFFER MEMORY (always zero)

8	7	6	5	4	3	2	1
0	0	1	1	0	0	0	0

- 1K bytes
- 2K bytes
- 4K bytes
- 8K bytes

Memory installed in addition to display memory that is available for use as data buffers. Note that the HP 2627A terminal always returns a 0 value.

BYTE 8 TERMINAL FIRMWARE CONFIGURATION

8	7	6	5	4	3	2	1
0	0	1	1	0	1	0	1

- 0 = Non-programmable terminal
- 1 = Terminal identifies self
- 1 = I/O firmware installed
- 0 = No APL Firmware

APL Firmware does not apply. Note that the HP 2627A returns a 5.

BYTE 9 CONFIGURATION STRAPS J–M (always zero)

8	7	6	5	4	3	2	1
0	0	1	1	0	0	0	0

- Strap M
- Strap J (Auto Terminate): 1 = yes (Enabled), 0 = no (Disabled)
- Strap K (Clear Terminator): 1 = yes (Enabled), 0 = no (Disabled)
- Strap L (Self-Test Inhibit): 1 = yes (Inhibit test), 0 = no (Allow test)

Straps J–M do not apply to the terminal.

BYTE 10 KEYBOARD INTERFACE KEYS (N–R)

8	7	6	5	4	3	2	1
0	0	1	1	0	1/0	1/0	1/0

Strap R

Q	P	
0	0	Compatibility Mode off
0	1	Scaled Compatibility mode
1	0	Unscaled Compatibility mode
1	1	Compatibility Mode off

- Strap N Printer (Escape Code Transfer): 1 = open (Send ESC code), 0 = closed (Do not send code)
- Strap P Compatibility Mode (Scaled): 1 = Enabled, 0 = Disabled
- Strap Q Compatibility Mode (Unscaled): 1 = Enabled, 0 = Disabled

Strap R does not apply to the HP 2627A terminal

BYTE 11 CONFIGURATION STRAPS S–V (always zero)

8	7	6	5	4	3	2	1
0	0	1	1	0	0	0	0

- Strap V
- Strap S
- Strap U
- Strap T

Straps S–V do not apply to the teminal.

BYTE 12 CONFIGURATION STRAPS W–Z (always zero)

8	7	6	5	4	3	2	1
0	0	1	1	0	0	0	0

- Strap Z
- Strap W (Data Comm Test): 1 = yes (Inhibit), 0 = no (Allow)
- Strap Y
- Strap X

Straps W, X, Y, and Z do not apply to the terminal.

BYTE 13 MEMORY LOCK MODE

8	7	6	5	4	3	2	1
0	0	1	1	0	1/0	1/0	1/0

- 1 = locked in row ≠ 0; 0 = locked in row 0 (memory overflow protect)
- 1 = memory lock on; 0 = memory lock off
- 1 = memory full, 0 = memory not full } only active when memory lock is enabled

DEVICE STATUS

The following information on device status only applies to the external device port on the HP 2627A.

The status of the external printer can be obtained by issuing a device status request. This request would typically be made following a print operation or after examining bytes 5 and 6 of the primary status. The device status bytes are shown in figure 8-3.

You request device status by issuing the following escape sequence:

Esc&p <device code>^

If <device code> is any value other than 4 (external printer), the escape sequence is ignored.

The terminal responds with the sequence Esc\p <device code>, followed by three status bytes followed by a terminator. A typical device status request and response are illustrated in figure 8-4.

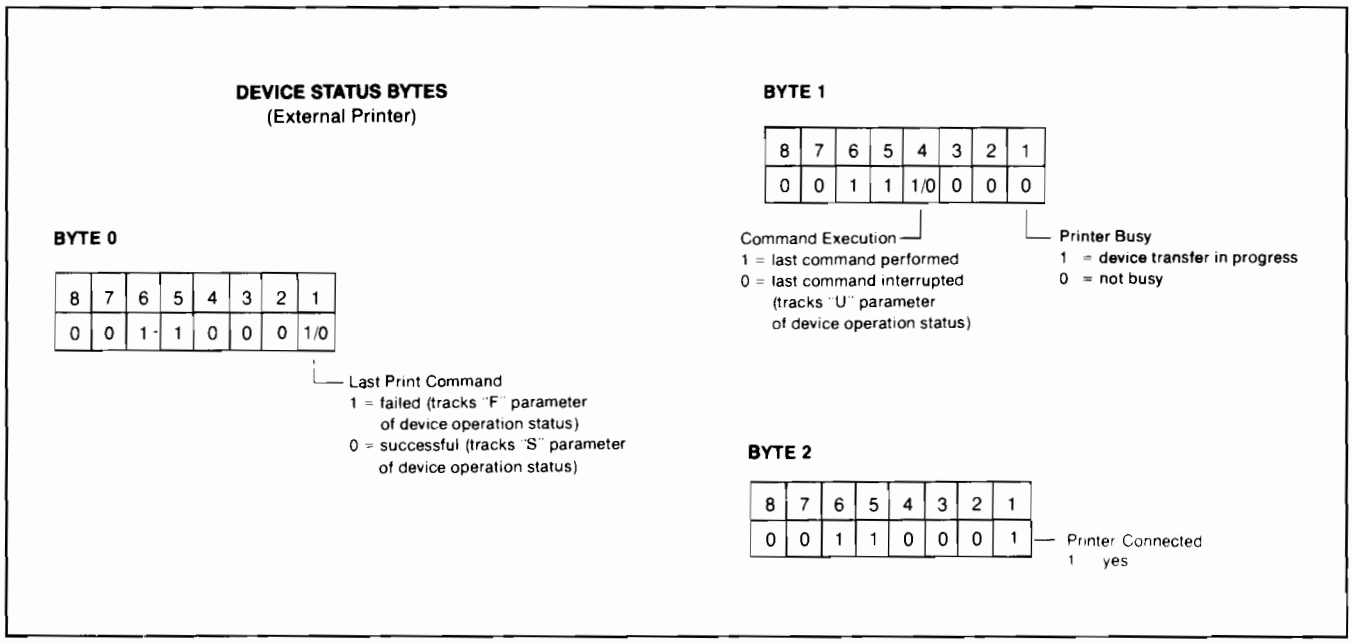


Figure 8-3. Device Status Bytes

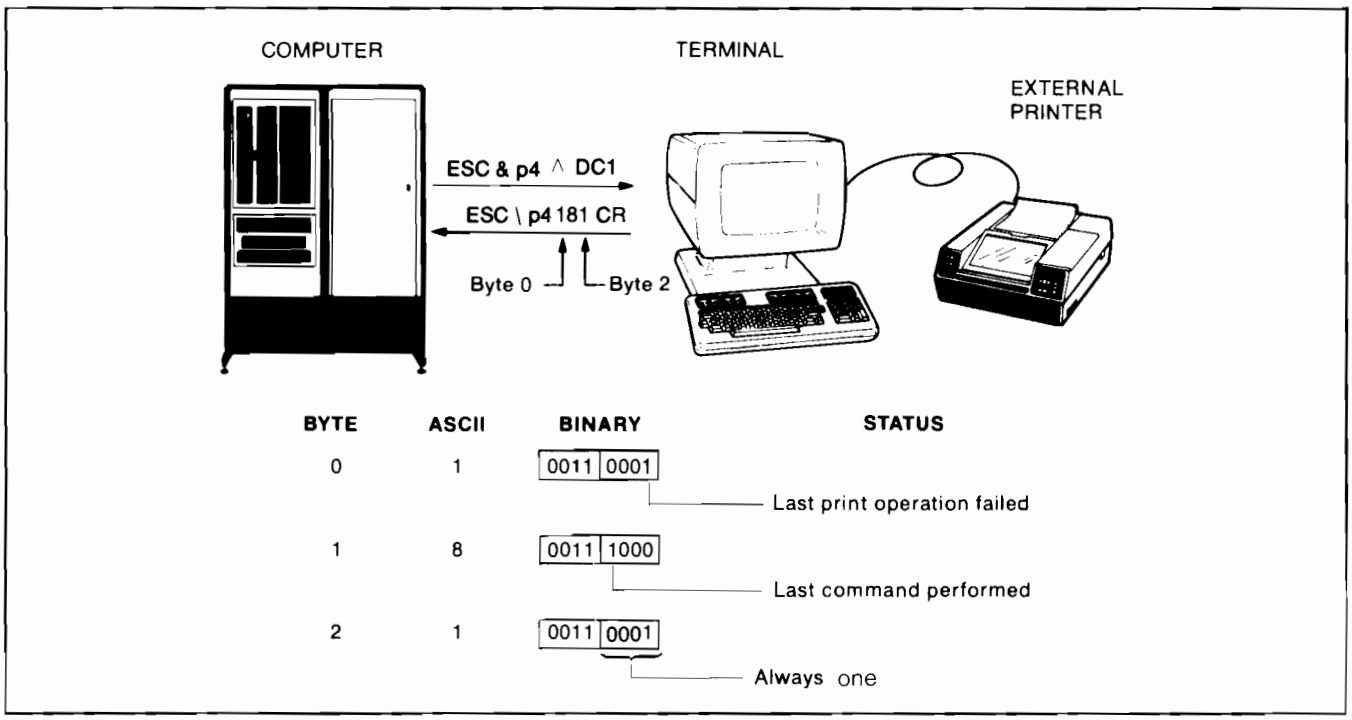


Figure 8-4. Device Status Example

GRAPHICS STATUS

In addition to normal HP 2627A terminal status you can request graphics status information. All graphics status requests are initiated by sending an ESC * s followed by a single parameter (1 through 12) followed by a ^. The single parameter selects the particular status block desired. If an invalid parameter is used, the terminal will simply return its I.D. (see Device I.D. Request, parameter = 1).

Graphics Status Request:

ESC * s <parameter> ^

where ESC * s is the graphics status escape sequence.

<parameter> is 1–12 and selects one of twelve blocks of graphics status information.

The graphics status blocks that can be requested are listed in table 8-2 together with the format of their terminal's response. Detailed descriptions of each of the status requests are contained in the following paragraphs.

The terminal will respond with one or more bytes of status information followed by a block terminator. All status information is returned in ASCII format, separated by commas. Coordinates are returned in a fixed format, consisting of a sign and five digits. Leading zeros are used as required to provide a fixed number of digits (ie. +00100, −01234). This allows you to use simple input statements without the need to mask or shift bits.

If the DC1 handshake protocol is enabled (i.e., "NO" is entered in the "InhHndShk(G)" and "Inh DC2(H)" fields of the Terminal Configuration menu—refer to Section II), the status block is not actually sent until receipt of a DC1 character. If the DC1 character is used, only one graphics status request can be enabled while the terminal is waiting for a DC1. When the DC1 is received, the last graphics status block requested will be sent.

While the terminal is waiting for the DC1, the Device Status Pending bit is set (see byte 6 of the primary status bytes on page 8-3).

The terminal's configuration determines the terminating character sent following the status block (CR, CR-LF, or RS)—refer to page 8-1. Graphics status requests turn on an echo suppress mode in the terminal. This prevents information echoed back from the computer from being displayed on the screen. Once a graphics status block has been sent, characters received by the terminal will not be displayed until one of the following control characters is received: Q, ESC, CR, ETX, ETB, HT, LF, RS, US, Y. With the exception of CR and LF the terminating control code itself will be executed.

The terminal expects the status information to be echoed and uses the terminating control character to turn off the suppress echo mode. If the computer does not echo the status back, a suitable control character must be returned to the terminal to turn off the echo suppress mode.

The graphics status blocks that can be requested are shown in table 8-2.

Read Device I.D. (Parameter=1)

When you request a device I.D. the terminal responds with its Hewlett-Packard model number, 2627A.

Device I.D. Request:

ESC * s 1 ^

The terminal responds: 2627A <terminator>

Read Current Pen Position (Parameter=2)

The pen position and status are returned as a string of ASCII characters.

Table 8-2. HP 2627A Graphics Status Requests

Parameter	Request	Response
1	Read device I.D.	2627A
2	Read current pen position	<X>,<Y>,<PEN>
3	Read graphics cursor position	<X>,<Y>
4	Read graphics cursor position with wait	<X>,<Y>,<KEY>
5	Read display size	<LLX>,<LLY>,<URX>,<URY>,<MMX>,<MMY>
6	Read device capabilities	<b1>,<b2>,. . .<b15>,<b16>
7	Read graphics text status	<X size>,<Y size>,<origin>,<angle>,<slant>
8	Read zoom status	001.,0
9	Read relocatable origin	<X>,<Y>
10	Read Reset status	<RESET>,<b1>. . .<b6>,<b7>
11	Read area shading capability	1,8,8
12	Read dynamics capability	1,1

Pen Position Request: ESC * s 2 ^

The terminal responds: <X>, <Y>,<Pen>,<terminator>

where: <X> = X coordinate
<Y> = Y coordinate
<Pen> = Pen state, 0=pen up, 1=pen down

For example, assume that the pen is at 360, 80, the pen is up,and the terminal is set for the DC1 handshake, with CR as the terminator:

The computer sends: ESC * s 2 ^ <terminator> DC1

X coordinate Pen state

The terminal responds: +00360,+0080,0↵

Y coordinate

Read Graphics Cursor Position (Parameter=3)

The graphics cursor position is returned as a string of ASCII characters.

Read Graphics Cursor Request: ESC * s 3 ^

The terminal responds: <X> = X coordinate
<Y> = Y coordinate

With the cursor positioned in the screen's lower left corner, the terminal's response is:

+00000, +00000↵

Read Cursor Position with Wait (Parameter=4)

This request allows the user to position the cursor, then strike a key to return the position. The ASCII decimal code for the key struck is also returned (not the actual character). The code is returned as three digits. For example, striking an uppercase A would return 065. Only ASCII character keys will generate a response (i.e. ROLL UP, ROLL DOWN, etc. are ignored). The graphics cursor is turned on, if not already on. If an escape sequence is received by the terminal after it has received the READ CURSOR with WAIT command and before a key is struck, the READ CURSOR command will be aborted. The new sequence will be executed instead.

Read Graphics Cursor with Wait Request: ESC * s 4 ^

The terminal responds: <X>, <Y>,<key code>
<terminator>

where: <X> =X coordinate
<Y> = Y coordinate
<key code> = Decimal value of key struck

For example, if you position the cursor to the screen's lower left corner then press the "A" key, the terminal responds:

+00000, +00000, 065↵

The position bytes are ordered as in the read pen request (Parameter 2). The decimal of ASCII characters are given in ASCII character set tables in Appendix C.

Read Display Size (Parameter=5)

This request returns the number of displayable units in the X and Y axes. It also returns the number of units per millimeter in the display. This request allows you to scale data for use on graphic devices with varying display areas.

Read Display Size Request: ESC * s 5 ^

The terminal responds: <LLX>,<LLY>,<URX>,<URY>,
<MMX>,<MMY><terminator>

where: <LLX>,<URX> = Lower left and upper right x coordinates
<LLY>,<URY> = Lower left and upper right y coordinates
<MMX>,<MMY> = Number of units per millimeter in the x and y axes, (five digits and a decimal point)

The terminal will always return a fixed response. The lower left corner has coordinates of 0,0. The upper right corner has coordinates of 511,389. There are approximately 2 units per millimeter in each axes.

Terminal response: +00000,+00000,+00511,+00389,
00002.,00002.<terminator>

Read Device Capabilities (Parameter=6)

The device capabilities request returns a list of graphic and plotting features available in the terminal. This allows you to use one program for a variety of graphic devices. Not all of the features listed are available in the terminal. The absence of a feature is indicated by a 0. If a feature is present, it may be necessary to send an additional request to determine the exact capabilities present. Where multiple response values are possible, the terminal's standard response is shaded.

Device Capability Request: Esc * s 6 ^

The terminal responds:

<b1>, <b2>,<b3>,<b4>,<b5>,<b6> ,<b7>,
<b8>,<b9>, <b10>,<b11>,<b12>,<b13>,< b14>,
<b15>,<b16><terminator>

where:

<b1> = Clear display
0 = no clear
1 = paper advance
2 = clear (total erase)
3 = partial clear by area

<b2> = Number of Pens (8)

<b3> = Color Capability
0 = black or white
1 = gray levels
2 = color

<b4> = Color Level Capability (2)
"2" refers to the number of color levels for each plane.

<b5> = Area Shading
0 = no
1 =yes (see Read Area Shading Capability)

<b6>,<b7> = Not Used (0,0)

<b8> = Dynamic Modification
0 =no
1 = yes (see Read Modification Capability)

<b9> = Graphics Character Size
0 = fixed
1 =Integer multiples of the basic cell size
2 = any size

<b10> = Graphics Character Angles
0 =fixed
1 =multiples of 90°
2 = multiples of 45°
3 = any angle

<b11> = Graphics Character Slant
0 = fixed
1 = 45°
2 = any angle

<b12> = Dot-Dash Line Patterns
0 = none
1 = predefined only
2 = user defined and predefined

<b13>-<b16> = Not Used (0,0,0,0)

The terminal will always respond:

3,8,2,2,1,0,0,1,1,1,1,2,0,0,0,0,<terminator>

Read Graphics Text Status (Parameter=7)

The terminal returns the current text size, orientation, slant, and type of justification. Refer to Section V, Graphics Display for a description of graphics text characteristics.

Read Graphics Text Request: Esc * s 7 ^

The terminal returns: <X size>,<Y size>,<origin> , <angle>,<slant><terminator>

where:

<X size> = X dimension of the character cell (sign plus 5 digits)
<Y size> = Y dimension of the character cell (sign plus 5 digits)
<origin> = Relative position of text to cursor (see text origin command) (1 digit)
<angle> = Text angle 0, 90, 180, or 270 (five digits and a decimal point)
<slant> = 00000. or 00045. degrees

Example terminal response:

+00007, +00010, 1, 00090.,00045.CR

Read Zoom Status (Parameter=8)

This request returns the zoom setting. Since the HP 2627A terminal does not have the zoom feature, it always returns constant values.

Read Zoom Status Request: Esc * s 8 ^

The terminal responds:

<zoom size>,<zoom on/off><terminator>

where:

<zoom size> = 001.
<zoom on/off> = 0 for Off

This response is always:

001., 0CR

Read Relocatable Origin (Parameter=9)

The position of the relocatable origin is returned as x and y coordinates.

Read Relocatable Origin Request: Esc * s 9 ^

The terminal responds:

<X coordinate>,<Y coordinate><terminator>

With the origin set to the screen's lower left corner, the terminal responds:

+00000, +00000

Read Reset Status (Parameter=10)

You can determine whether or not the terminal has executed a full reset (or Power On) since the last time reset status was checked. This will tell you whether or not you need to reestablish terminal settings or images before resuming terminal functions. An additional seven bytes are returned but are not used.

Read Reset Status Request: Esc * s 10 ^

The terminal responds: <reset>,<b1>,<b2>, <b3>,<b4>,<b5>,<b6>, <b7><terminator>

where <reset> = 0 No full reset since last check
or
1 Terminal has been reset
<b1>-<b7> = 0 (not used)

Read Area Shading Capability (Parameter=11)

The area shading capability of the terminal can be read. These are fixed for the terminal.

Read Area Shading Request: Esc * s 11 ^

The terminal will always respond: 2,8,8, <terminator>

The "2" indicates that the area shaded can be a polygon. The first "8" indicates that the shading pattern is 8 units wide. The second "8" indicates that the shading pattern 8 units high.

Read Graphic Modification Capabilities (Parameter=12)

You can read the terminal's dynamic graphics capabilities. This is the ability of the terminal to change selected portions of the display. These are fixed for the terminal.

Read Graphic Modification Capabilities Request: Esc * s 12 ^

The terminal will always respond: 1,1 <terminator>

These two bytes indicate that the terminal has selective erase and complement capabilities.

Any Other Parameter

Any other parameter which has not been assigned causes the terminal I.D. to be returned. This is to prevent an invalid status request from tying up the requesting computer while waiting for a response.

2627A <terminator>

COLOR PAIR STATUS

You can obtain the current color pair definition through a color pair status request. You accomplish this by issuing the following escape sequence:

Esc & v <color pair#> ^

The format for the returned color pair definition is

Esc & v <0/1>m< >a< >b< >c< >x< >z##I

where the:
m parameter shows the color method (0 = RGB, 1 = HSL); < > represents a nonzero number in the range 0.00 to 1.00 (the returned format is an integer followed by two decimal digits; however, a zero value suppresses the return of the associated parameter);
represents a two-digit number in the range 00 to 07

The noted response depends upon the value of <color pair#>.

1. If <color pair#> is an integer between 0 and 7, inclusively, then the color pair definition escape sequence for that color pair number is returned, along with the current color method.

   For example, if color pair 1 is Red on Black and the current color method is RGB, then Esc&v1^ returns:

   (handshake) Esc & v 0m 1.00a 01I<terminator>

2. If <color pair#> is omitted, then the color pair definitions for all eight color pairs (0-7) are returned. (The non-zero foreground and background information for all eight color pairs are returned in one escape sequence.)

For example, if the default color pairs are in effect and the current color method is RGB, then ^[&v^ returns:

(handshake) ^[&v0m1.00a1.00b1.00c00i1.00a01i
1.00b02i1.00a1.00b03i1.00c04i1.00a
1.00c05i1.00b1.00c06i1.00x1.00y
07I<terminator>

2. If <color pair#> is outside the range 0 to 7 then only the current color method is returned.

   For example, if HSL is the current color method, issuing ^[&v99^ returns

   (handshake) ^[&v1M<terminator>

Error Messages and Self-Test

SECTION IX

INTRODUCTION

This section is divided into two portions. The first discusses the various error messages that may appear on the terminal's screen while you are attempting to perform operations through the keyboard. The second discusses the various types of self-tests that are incorporated into the terminal.

ERROR MESSAGES

When the terminal detects a parameter inconsistency or error condition, it locks the keyboard and displays an appropriate error message across the bottom of the screen (replacing the function key labels). Press RETURN to unlock the keyboard, clear the message, and reinstate the current function key labels.

The various possible error messages and their general meanings are as follow:

`Default configs used`
`Press RETURN to clear`

This message is displayed when the terminal attempts to read the content of non-volatile memory but detects a CRC error (e.g., at power-on time, during a hard reset).

To determine whether the problem is a bad battery or a bad RAM chip, run the Terminal Test described later in this section. If the RAM chip used for non-volatile memory is bad, the Terminal Test will fail and generate an appropriate "CMOS RAM" message identifying the faulty chip. If the test passes, then the "default configs used" power-on message indicates that the battery needs to be changed. Instructions on how to change the battery are provided in Section X, "Terminal Maintenance Procedures".

After clearing the message (by pressing RETURN), you may then reconfigure the terminal as you desire.

`No 'TO' device`
`Press RETURN to clear`

You attempted to initiate a device control data transfer (copy line, copy page, copy all) but no destination device is currently defined. Press RETURN, use the "device control" set of function keys to define an external printer as the "to" device, and then retry the copy operation.

TERMINAL SELF-TESTS

The terminal includes five types of self-tests:

- Power-On Test
- Terminal Test
- Identify ROMs
- Datacomm Test
- ROM Test Loop

The Power-On Test is automatically initiated as the result of a power-on sequence. All of the other tests must be initiated using the "service keys" (except the Terminal Test, which can also be initiated programmatically or by using a "MODES" function key).

Power-On Test

The Power-On Test, which is performed automatically whenever you turn on the terminal's power, does the following:

1. Tests the processor and verifies the integrity of all ROM (Read-Only Memory) and RAM (Random-Access Memory) chips, the graphics system, and CMOS memory within the terminal.

If the Power-On Test results are normal:

1. at power on, the terminal beeps once,
2. does the test for about 24 seconds,
3. beeps once again,
4. brings up the MODES group of softkey labels on the terminal screen.

If an error is found, one of the following will occur when the terminal is turned on:

a. The terminal will fail to beep at all.
b. The terminal will beep continuously.
c. After the first beep, the terminal will beep 1 to 14 times and no softkey labels will appear on the screen.

If the terminal fails to beep, check to make sure the keyboard is connected properly, and try again. If the problem persists, call the nearest HP Sales and Service Office and arrange to have the terminal repaired.

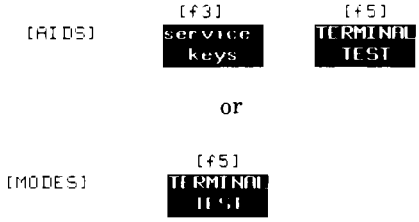
If one of the other error conditions occur (beeping too many times), also call the nearest HP Sales and Service Office and arrange to have the terminal repaired.

Terminal Test

This test does the following:

1. Performs a graphics test on the 2627A. The test checks both the vector generating function as well as the graphics memory. If this test fails, a message indicating the failing component may be displayed. The display shows one pattern in which a graphics raster is swept from left to right, first in red, then in green, and finally in blue.
2. Displays the message "TESTING" at the bottom of the screen (on the same line where the function key labels normally appear).
3. Verifies the integrity of all firmware ROM chips within the terminal.
4. Non-destructively verifies the integrity of all RAM chips within the terminal (including the one used for non-volatile memory).
5. Displays the test pattern shown in figure 9-1.

To initiate the Terminal Test press the following keys in the sequence shown:



If a ROM error is detected, the following message is displayed across the bottom of the screen:

```
ROM ERR #x
Press RETURN to clear
```

where "x" will be a number from 1 to 6. This message contains information identifying the bad ROM chip(s) and describing the nature of the detected error condition. In such a case, or for any other error message, write down the message so you can relate it accurately to your HP Service Representative over the telephone (this allows him to arrive prepared with the proper replacement parts).

If a RAM error is detected, the following message is displayed across the bottom of the screen:

```
RAM ERR #x
Press RETURN to clear
```

where "x" is 1 to 8. This message also contains information identifying the bad RAM chip(s) and describing the nature of the detected error condition. Write down the message and call your nearest HP Service Representative.

If the ROM and RAM chips all pass the test but the test pattern on the screen is malformed, then this would suggest a problem with the video portion of the terminal (the sweep mechanism, the yoke alignment, and so forth).

Identify ROMS

To generate a descriptive list of all ROM chips installed in the terminal, press the following keys in the order shown:

A list similar to the one shown in figure 9-2 is displayed on the screen.

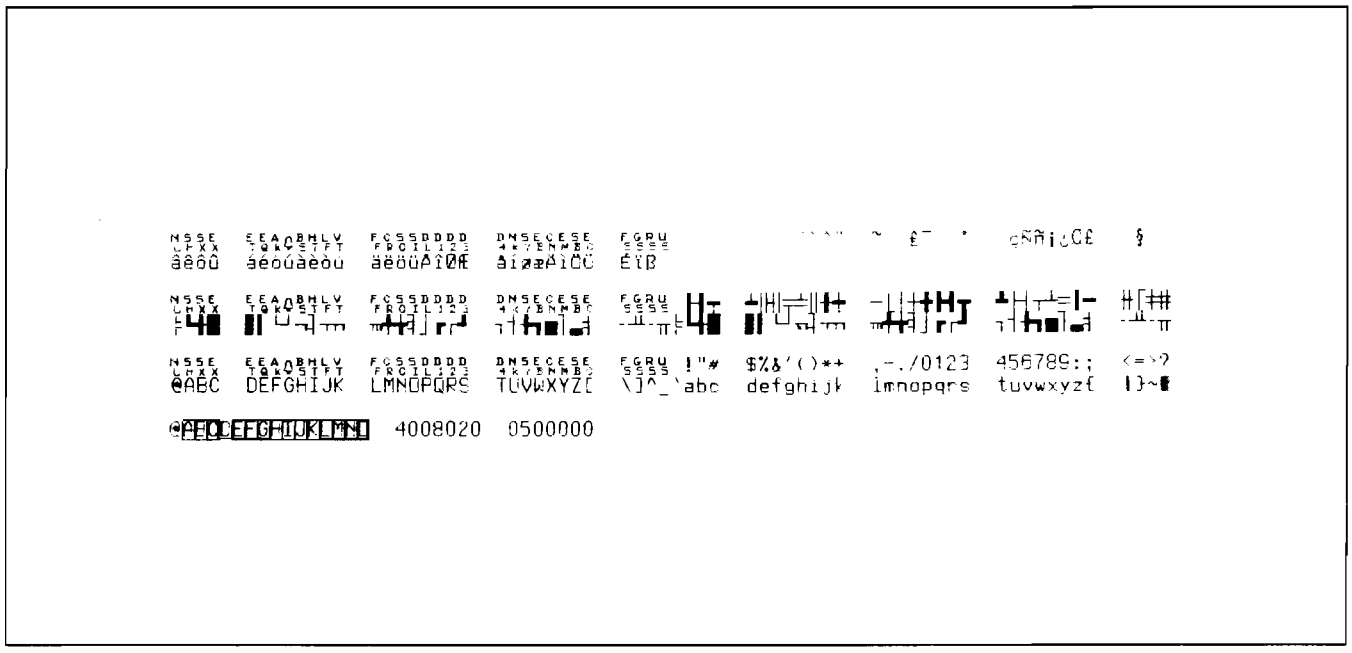


Figure 9-1. Screen Test Pattern, Standard Terminal

Datacomm Test

The data communications (datacomm) self-test checks the 50-pin port at the rear of the terminal. It also checks the 25-pin external peripheral port of the HP 2627A.

To enable the data communications (datacomm) self-test press the following keys in the sequence shown:

A test hood (HP part no. 02620-60056 or 02620-60321, used for HP Direct Connect Type 422) must be connected to the 50-pin datacomm port, and test hood (part no.02620-60062) must be connected to the 25-pin external peripheral port to perform the datacomm test. Otherwise, "Datacomm Error 1" will appear if the test is attempted without them being connected.

Data Communications Test Hoods

Part No.	Description
02620-60056	This is a male 50-pin test hood for use on the datacomm port on the HP 2627A terminal.
02620-60062	This is a male RS-232C test hood for use on the external peripheral port on the HP 2627A terminal.
02620-60321	This is a male HP Direct Connect Type 422 test hood for use on the terminal's datacomm port.

Firmware ROMs

1818-3053	22XX
1818-3054	22XX
1818-3055	22XX
1818-3056	22XX
1818-3057	22XX
1818-3058	22XX

Figure 9-2. ROM Identification Listing

These test hoods can be ordered separately. Contact your local HP Sales Office.

The loopback test consists of a data loopback operation (which checks to see if the character sent is the same as the character received), a baud rate test (which verifies that the baud rate mechanism is functioning properly within ±2% of the configured baud rate), and a modem control line test.

While the test is executing, "Testing" is displayed on the terminal screen. If no errors are found, the terminal beeps and displays the softkeys. If an error is found, an error message will appear on the terminal screen, similar to a ROM error.

The "Err#" field contains a numeric error code which is interpreted as follows:

1 = Test connector not present
2 = Baud rate too fast
3 = Baud rate too slow
4 = Error in Control lines
5 = Character did not loop back
6 = Received character NOT same as one transmitted
7 = Framing error in character
8 = A character was overrun

ROM Test Loop

The ROM Test Loop checks the positioning of the firmware ROMs and performs a continual or looping cyclic redundancy check (CRC) on the ROMs. If an error is detected, the test will stop with an error message, indicating which ROM failed. If no error is detected, the test will continue until a hard reset is performed.

To initiate the ROM Test Loop, press the following keys in the sequence shown:

To exit the ROM Test Loop, perform a hard reset by pressing the following keys simultaneously:

Terminal Maintenance Procedures

SECTION X

INTRODUCTION

This section provides information on preventive maintenance for your terminal, such as cleaning the screen and keyboard, and on routine maintenance such as replacing the batteries. Procedures for degaussing the screen and adjusting the screen brightness are also included.

CLEANING THE SCREEN AND KEYBOARD

The display screen and the keyboard should be cleaned regularly to remove dust and grease. First, lightly dust the entire terminal using a damp, lint-free cloth or paper towel. The cloth or paper towel should be damp enough to pick up any dust, but should not be wet. Avoid wiping dust or lint into the key area of the keyboard.

Greasy smudges and fingerprints can be removed using a mild solution of soap and water. Avoid spilling any cleaning solution between the keys.

DO NOT use petroleum-based cleaners (such as lighter fluid) or cleaners containing benzene, trichlorethylene, ammonia, dilute ammonia, or acetone because these chemicals could damage the terminal's plastic surfaces.

BATTERY MAINTENANCE

The non-volatile portion of memory that contains the terminal's configuration data is protected against destruction by two "AA" batteries located below the mainframe at the pedestal rear. Figure 10-1 shows the rear panel and the location of the batteries.

The batteries require no special care or maintenance. They should, however, be replaced with two new batteries every 12 months. You may purchase replacement batteries through conventional retail stores. When doing so, request two "AA" alkaline batteries.

The batteries may be replaced with terminal power on or off.

If the terminal is to be turned off, first record the configuration menus, turn power off, and then replace the batteries.

Replace the batteries as follows:

1. Grasp the battery support at points A and B as shown in Figure 10-1.
2. Squeeze the tabs at points A and B toward the center of the battery support with enough pressure to disengage the flanges that hold the battery support in place.

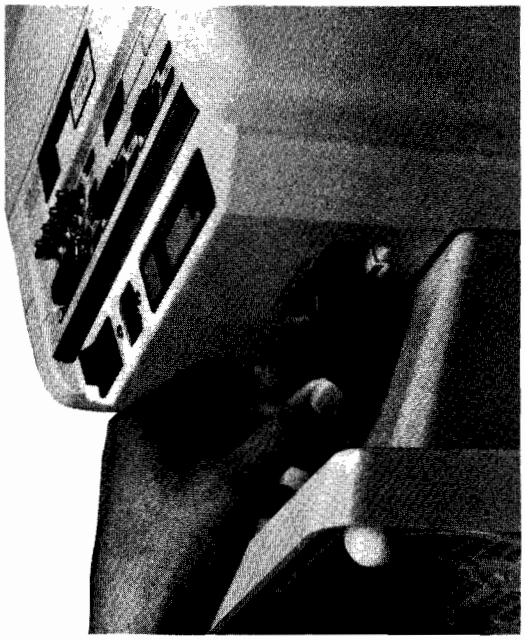


Figure 10-1. Removing the Batteries

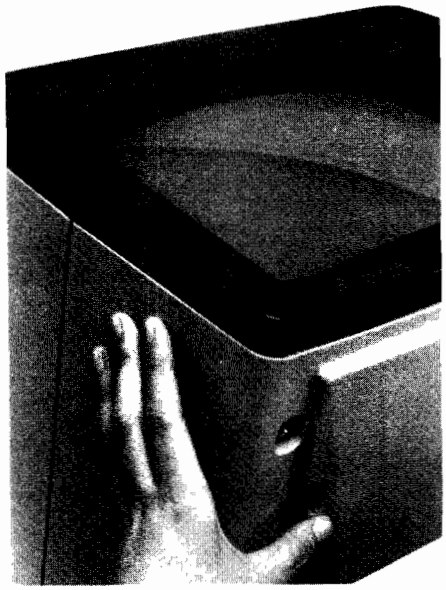


Figure 10-2. Degaussing Switch

3. Gently pull the support downward until it is completely free from the terminal housing.

4. Remove the old batteries from the support.

5. Install the new batteries in the support making sure that they are oriented correctly as shown on the etched side of the support.

6. Reinsert the battery support into the terminal. A raised guide along one side of the battery support ensures that the support is inserted correctly. The raised guide must be facing toward the terminal case when you reinsert the support (otherwise the support will not fit back in the terminal).

DEGAUSSING THE SCREEN

The terminal always degausses itself at power on. If the terminal is moved while it is turned on, however, the colors on the screen may become fuzzy. To refresh the colors, press the degaussing switch up for about three seconds and release. The degaussing switch is located at the bottom of the mainframe on the left side (figure 10-2).

ADJUSTING THE SCREEN BRIGHTNESS

The screen brightness control is located at the bottom of the mainframe on the right side (figure 10-3). To adjust the screen brightness, type in a few characters on the screen, observe the characters for brightness, and adjust the screen brightness control to the desired eye comfort. Avoid full-bright adjustments to the screen, otherwise, permanent etchings to the screen may result.

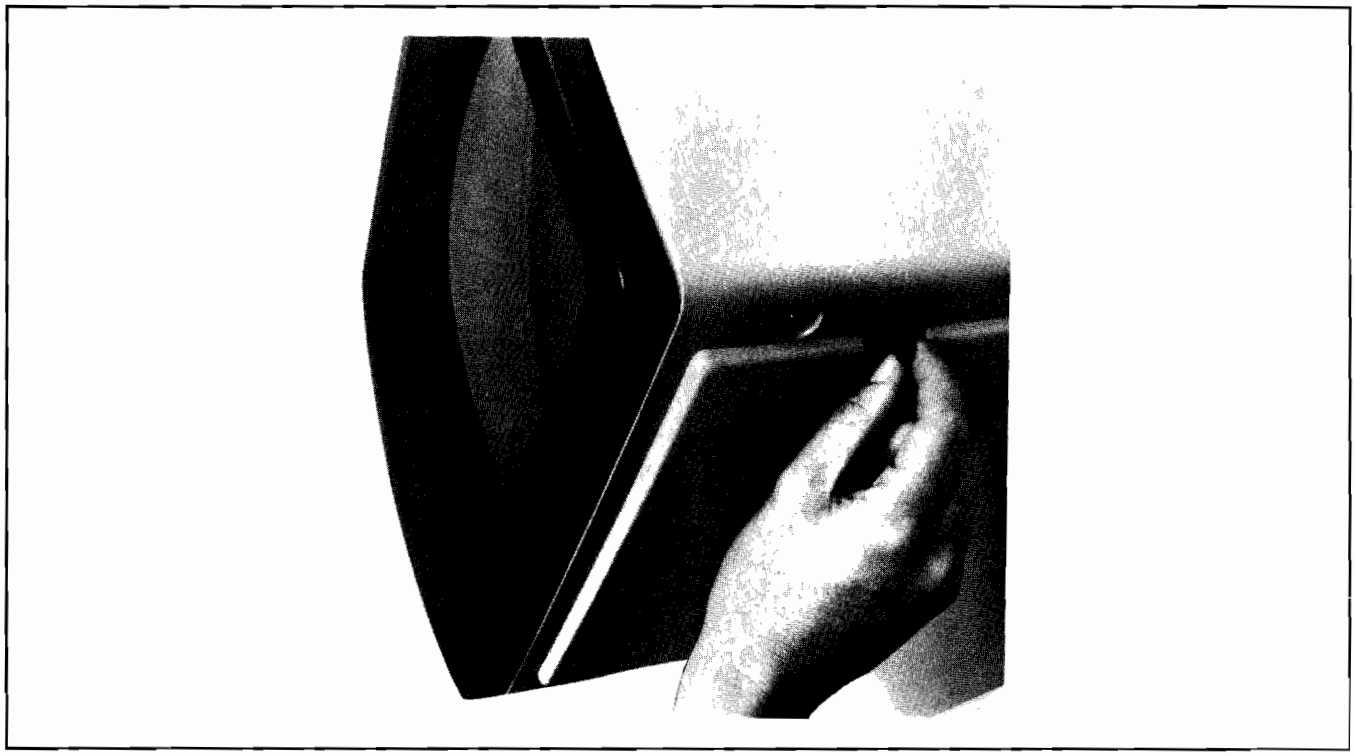


Figure 10-3. Screen Brightness Control

Color Technology

APPENDIX A

INTRODUCTION

If you have never used a color terminal or color output device such as a plotter, your color graphics terminal will provide you with a new dimension in graphics. The terminal has the ability to display data using up to eight different colors. Additional colors can be displayed in graphics using predefined and user-defined colors and patterns.

This appendix briefly discusses the concepts of color and how the terminal generates color displays.

Why Color?

The proper use of color in presenting data can enhance almost any display. Some of the benefits of using color are:

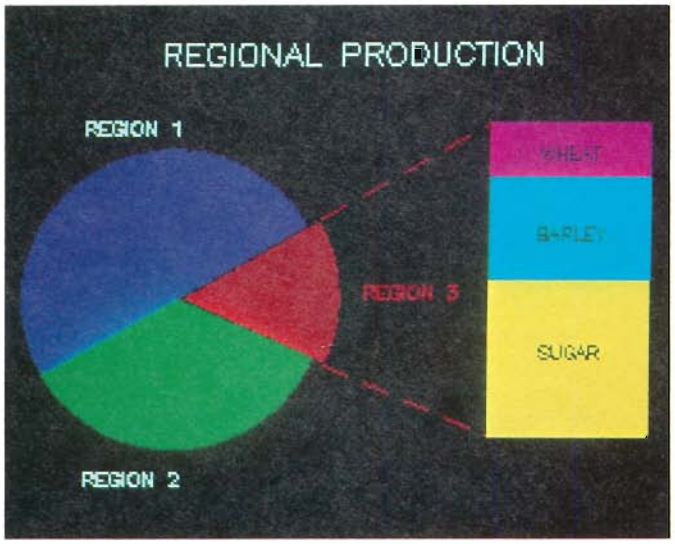
- Highlights data
- Allows rapid comparisons
- Aids understanding
- Increases data density

Using color makes it easier for business managers or professionals to graphically display business data for analysis and presentation. Business graphs can be comprehended much more concisely and clearly, thus increasing efficiency and productivity.

Sales forecasts for different products can easily be understood at a glance. For instance, if corn is represented in yellow, barley in green, and alfalfa in blue, you can easily tell how many tons of each grain are expected to sell.

Color also makes it easier for technicians to monitor expensive or dangerous processes, and decreases errors in engineering design. Data displayed in color means that the task of interpreting information is easily and quickly done. Figure A-1 gives examples of displays using color.

You don't have to consciously read the headings on figure columns if they have been color coded. Similarly you don't have to identify a symbolic shape on a process diagram if you know that pumps are blue or that liquid sodium is red.



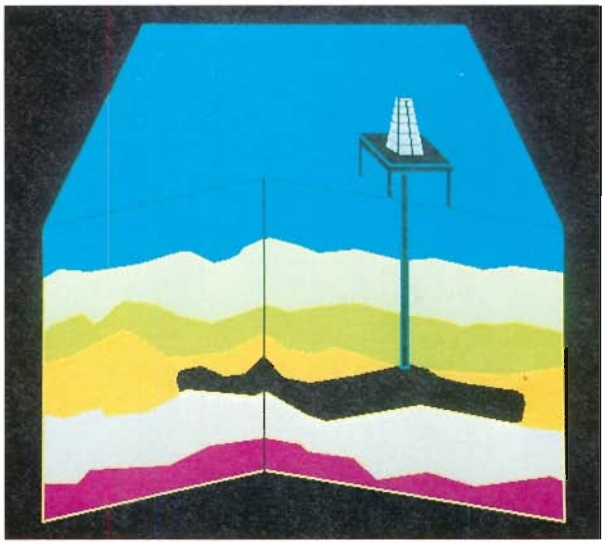


Figure A-1. Examples of Color Graphics

In chemistry for example, atoms used in molecular models are color coded to simplify their study. Oxygen can be white, carbon black, nitrogen blue, hydrogen red, etc. This eliminates the need to label each element and contributes to the scientist's ability to visualize his subject.

In computer aided circuit design of either printed or integrated circuits, color is frequently used to indicate the various layers of material. This provides a handy guide for determining the area being displayed and helps prevent costly errors.

COLOR CONCEPTS

The following paragraphs describe the mechanics of color, the way people perceive color and the way the terminal uses color.

Perception of Color

The perception of color varies from person to person. People have individual preferences for certain colors and combinations of colors. How people perceive color can vary from time to time and under different sets of conditions. These variations are due to the way the brain receives color information.

When we "see" color we are really responding to a mixture of electromagnetic radiation at various frequencies. The normal human eye responds to radiation with wavelengths between 400 nanometers (red) and 700 nanometers (blue). Figure A-2 shows the visible spectrum. The perceived color is a mixture of these wavelengths. The exact mixture of wavelengths determines the color content of the light without regard to intensity. For a given color mix, the strength of radiation is perceived as intensity.

Primary Colors

The light that we see is broken into colors or frequency mixtures by the type of material absorbing or reflecting the light. Any color can be represented by combinations of three colors. These three colors are called primary colors. The terminal uses three basic primary colors (red, green, and blue) to produce additional colors (cyan, yellow, magenta, black, and white). There are two primary color sytems: additive and subtractive. Red, green, and blue are the primary colors for the additive system. Cyan, yellow, and magenta are the primary colors for the subtractive system. The additive and subtractive primary color systems are illustrated in figure A-3.

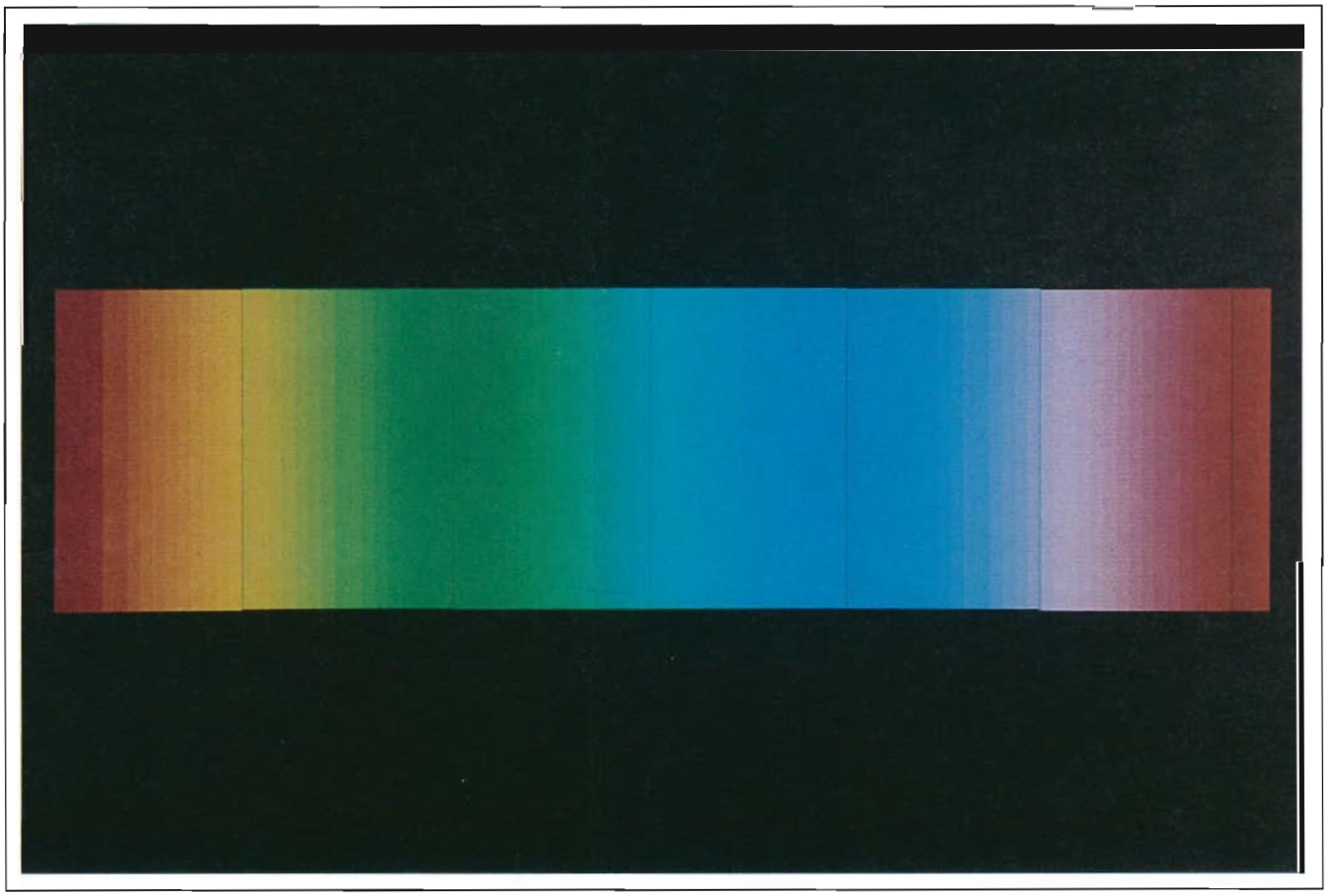


Figure A-2. The Visual Spectrum

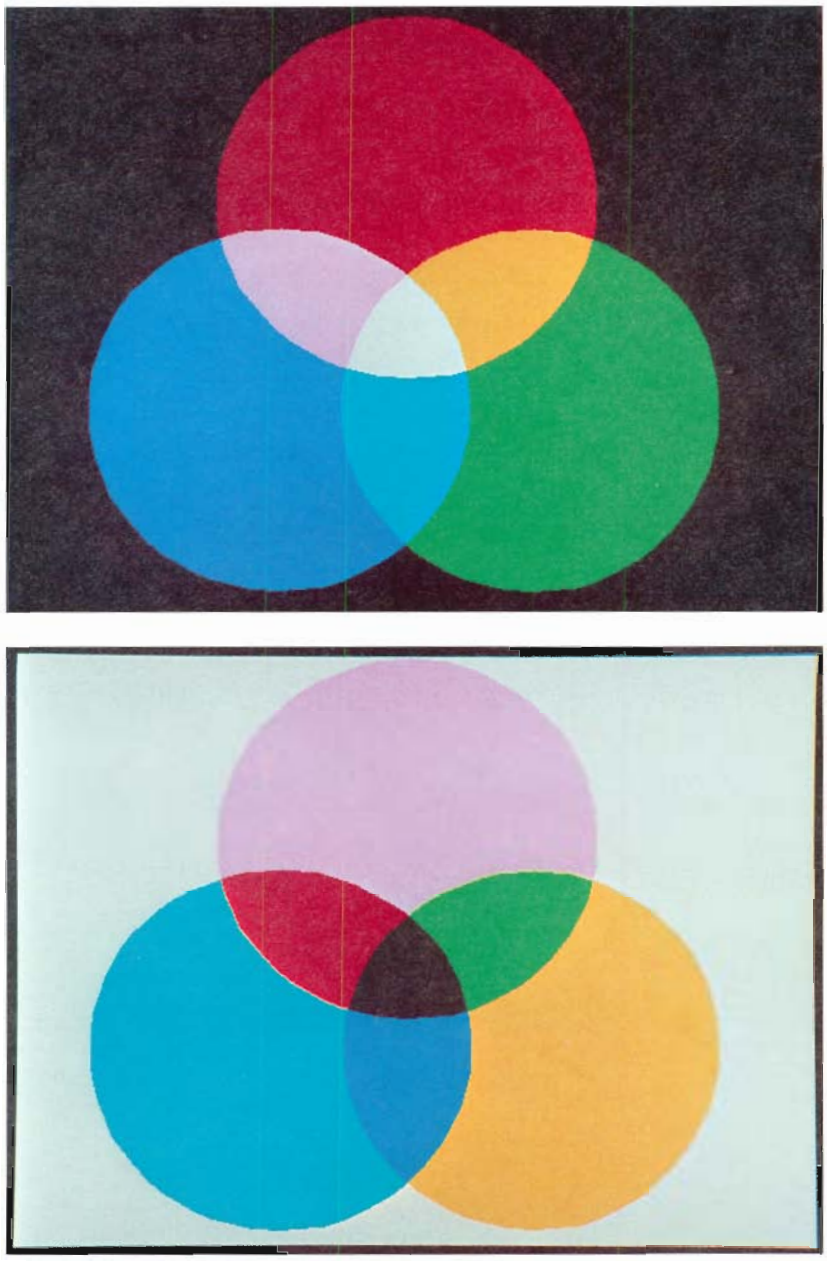


Figure A-3. Creating Colors Using Color Primaries

Additive Color System

The rules of additive primary colors apply when dealing with radiant energy, whether it comes directly from a light source, has been changed by a color filter, or is reflected by some substance. Additive mixture occurs whenever colors are mixed. The additive primaries (red, green, and blue) are used to produce color light as in color terminals and televisons.

By mixing the additive primaries in their purest forms, white is produced (figure A-3). In additive primary color systems, black is the starting reference point for adding colors.

Subtractive Color System

The subtractive primary color system uses three primary colors in certain proportions to create a desired color for printing inks and paints. The rules of subtractive primary colors apply when color is absorbed as by the dye in inks and paints. The subtractive primary colors are cyan, magenta, and yellow.

By mixing various amounts of these subtractive primary colors together, the printer can create most colors. Because inks absorb light imperfectly, a few colors cannot be reproduced. Mixing all three subtractive primary colors in equal amounts produces black (figure A-3).

Color Notation Systems

In order to manipulate color, various means of absolute color definition have been developed. These definition systems allow you to describe the mix of frequencies and the relative strength of each of the components. The color graphics terminal uses two of these notation schemes or color definition "methods". The first method is refered to as "HSL" or Hue, Saturation, and Luminosity. The second is "RGB" or Red, Green, and Blue.

HSL. The Hue, Saturation, and Luminosity method breaks color down into the mix of frequencies (Hue), the purity of the color (Saturation), and the lightness or intensity of the color (Luminosity). The concept of HSL applies to any color, however, the terminal uses it only when generating alphanumeric color pairs.

Hue. Hue is what we normally refer to as color. It refers only to the frequencies and not their purity or intensity. A rainbow is a simple display of various hues. The various frequencies have been scaled and assigned numeric values between 0 and 1. Figure A-4 shows the hue values used by the terminal. Note that the values have been laid out so that the scale can be circular. The hues range from red thru green and blue and back to red.

Saturation. Saturation indicates the amount of color present relative to a gray tone at a constant luminosity. Figure A-5 gives an example of a single hue, red (at luminosity = 1), displayed at various saturation levels.

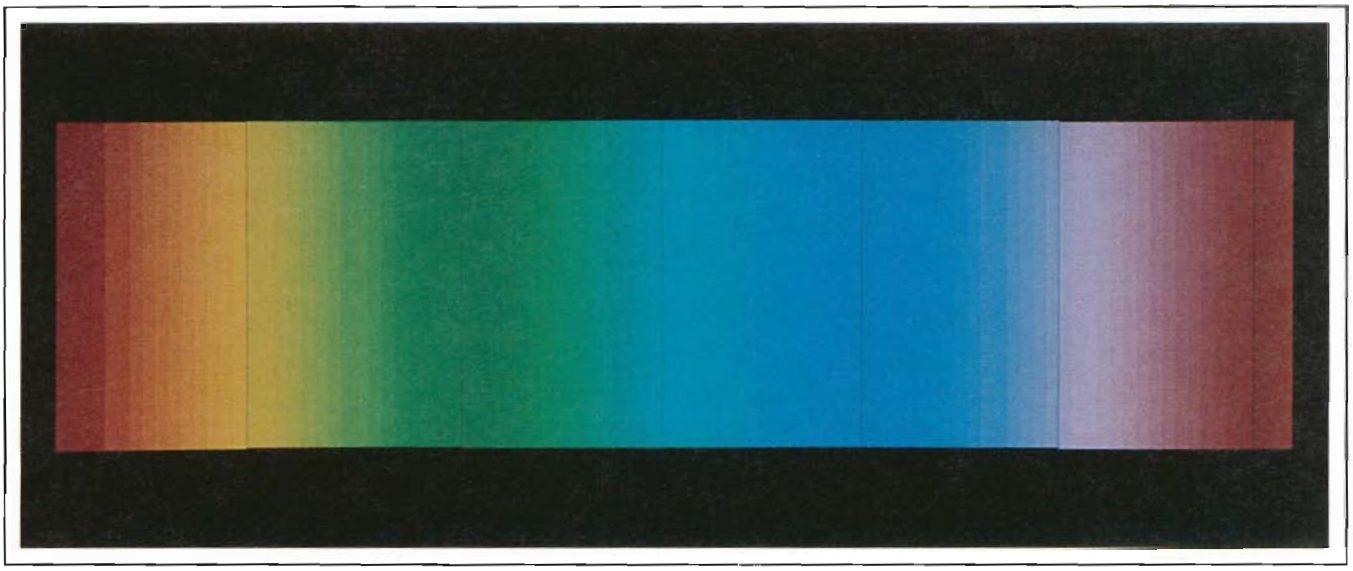


Figure A-4. Color Hues

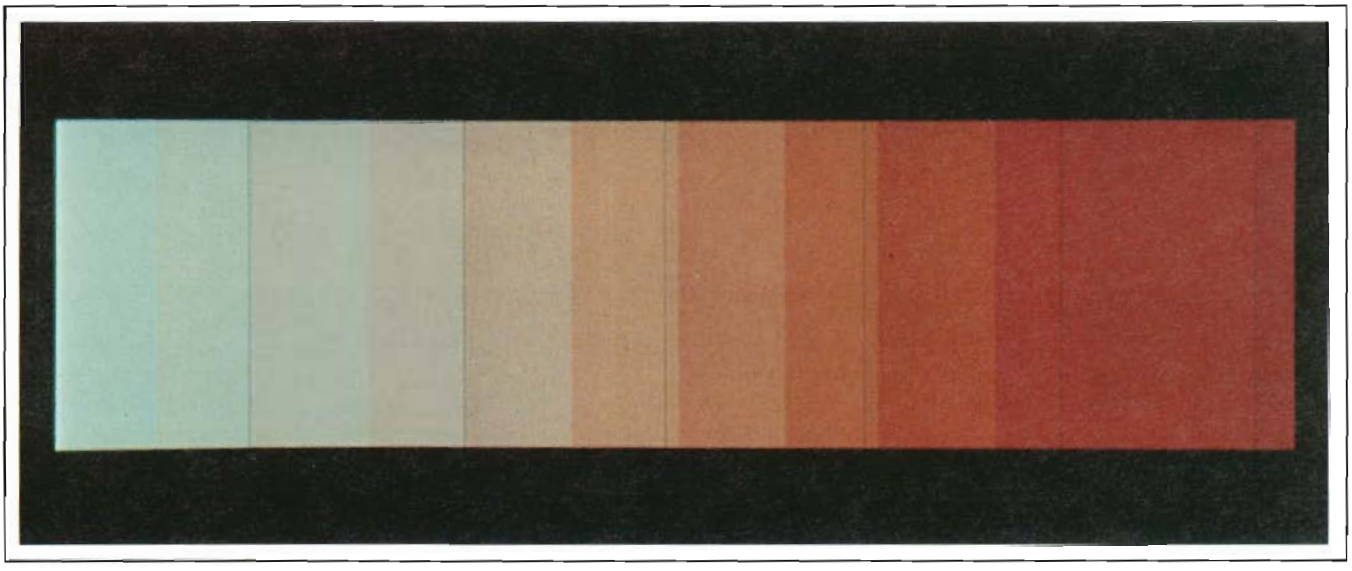


Figure A-5. Saturation

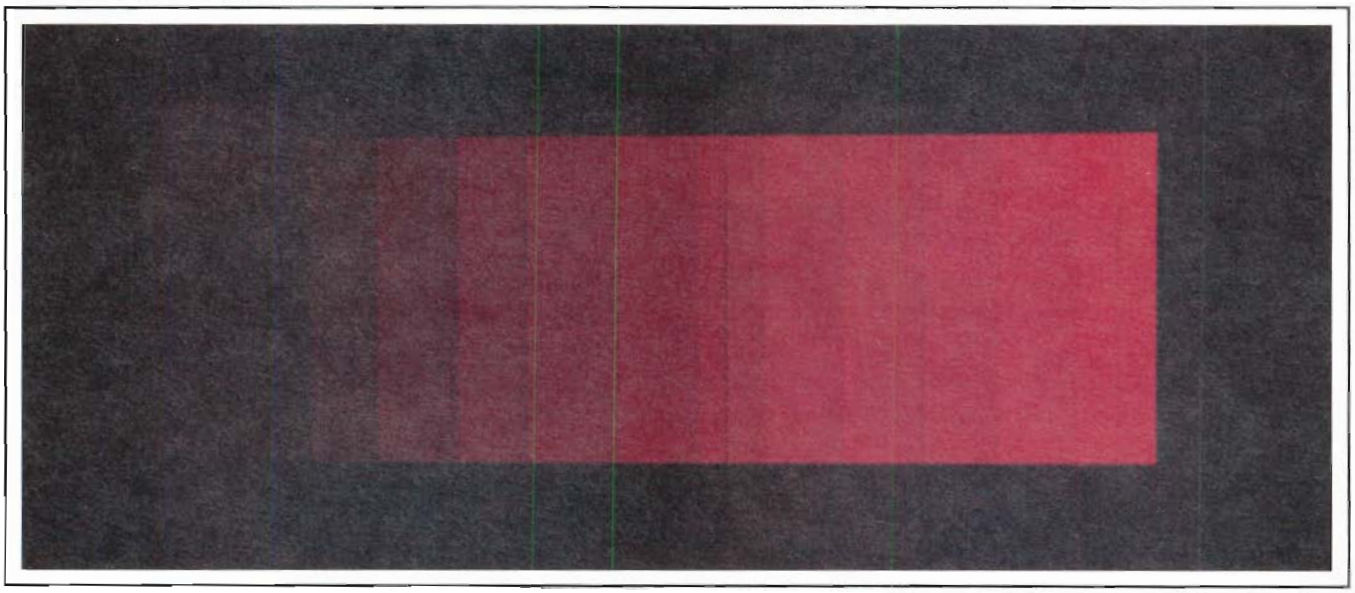


Figure A-6. Luminosity

Luminosity. Luminosity is scaled between 0 (black) and 1 (maximum intensity). Figure A-6 illustrates the effect of the luminosity parameter on color selection.

The HSL color notation method is frequently described using a cylinder as a model. The three parameters, hue, saturation, and luminosity, correspond to three directions on the color cylinder shown in figure A-7. The parameters may range from 0 to 1 as a decimal or common fraction.

Looking at the model, the hue goes circularly around the cylinder. Red is either 0 or 1, green is 1/3, and blue is 2/3. The subtractive primaries are located halfway between these values with yellow being 1/6, cyan 1/2, and magenta being 5/6.

Saturation is the radial distance from the central axis of the cylinder. The value 1 represents the outside (perimeter) while 0 represents the axis or center of the cylinder. Zero saturation gives a shade of gray which depends on the luminosity value. A value of 1 gives the purest color, e.g., if the hue is red then the color red would be at its maximum or full saturation. Luminosity is the vertical axis, the top being 1 or full luminosity and the bottom, zero luminosity, being black. Figure A-7 shows the HSL model.

RGB. The RGB method of notation allows you to specify directly the amounts of red, green, and blue light to be generated in an additive system. The ratio of red, green, and blue correspond roughly to hue; while amounts of red, green, and blue used (compared to the maximum amounts that could be used) correspond roughly to saturation and luminosity.

The R, G, and B values (in the model) range between 0 and 1. They combine in an additive process to create a resultant color. Figure A-8 shows sample colors created using R, G, and B values.

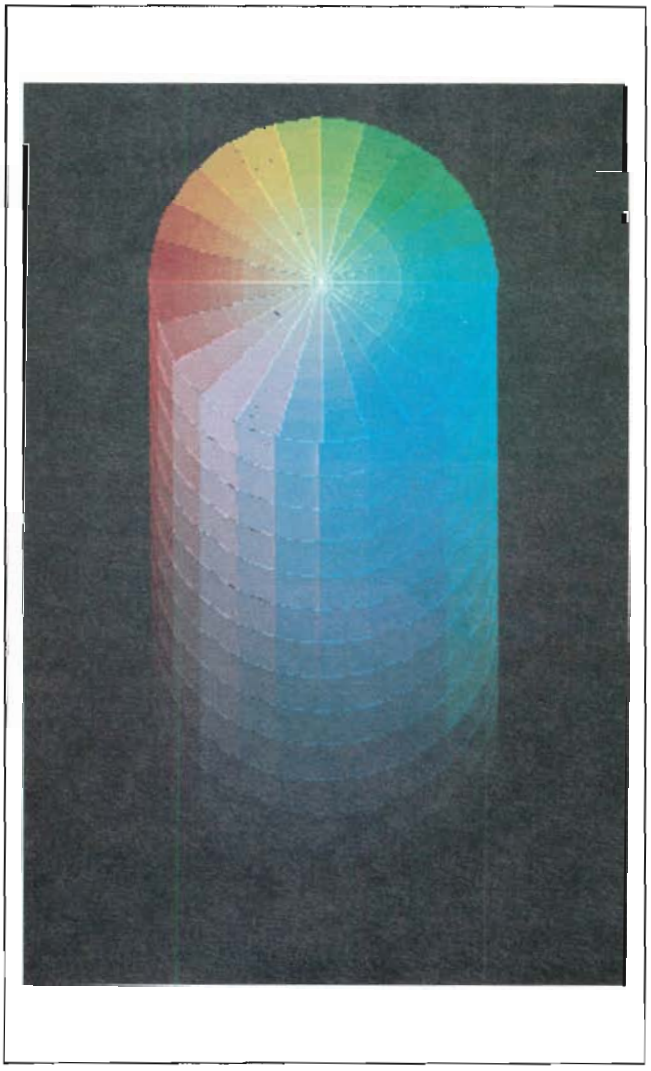


Figure A-7. HSL Cylinder Color Model

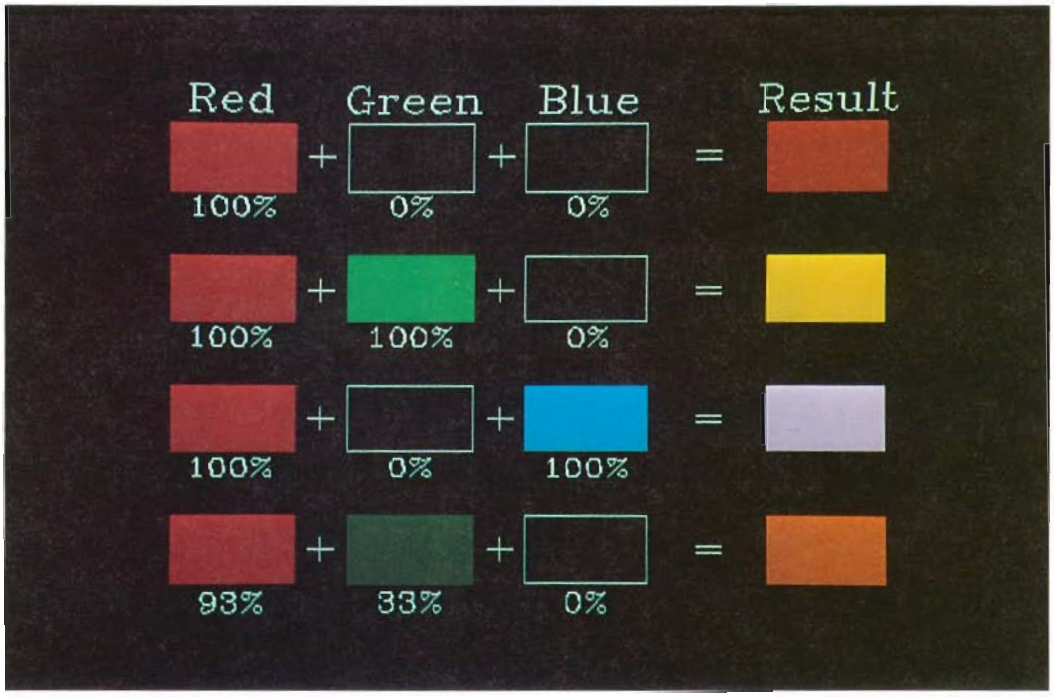


Figure A-8. RGB Color Examples

The RGB notation uses a cube as its color model. The R, G, and B values make up the three axes of the cube (see figure A-9). The hidden corner is the origin, where R=G=B=0 (black).

Figure A-9. RGB Cube Color Model

HOW COLOR IS GENERATED IN THE TERMINAL

The terminal uses the additive primary colors red, green, and blue to generate colors. Figure A-10 illustrates the terminal's color generation system.

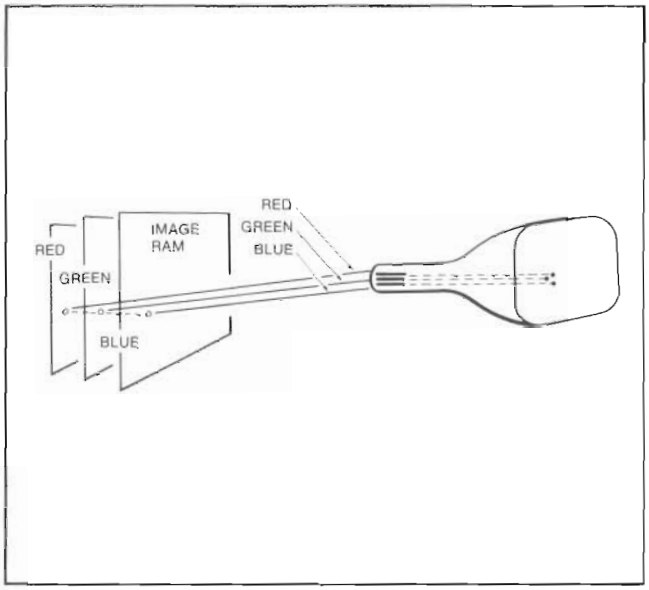


Figure A-10. Terminal Color System

Color CRT

The terminal uses three separate electron beams to energize three different phosphor coatings on the face of the CRT. One phosphor glows red when hit by the beam, the others green and blue. The phosphors are so placed and shielded with a mask so that only the beam of electrons intended for the red phosphor can reach it. The same is true of the electrons from green and blue guns. By causing the R, G, and B phosphor coatings in a particular area to be on or off, almost any resultant color can be generated. Figure A-11 illustrates this process.

The terminal can display eight basic colors: black, red, green, blue, yellow, magenta, cyan, and white. These colors correspond to all combinations of the three CRT guns being fully on or off. Additional colors can be created in graphics display using dithered area pattern techniques. Dithered area patterns use a color half-tone technique to create these additional colors.

Raster Memory

The terminal stores the graphics image to be displayed on the screen in a special area called "raster memory". The raster memory is used to refresh the terminal screen and is made up of three planes of bits. Each plane corresponds to one electron gun or one primary color. There are 390 rows of 512 bits in each plane. The planes are assigned binary values so that a bit on the first plane (red) is a "1", a bit on the second plane (green) is a "2", and a bit on the third plane (blue) is a "4".

For any given point on the terminal screen, the corresponding bit in each memory plane is added to produce a resultant value. The resultant value is in the range 0-7 and corresponds to the pen number used to draw the figure. The result of combining the data in the three memory planes is to create a "paint by the numbers" picture. Figure A-12 illustrates this concept.

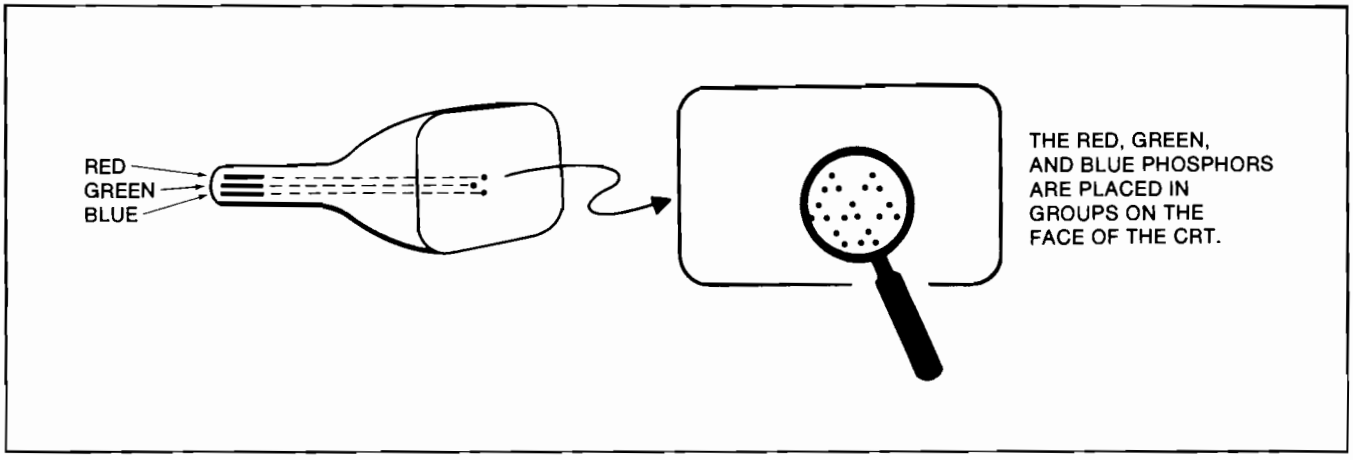


Figure A-11. Color CRT

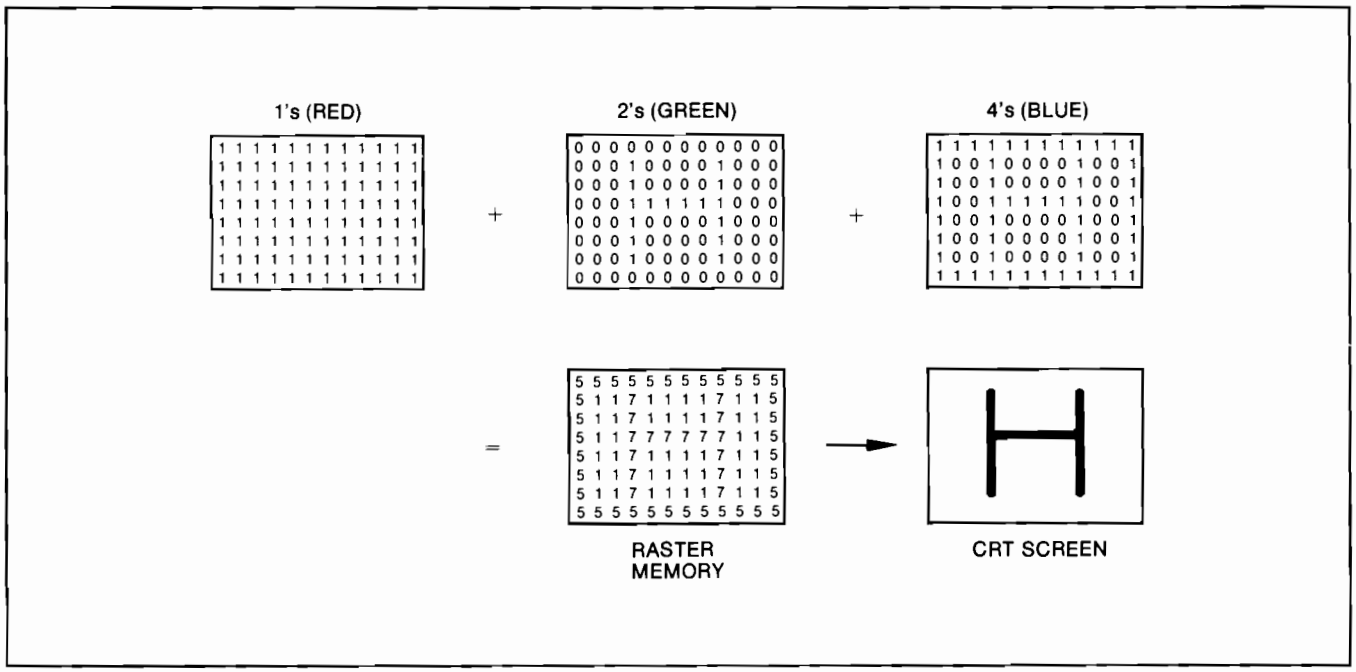


Figure A-12. Raster Memory Planes

Dithering

The terminal uses the additive method with the three primary colors red, green, and blue to produce the basic eight colors. Additional colors can be created using the dithering technique. This technique, which applies to filled areas only, depends on the eyes' ability to integrate closely packed color dots or pixels. By varying the dot or pixel density in each color plane within an area fill, colors can mixed or combined into other colors. The terminal supports 11 predefined dither patterns whose resultant colors closely match the Hewlett Packard plotter pen colors. User defined dither patterns can also be selected.

Dithering specifies the density of dots in each plane of image RAM (random-access-memory). When all three color planes' densities are combined on the screen, dithered colors are created. By using escape sequences, the user can assign varying density values (from 0.00 to 1.00) to each color plane. These values are stored in the terminal until they are requested to fill a polygonal area.

The terminal has three color planes of image memory. Dithering software divides each color plane into 4x4 arrays of dots or pixels. There are 16 pixels in an array. These 16 pixels are arranged into a priority matrix to determine the order in which each pixel is turned on in each color plane (a pixel may be turned on or off). Each color plane may have from 0 to 16 pixels turned on for a given density value. The priority matrix for the terminal is shown below:

1	13	4	16
9	5	12	8
3	15	2	14
11	7	10	6

Each of the three color planes correspond to one of the three primary colors. The first color plane is red, the second is green, and the third is blue.

When density parameters are assigned to the color planes, area density levels are internally calculated for each color plane. These density parameters specify how many pixels are to be turned on for each color plane. The density parameter n ($0<n<1$) is internally mapped to 17 discrete density levels (0 thru 16) and the result is the number of pixels to be turned on.

Now that you are acquainted with the dithering concept, let's continue with an example utilizing all three color planes. Let's assign the density parameters .33, .50, and .25 to the first, second, and third color planes, respectively.

1. The terminal internally maps the density parameters for each color plane to the 17 discrete density levels with the following pixel results:

Color plane 1: .33 = 5/16 = 5 pixels in matrix
Color plane 2: .50 = 8/16 = 8 pixels in matrix
Color plane 3: .25 = 4/16 = 4 pixels in matrix

2. The selected pixels are turned on in each color plane matrix.

Color Plane 1 (Red)

X		X	
	X		
X		X	

Color Plane 2 (Green)

X		X	
	X		X
X		X	
	X		X

Color Plane 3 (Blue)

X		X	
X		X	

Note

X = Pixel is turned on.
Blank = Pixel is turned off.

3. The dithered or resultant color combination is as follows:

WHT	BLK	WHT	BLK
BLK	YEL	BLK	GRN
WHT	BLK	WHT	BLK
BLK	GRN	BLK	GRN

NOTE

When doing area fills with predefined or user defined dithered colors, there is some loss in the effective screen resolution. Therefore, vectors and text superimposed on a dithered background may require extra width or size to be adequately seen.

CONVERTING COLOR NOTATION SYSTEMS

The color notation systems, RGB and HSL, can be converted from one system to the other and vice versa. The following tables provide color notation conversion algorithms. Table A-1 converts RGB values into HSL values and table A-2 converts HSL values to RGB values.

Table A-1. Converting from RGB to HSL

CONDITION	ALGORITHM
Case 1: (R=G=B)	H=0 S=0 L=R(=G=B)
Case 2: (B≤G≤R)	H=(1-(R-G)/(R-B))/6 S=(R-B)/R L=R
Case 3: (G≤B≤R)	H=(5+(R-B)/(R-G))/6 S=(R-G)/R L=R
Case 4: (G≤R≤B)	H=(5-(B-R)/(B-G))/6 S=(B-G)/B L=B
Case 5: (B≤R≤G)	H=(1+(G-R)/(G-B))/6 S=(G-B)/G L=G
Case 6: (R≤B≤G)	H=(3-(G-B)/(G-R))/6 S=(G-R)/G L=G
Case 7: (R≤G≤B)	H=(3+(B-G)/(B-R))/6 S=(B-R)/B L=B
NOTE: Both the input (R,G,B) and output (H,S,L) values are in the range from 0 to 1, inclusively.	

The terms used in table A-2 are defined as follows:

Sextant=Integer (6*H)

Fraction=(6*H)-Sextant

X= L*(1-S)

Y= L*(1-(S*Fraction))

Z= L*(1-(S*(1-Fraction)))

Table A-2. Converting from HSL to RGB

CASE	ALGORITHM
Sexant: 0,6	R=L G=Z B=X
Sexant: 1	R=Y G=L B=X
Sexant: 2	R=X G=L B=Z
Sexant: 3	R=X G=Y B=L
Sexant: 4	R=Z G=X B=L
Sexant: 5	R=L G=X B=Y
NOTE: Both the input (H,S,L) and output (R,G,B) values are in the range from 0 to 1, inclusively.	

Escape Codes

APPENDIX B

KEY(S)		CODE	FUNCTION
TERMINAL CONTROL FUNCTION			
ENTER	(as used in Local mode)	↯ 0	Copy memory to destination(s)
AIDS	margins/ tabs/col SET TAB	↯ 1	Set tab
AIDS	margins tabs/col CLEAR TAB	↯ 2	Clear tab
AIDS	margins/ tabs/col CLR ALL TABS	↯ 3	Clear all tabs
AIDS	margins/ tabs/col LEFT MARGIN	↯ 4	Set left margin
AIDS	margins/ tabs/col RIGHT MARGIN	↯ 5	Set right margin
AIDS	margins/ tabs/col CLR ALL MARGINS	↯ 9	Clear all margins
		↯ @	Delay one second
↑		↯ A	Cursor up
↓		↯ B	Cursor down
→		↯ C	Cursor right
←		↯ D	Cursor left
SHIFT	CTRL RESET	↯ E	Hard reset (power on reset)
SHIFT	↖	↯ F	Cursor home down
RETURN	(with Auto LF disabled)	↯ G	Move cursor to left margin
↖		↯ H	Cursor home up
TAB◄	or TAB►	↯ I	Horizontal tab
CLEAR DISPLY		↯ J	Clear display from cursor to end of memory
CLEAR LINE		↯ K	Clear line from cursor to end of line
INS LINE		↯ L	Insert line
DEL LINE		↯ M	Delete line
DEL CHAR		↯ P	Delete character
INS CHAR		↯ Q	Start insert character mode
INS CHAR		↯ R	End insert character (↯ Q)
ROLL UP		↯ S	Roll up
ROLL DOWN		↯ T	Roll down

KEY(S)		CODE	FUNCTION
NEXT PAGE		↯ U	Next page
PREV PAGE		↯ V	Previous page
		↯ W	Format mode on
		↯ X	Format mode off
MODES	DISPLY FUNCTN	↯ Y	Display Functions mode on
MODES	DISPLY FUNCTN *	↯ Z	Display Functions mode off
		↯ [	Start unprotected field*
		↯]	End unprotected field
		↯ ^	Primary terminal status request
		↯ `	Sense cursor position (relative)
		↯ a	Sense cursor position (absolute)
		↯ b	Unlock keyboard
		↯ c	Lock keyboard
		↯ d	Transmit a block of text to computer
		↯ f	Modem disconnect
		↯ g	Soft reset
↖		↯ h	Cursor home up
TAB◄	or SHIFT TAB►	↯ i	Backtab
SHIFT	USER KEYS	↯ j	Enable User Softkey Definition menu
USER KEYS	or AIDS or MODES	↯ k	Disable User SoftkKey Definition menu
MODES	MEMORY LOCK	↯ l	Begin Memory Lock mode
MODES	MEMORY LOCK *	↯ m	End Memory Lock mode
USER KEYS	f1	↯ p	Default definition for user definable function key f1
USER KEYS	f2	↯ q	Default definition for user definable function key f2

*↯[should be used when designing forms. However, ↯{ will be interpreted as ↯[if it is encountered.

KEY(S)	CODE	FUNCTION
TERMINAL CONTROL FUNCTION (continued)		
USER KEYS f3	ESC r	Default definition for user definable function key f3
USER KEYS f4	ESC s	Default definition for user definable function key f4
USER KEYS f5	ESC t	Default definition for user definable function key f5
USER KEYS f6	ESC u	Default definition for user definable function key f6
USER KEYS f7	ESC v	Default definition for user definable function key f7
USER KEYS f8	ESC w	Default definition for user definable function key f8

KEY(S)		CODE	FUNCTION
SHIFT service keys	TERMINAL TEST	ESC z	Initiate terminal self test
MODES TERMINAL TEST			
		ESC ~	Secondary terminal status request
SHIFT NUM		ESC &k <x>0	x = 0 **Enable numeric pad**
			x = 1 **Enable graphics pad**

CURSOR CONTROL OPERATIONS

NOTE

Columns and rows are numbered starting with 0 as the leftmost column and the top row.

ESC &a<col>c <row>Y	Moves the cursor to column "col" and screen row "row" on the screen (screen relative addressing).
ESC &a <col>c <row>R	Moves the cursor to column "col" and row "row" in memory (absolute addressing).
ESC &a ±<col>c ±<row> Y	Moves the cursor to column "col" and row "row" (on the screen) relative to its present position ("col" and "row" are signed integers). A positive number indicates right or downward movement and a negative number indicates left or upward movement.
ESC &a ±<col>c ±<row> R	Moves the cursor to column "col" and row "row" relative to its present cursor position in memory ("col" and "row" are signed integers). A positive number indicates right or downward movement and a negative number indicates left or upward movement.

CONFIGURATION OPERATIONS

ESC &q 0L	Unlock configuration.
ESC &q 1L	Lock configuration.

ESCAPE SEQUENCE	MENU FIELD	ENTRY VALUE	X
ESC &k <x>A	AUTO LF*	OFF	x=0
		ON	x=1
ESC &k <x>B	BLOCK MODE*	OFF	x=0
		ON	x=1
ESC &k <x>C	Caps Lock	OFF	x=0
		ON	x=1
ESC &k <x>I	ASCII 8 Bits	NO	x=0
		YES	x=1
ESC &k <x>J	FrameRate	60	x=0
		50	x=1
ESC &k <x>K	Auto Keyboard Lock Mode	OFF	x=0
		ON	x=1
ESC &k <x>L	LocalEcho	OFF	x=0
		ON	x=1
ESC &k <x>M	MODIFY ALL*	OFF	x=0
		ON	x=1
ESC &k <x>N	SPOW Latch#	OFF	x=0
		ON	x=1
ESC &k <x>P	CAPS+	OFF	x=0
		ON	x=1
ESC &k <x>R	REMOTE MODE*	OFF	x=0
		ON	x=1
ESC &s <x>A	XmitFnctn(A)	NO	x=0
		YES	x=1
ESC &s <x>B	SPOW(B)	NO	x=0
		YES	x=1
ESC &s <x>C	InhEolWrp(C)	NO	x=0
		YES	x=1
ESC &s <x>D	Line/Page(D)	LINE	x=0
		PAGE	x=1
ESC &s <x>G	IndHndShk(G)	NO	x=0
		YES	x=1
ESC &s <x>H	Inh DC2(H)	NO	x=0
		YES	x=1
ESC &s <x>N	Esc Xfer(N)	NO	x=0
		YES	x=1
ESC &s <x> p <y>Q	Compat(P,Q)	UNSCALED	x=0 y=1
		SCALED	x=1 y=0
		OFF	x=1 y=1 or x=0 y=0

* Softkey label in MODES selection menu: asterisk in label indicates "ON"; no asterisk indicates "off."

#"internal logical latch" that is distinct from spow(B) field in terminal configuration menu.

+Terminal key cap.

DATA OPERATIONS

The following escape sequences control data transfers.

`ESC&p<x>S` Assigns "x" as source device;
where x = 3 device = alpha display
= 7 = graphics display

`ESC&p<x>D` Assigns "x" as the destination device;
where x = 3 device = alpha display
= 4 = external printer

Note: You may set up one source and multiple destinations within one sequence.

`ESC&pB` Copy the cursor's current line to all designated destination devices.

`ESC&pF` If source device is alpha display:
Copy the display screen from the cursor's current line through the last displayed line to all destination devices.
If source device is graphics display:
Copy entire graphics display to selected destination device.

`ESC&pM` **Copy the contents of alpha display memory from the cursor's current line through the end of display memory to all destination devices.**

If source device is graphics display, copy entire graphics display to selected destination device.

Example: To copy graphics display to external printer,
Enter: `ESC&p7s4dF`

In the following sequences, device "4" (external printer) is the only valid assignment for the "x" parameter.

`ESC&p<x>^` Requests the status of device "x".
`ESC&p<x>u0C` Generates a form feed on device "x".
`ESC&p<n>p<x>u1C` Skips "n" lines on device "x".
`ESC&p<x>u<2-10>C` Generates one form feed to device "x" (same as 0C since the 2627 interprets any integer between 2 and 10 as 0).

Note: The "u" parameter denotes a temporarily assigned destination device. It does not affect any previously selected "to" device. Whenever you omit the "u" parameter, the currently selected "to" device in the softkey tree becomes the default destination.

The "d" parameter selects a new destination device. This choice is indicated by an asterisk in the corresponding softkey label. Escape sequence alterations are not stored in non-volatile memory. Therefore, a hard reset or power off ends the selection's active state.

`ESC&p11C` Turns on log bottom mode.
`ESC&p12C` Turns on log top mode.
`ESC&p13C` Turns off either logging mode.
`ESC&p20C` Turns on record mode.
`ESC&p<n>W<data string>` Transfers "n" bytes of the data string, in binary form, from the computer to the selected destination device.
`ESC&pW<data string>` Transfers the data string, in ASCII form, from the computer to the selected destination device. Either the 256th byte or the ASCII linefeed character terminates the string.

FORMAT MODE

ESC [*	Starts a field.
ESC]	Ends the field.

*Same effect achieved with ESC {

FUNCTION KEY AND ERROR MESSAGE OPERATIONS

To enable and disable the function keys (F1 thru F8), use the following escape sequence:

ESC&j <x>

x	MEANING
A	Display the Modes set of function key labels.
B	Enable the User function keys. (The user key labels are displayed.)
C	Clear message and return function key label to the screen.
@	Remove the function key labels from the screen. However, the user function keys are still active.

To enable or disable the Function Control keys:

S	Disables the [MODS], [MODES], and [USER KEYS] keys.
R	Enables the [MODS], [MODES], and [USER KEYS] keys.

To replace the function key definition with your own message:

ESC &j <string length>L <message>

"String Length"—A number (up to 160) indicating the number of characters in the string.

"Message"—The content of the message.

Note: Escape sequences and control characters are displayed as if display functions were on.

To define the function keys:

ESC&f <attribute>a <key>k <color pair>c <label length>d
<string length>l <video enh>V <label> <string>

TERM	SYMBOL	MEANING	DEFAULT
Attribute (a)	0	Normal (N)	0
	1	Local only (L)	
	2	Transmit only (T)	
Key (k)	1	[f1] function key	
	2	[f2] function key	
	3	[f3] function key	
	4	[f4] function key	
	5	[f5] function key	
	6	[f6] function key	
	7	[f7] function key	
	8	[f8] function key	
Label length (d)	0 thru 16	Number of characters in the label.	0
String length (l)	0 thru 80	Number of characters in the string.	1
	−1	Clears the content of the string.	
Label	(none)	The label is entered at this point in the sequence.	
String	(none)	The character string is entered at this point in the sequence.	
Color pair (c)	0 thru 7	Selects associated color pair	
Video Enh (v)	0 thru 15	Selects the video enhancement indicated by value "v".	

"v" Value	Blinking	Inverse Video	Under-line	Half-Bright
0				
1	X			
2		X		
3	X	X		
4			X	
5	X		X	
6		X	X	
7	X	X	X	
8				X
9	X			X
10		X		X
11	X	X		X
12			X	X
13	X		X	X
14		X	X	X
15	X	X	X	X

DISPLAY ENHANCEMENTS OPERATIONS

To start and end display enhancements:

Esc&d <char> Selects the display enhancement indicated by "char" to begin at the present cursor position.

	"char"														
	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Half-Bright									X	X	X	X	X	X	X	X
Under-line					X	X	X	X					X	X	X	X
Inverse Video			X	X			X	X			X	X			X	X
Blinking		X		X		X		X		X		X		X		X
End Enhance-ment	X															

ALTERNATE CHARACTER SET SELECTION

Selects alternate character set as:

Esc) @ Base Set

Esc) A Base Set (in 7-bit mode)

Esc) B Line Drawing Set

COLOR CONTROL

Esc&v <parameter>	
<decimal>a	Parameter 1 for foreground (Red or Hue)
<decimal>b	Parameter 2 for foreground (Green or Saturation)
<decimal>c	Parameter 3 for foreground (Blue or Luminosity)
<0/1>m	Method to be used (RGB or HSL)
<color-pair #>s	Select color pair
<decimal>x	Parameter 1 for background (Red or Hue)
<decimal>y	Parameter 2 for background (Green or Saturation)
<decimal>z	Parameter 3 for background (Blue or Luminosity)
<color-pair #>i	Color-pair to be initialized
<color-pair #>^	Color-pair status

Note: Valid "decimal" values range from 0.00 through 1.00. Operative "color-pair" values are 0 through 7. To return all 8 color pair values from the color-pair status request, omit the number parameter.

DISPLAY WINDOW

Esc&w12F	Turns on alpha display window (topmost 24 rows) of screen.
Esc&w13F	Turns off alpha display window.
Esc&x<y>C	Send cursor position (SCP) Mode. OFF y=0 ON y=1

KEY(S)		CODE	FUNCTION

GRAPHICS CONTROL SEQUENCES

ESC * <control sequence>

d = Display control
e = Image control
I = Graphics text label
m = Mode control
n = Graphics text
p = Plot control
s = Status
t = Compatibility mode
w = Graphics initialization

DISPLAY CONTROL

ESC *d <parameters>

GRAPH CLEAR	<pen#>	a	Clear graphics memory
— — —	<pen#>	b	Set graphics memory
GRAPH DSPLY		c	Turn on graphics display
GRAPH DSPLY		d	Turn off graphics display
ALPHA DSPLY		e	Turn on alphanumeric display
ALPHA DSPLY		f	Turn off alphanumeric display
GRAPH CURSOR		k	Turn on graphics cursor
GRAPH CURSOR		l	Turn off graphics cursor
— — —		m	Turn on rubberband line
— — —		n	Turn off rubberband line
— — —	<x,y>	o	Move graphics cursor absolute
<, ^, >, v	<x,y>	p	Move graphics cursor incremental
— — —		q	Turn on alphanumeric cursor
— — —		r	Turn off alphanumeric cursor
— — —		s	Turn on graphics text mode
— — —		t	Turn off graphics text mode
— — —		z	NOP

Example: Clear the graphics display, position the cursor at x=100, y=100, and turn the cursor on.

```
ESC * d a 100,100o K
```

IMAGE CONTROL

ESC * e	<pen #>b		Set background pen #

GRAPHICS LABEL

ESC * l <text label><CR>, <CR><LF>, <LF><CR>, or <LF>

Example: Send the text "X=TIME, Y=TEMP"

ESC * l X=TIME, Y=TEMP <CR><LF>

VECTOR DRAWING MODE

ESC * m <parameters>

———	<mode>	a	Select drawing mode (0–7)*
———	<line type>	b	Select line type (1–11)**
———	<pattern scale>	c	Define line pattern (2 bytes)
———	<pattern>	d	Define area shading pattern (8 bytes)
———	<x1,y1,x2,y2>	e	Fill area, absolute
———	<x1,y1,x2,y2>	f	Fill area, relocatable
———	<area pattern>	g	Select area pattern

where <area pattern>
= 0 Current dither pattern
= 1 Solid area fill pattern
= 2 User defined area fill pattern
= 3 Predefined area pattern
= 4 Predefined area pattern
= 5 Predefined area pattern
= 6 Predefined area pattern
= 7 Predefined area pattern
= 8 Predefined area pattern
= 9 Predefined area pattern
= 10 Predefined area pattern

———	<pen #>	h	Set area boundary pen
———	<x,y>	j	Select relocatable origin
———		k	Set relocatable origin to current pen position
———		l	Set relocatable origin to graphics cursor position
———	<size>	m	Set graphics text size (1–8)
———	<rotation>	n	Set graphics text orientation (1–4)
———		o	Turn on text slant
———		p	Turn off text slant
———	<0–9>	q	Set graphics text origin
———	<default flag>	r	Set graphics default
———	<d1,d2,d3>	v	Define dither pattern
———	<dither pattern>	w	Select dither pattern (1–12)

where <dither pattern>
= 1 User defined dither pattern
2-12 Predefined dither patterns

———	<pen #>	x	Set primary pen
———	<pen #>	y	Set secondary pen
———		z	NOP (No OPeration)

* 0 (no effect), 1 (clear 1), 2 (jam 1), 3 (complement 1), 4 (jam 2), 5 (or), 6 (complement 2), 7 (clear 2)

** 1 (solid line), 2 (user line pattern), 3 (current pattern), 4 (line #1), 5 (line #2), 6 (line #3), 7 (line #4), 8 (line #5), 9 (line #6), 10 (line #7), 11 (point plot)

Example: Once graphics text is turned on (ESC * dS), select the set drawing mode, a graphics text size of 2 and slanted. Set the text to be center justified.

ESC * m 2a 2m o 4Q

The text is not displayed until a CR is executed.

GRAPHICS TEXT

ESC*n <pen #> x	Set graphics text pen #

PLOTTING COMMANDS

ESC * p <parameters>

———	a	Lift the pen
———	b	Lower the pen
———	c	Use graphics cursor as new point
———	d	Draw a point at the current pen position and lift the pen
———	e	Set relocatable origin to the current pen position
———	f	Data is ASCII absolute
———	g	Data is ASCII incremental
———	h	Data is ASCII relocatable
———	i	Data is absolute
———	j	Data is short incremental
———	k	Data is incremental
———	l	Data is relocatable
———	s	Starts area fill
———	t	Ends area fill
———	u	Lift area boundary pen
———	v	Lower area boundary pen
———	z	NOP

Example: Draw a box 25 units wide and 10 units high, beginning at x=100, y=50.

```
ESC * p e f 100 50 g 25,0 0,10 -25,0 0, -10Z
```

GRAPHICS STATUS

ESC * s <parameter> ^

———	1	Read device I.D.
———	2	Read pen position
———	3	Read graphics cursor position
———	4	Read cursor position and wait for key
———	5	Read display size
———	6	Read graphics capabilities
———	7	Read graphics text status
———	8	Read zoom status
———	9	Read relocatable origin
———	10	Read reset status
———	11	Read area shading
———	12	Read dynamics

Example: Read text status.

ESC * s 7 ^

COMPATIBILITY MODE

ESC * t <parameter>

———	<0/1/2>	a	Set graphics input terminator (0=CR, 1=CR EOT, 2=none)
———	<0/1>	b	Set Page Full Break strap (0=out, 1=in)
———	<0/1>	c	Set Page Full Busy strap (0=out, 1=in)
———		z	NOP

Example: Select a CR input terminator and set the Page Full Busy strap.

ESC * t 0a 1C

GRAPHICS INITIALIZATION

ESC * w r Graphics Hard Reset

Alternate and International Character Sets

APPENDIX C

INTERNATIONAL LANGUAGE CAPABILITY

The terminal comes with an extended character set that supports the special characters associated with several international languages. Regardless of the keyboard used, the extended character set allows you to configure the terminal so that various keys and data communication codes are interpreted (and displayed) in your chosen language. Table C-1 shows the replacement characters for each language selection based upon their ASCII decimal value.

You typically use these numeric values within programs. An example illustrating several concepts introduced in this appendix, including the use of decimal ASCII values, occurs under the discussion of 8-bit mode.

You must note that no correlation necessarily exists between the ASCII numerical values and the corresponding position of keys upon the associated keyboards. For example, the ASCII decimal value "92" maps into the back slant (\) on the USASCII keyboard. On the French keyboards, "92" maps into c-cedilla (ç). However, the c-cedilla key on the French keyboard does not physically correspond to the USASCII keyboard's back slant key.

Setting the Terminal Configuration menu's `Language` field to a specific language automatically changes the keyboard to agree with that language's keyboard layout. If you are using the standard USASCII keyboard, for example, but change the `Language` field to "`FRANCAIS azM`", typing "Francais" requires your pressing the "Q" key to enter each "a".

Selecting An International Language

Languages are selected in the `Language` field of the Terminal Configuration menu. The values for this field are:

`USASCII`	(United States)
`SVENSK/SUOMI`	(Swedish/Finnish)
`DANSK/NORSK`	(Danish/Norwegian)
`FRANCAIS azM`	(French AZERTY layout with mutes)
`FRANCAIS qwM`	(French QWERTY layout with mutes)
`FRANCAIS az`	(French AZERTY layout)
`FRANCAIS qw`	(French QWERTY layout)
`DEUTSCH`	(German)
`UK`	(United Kingdom)
`ESPANOL M`	(Spanish with mutes)
`ESPANOL`	(Spanish)

`USASCII` is the default setting for this field.

A "M" in the `Language` field represents a mutes designator. For the French and Spanish keyboard layouts, the mutes designator indicates how diacritic and accent keystrokes are handled. The procedure is as follows:

> Upon entering a diacritic mark (such as ^ or `), the cursor remains in the same position. If the next-typed character can be combined with that mark, the two characters are merged before the cursor advances to the next position. (The acceptable characters form the set: a, e, i, o, u, A, O, U). If the next-typed character is unacceptable, the just entered character replaces the "mute" symbol as the displayed character and the cursor advances to the next position. The case may arise, however, when you want to enter just the diacritic mark or accent character. Therefore, if you type a space after a "mute" symbol, the "mute" symbol remains displayed and the cursor advances to the next character position.

Table C-1. Characters Which Change with Character Set Selection

	KEYBOARD	DECIMAL VALUE										
LANGUAGE	OPTION #	35	64	91	92	93	94	96	123	124	125	126
USASCII	(standard)	#	@	[	\	]	^	`	{	¦	}	~
Swedish/Finnish	001	#	É	Ä	Ö	Å	ü	é	ä	ö	å	ü
Danish/Norwegian	002	#	@	Æ	Ø	Å	^	`	æ	ø	å	~
French	003	£	à	°	ç	§	^	`	é	ù	è	¨
German	004	£	§	Ä	Ö	ü	^	`	ä	ö	ü	ß
United Kingdom	005	£	@	[	\	]	^	`	{	¦	}	~
Spanish	006	#	@	¡	Ñ	¿	´	`	{	ñ	}	~

7-Bit Vs 8-Bit Operation

The terminal has two modes of operation that affect how characters received from datacomm are interpreted by the terminal. The modes are named for the number of significant bits they contain. In 8-bit mode all bits are significant; thus no bit is available for parity checking. In 7-bit mode, the seven low-order bits contain valid data. The eighth bit may be used for parity checking, or it may be ignored.

7-BIT MODE. When the terminal is configured for 7-bit mode, the least significant seven bits of the character byte determine the character's identity. That is, the seven bits are translated into the appropriate character according to the language selection in the Terminal Configuration menu. It is important to note that in 7-bit mode, the only accessible alphanumeric characters are those available from the selected language.

Example: While in 7-bit mode, if the terminal is configured for USASCII, the host computer and terminal can only recognize the standard ASCII characters.

In interpreting data transfers from the host computer, the terminal uses the least significant 7 bits. The designated character depends upon the current language selection.

Example: Refer to table C-1. Notice that when the terminal is in 7-bit mode if the host sends the decimal value 35 and is configured for either USASCII, Swedish/Finnish, Danish/Norwegian, or Spanish, the terminal interprets the character as the pound sign (#). If the host sends the same code, however, to a terminal configured for either French, German , or United Kingdom, the terminal interprets the code as "£".

To configure the terminal for 7-bit mode, set the ASCII 8 Bits field in the Terminal Configuration menu to "NO", or send the escape sequence:

ESC&k0I

NOTE

You or the host computer may always access the characters in the Line Drawing set if this set is chosen as the alternate character set. This is true regardless of the language chosen or the bit mode used.

8-BIT MODE. When the terminal is configured for 8-bit mode, the host and terminal can access any alphanumeric character without reconfiguring the terminal. For keyboard access, refer to the following section on Foreign Characters mode.

In interpreting data transfers from the host computer, the terminal uses all eight data bits. If the eighth bit is set (value = 1), the terminal interprets the code as a Roman Extension character. If the eighth bit is cleared (value = 0) the terminal interprets the code as appropriate for its language configuration. Tables C-2 and C-3 list the binary representation for both the ASCII and the Roman Extension character sets.

When using 8-bit mode, you must set the Parity field in the Datacomm Configuration menu to "None". Failure to do so will cause data communication problems with the host computer.

8-bit mode also changes the operation of the graphics/numeric keypad. When the terminal is in 8-bit mode and the keypad is defined for numeric operation, the numeric keys have shifted functions as shown in table C-4.

To configure the terminal for 8-bit mode, set the ASCII 8 Bits field in the Terminal Configuration menu to "YES", or send the escape sequence:

ESC&k1I

NOTE

The extended character set is used by the HP 300 and HP 250 computer systems and the HP 2631 and HP 2608 printers. Also, since the default parity used by the HP 3000 system is 000, you must log onto the computer using "Term Type 12" in order to receive all 8 data bits.

The following exercise summarizes several concepts presented thus far.

The task at hand is to write a Basic program to execute on the HP 3000. Heeding the previous note, you should log-on using terminal type 12. This alerts the computer that 8-bit binary data will be passed.

After you receive the system prompt (:), you enter Basic by typing "basic" then pressing the return key.

Next, type in the following Basic program and end each line by pressing the return key.

```
10 PRINT "Eight bit—with any language configuration: ";
20 PRINT '27"&k1I";
30 PRINT "Fran" + CHR$(181) + "ais"
40 PRINT '27"&k0I";
50 PRINT "Seven bit—using current language configuration: ";
60 PRINT "Fran" + CHR$(92) + "ais"
```

As 27 is the ASCII decimal value for the escape key, line 20 programmatically turns on 8-bit mode, while line 40 turns it off.

Running this program with the terminal's default configurations produces the results:

```
Eight bit—with any language configuration:
        Fran5ais
Seven bit—using current language configuration:
        Fran\ais
```

Table C-2. Standard ISO/ASCII Character Codes

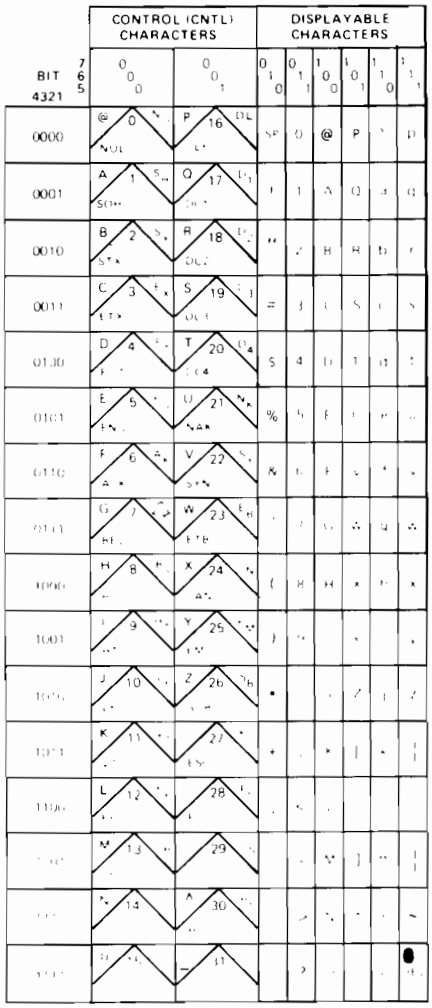


Table C-3. Extended Roman Character Codes

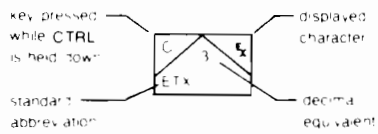
BIT 7 6 5 4 3 2 1	$B_8 = 1$ EXTENDED ROMAN CHARACTERS							
	0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1
0000				¯ / –	â	Â		
0001					ê	î		
0010					ô	Ø		
0011				•	û	Æ		
0100					á	å		
0101				Ç	é	í		
0110				Ñ	ó	ø		
0111				ñ	ú	æ		
1000			¸	¡	à	Ä		
1001			`	¿	è	ì		
1010			^	¤	ò	Ö		
1011			¨	£	ù	ü		
1100			~		ä	É		
1101				§	ë	ï		
1110					ö	ß		
1111			£		ü			

^K — ACKNOWLEDGE
^G — BELL
^H — BACKSPACE
^X — CANCEL LINE
^M — CARRIAGE RETURN
^P — DATA LINK ESCAPE
^Q — DEVICE CONTROL 1
^R — DEVICE CONTROL 2
^S — DEVICE CONTROL 3
^T — DEVICE CONTROL 4
■ — DELETE
^Y — END OF MEDIUM
^E — ENQUIRY
^D — END OF TRANSMISSION

^[— ESCAPE
^_ — END OF BLOCK
^C — END OF TEXT
^L — FORM FEED
^\ — FILE SEPARATOR
^] — GROUP SEPARATOR
^I — HORIZONTAL TAB
^J — LINE FEED
^N — NEGATIVE ACKNOWLEDGE
^^ — RECORD SEPARATOR
^O — SHIFT IN
^N — SHIFT OUT
SP — SPACE

^A — START OF HEADING
^B — START OF TEXT
^Z — SUBSTITUTE
^V — SYNCHRONOUS IDLE
^W — UNIT SEPARATOR
^K — VERTICAL TAB

Control Character Legend



The default value for the `Language` field is "`USASCII`". Since table C-1 shows USASCII decimal value 92 to be "\" and since 7-bit mode is in effect when line 60 executes, the second line of output should be no surprise. However, the contention was made that 8-bit mode works for any language configuration. Since both the computer and terminal were prepared for eight bit data, the first line of output is, initially, a mystery. Inspecting table C-5 for possible clues, you discover that CHR$(181) was probably interpreted as 53 (thus producing the "5") and not 181. This implies bit 8 was seen as a zero and not as a one. Why? Thinking back, you remember you made no changes to the Datacomm Configuration menu. Displaying this menu, you see that the default setting for the `Parity` field is "`0's`". As this setting instructs the terminal to interpret the eighth bit as a zero, it readily explains the noted discrepancy.

Setting the `Parity` field to "`None`", saving this new configuration, and rerunning the program produces this output:

Eight bit—with any language configuration:
Français
Seven bit—using current language configuration:
Fran\ais

The first part of the program now works as intended. Next, enter the Terminal Configuration menu and change the Language field to "FRANCAIS azM". Running the program for a final time produces the results:

Eight bit—with any language configuration:
Français
Seven bit—using current language configuration:
Français

Notice the following:

1. Using French has no bearing on 8-bit codes. The Roman Extension set maps into the same characters regardless of the chosen language.
2. Using French does affect the 7-bit ASCII codes. Whereas decimal value 92 maps into "\" in USASCII, it is interpreted as "ç" in French. In Spanish, it becomes "Ñ"; in German, "Ö"; and so on. (See table C-1).

Though simply, this sample program illustrated several key points. Most importantly, as the programmer, you must ensure that the host computer, terminal, and application program are in complete agreement concerning the format of passed data. As demonstrated, failure to do so may produce data communication problems.

Figure C-1 shows how a sample screen would look after completing the above exercise.

Table C-4. Shifted Functions of the Numeric Pad in 8-Bit Mode

KEY	SHIFTED CHARACTER
0	^
.	~
1	{
2	|
3	}
4	[
5	\
6	]
7	'
8	`
9	@
-	transforms pad to Graphics operation

```
hello brilf.bix; term=12
HP3000 / MPE IV C.P0.C4.      MON, SEP 23, 1982, 2:28 PM

                              BENJI

:basic

HP32101B.00.15(4WD)  BASIC  (C)HEWLETT-PACKARD CO 19/9
>10 PRINT ''Eight bit—with any language configuration: '';
>20 PRINT '27''&k1I'';
>30 PRINT ''Fran'' + CHR$(181) + ''ais''
>40 PRINT '27''&k0I'';
>50 PRINT ''Seven bit—using current language configuration: '';
>60 PRINT ''Fran'' + CHR$(92) + ''ais''

>run
Eight bit—with any language configuration: Fran5ais
Seven bit—using current language configuration: Fran\ais

>run
Eight bit—with any language configuration: Français
Seven bit—using current language configuration: Fran\ais

>run
Eight bit—with any language configuration: Français
Seven bit—using current language configuration: Français
```

Figure C-1. Sample Screen of Basic Program

Foreign Characters Mode

Foreign Characters mode grants full access to the extended Roman character set. Two critical points are:

1. You must configure the terminal for 8-bit mode (set the `ASCII 8Bits` field in the Terminal Configuration menu to "YES").
2. You can only enter Foreign Characters mode through the keyboard.

Using this mode, you may select any character foreign to your chosen language without reconfiguring the `Language` field of the Terminal Configuration menu. Table C-5 shows the correspondence between key caps on the USASCII keyboard and the extended characters they generate. Figures C-2 to C-8 illustrate the placement of "foreign" characters for each of the available keyboards.

From your keyboard, you enter Foreign Characters mode by simultaneously pressing the Control key and period

Table C-5. Generation of Extended Foreign Characters

7-bit Character Code			8-bit Character Code	
Decimal Value	**ASCII Character**	**USASCII Keyboard**	**Extended Graphic**	**Decimal Value**
40	Opening parenthesis	(	´	168
41	Closing parenthesis	)	`	169
42	Asterisk	*	^	170
43	Plus	+	¨	171
44	Comma	,	~	172
47	Slant	/	£	175
48	Zero	0	¯	176
51	Three	3	·	179
53	Five	5	Ç	181
54	Six	6	Ñ	182
55	Seven	7	ñ	183
56	Eight	8	¡	184
57	Nine	9	¿	185
58	Colon	:	¤	186
59	Semicolon	;	£	187
61	Equals	=	§	189
64	Commercial at	@	â	192
65	Uppercase A	A	ê	193
66	Uppercase B	B	ô	194
67	Uppercase C	C	û	195
68	Uppercase D	D	á	196
69	Uppercase E	E	é	197
70	Uppercase F	F	ó	198
71	Uppercase G	G	ú	199
72	Uppercase H	H	à	200
73	Uppercase I	I	è	201
74	Uppercase J	J	ò	202
75	Uppercase K	K	ù	203
76	Uppercase L	L	ä	204
77	Uppercase M	M	ë	205
78	Uppercase N	N	ö	206
79	Uppercase O	O	ü	207
80	Uppercase P	P	Å	208
81	Uppercase Q	Q	î	209
82	Uppercase R	R	Ø	210
83	Uppercase S	S	Æ	211
84	Uppercase T	T	å	212
85	Uppercase U	U	í	213
86	Uppercase V	V	ø	214
87	Uppercase W	W	æ	215
88	Uppercase X	X	Ä	216
89	Uppercase Y	Y	ì	217
90	Uppercase Z	Z	Ö	218
91	Opening bracket	[	ü	219
92	Reverse slant	\	É	220
93	Closing bracket	]	ï	221
94	Circumflex	^	ß	222

key. You leave Foreign Characters mode by simultaneously pressing the Control key and comma key. Unlike the alternate character sets which end when a new line is begun, Foreign Characters mode remains active until you explicitly turn it off using Control-Comma. (Foreign Characters mode also ends when the terminal is turned off or a hard reset is performed.)

Being a keyboard function, Foreign Characters mode only affects data entered from the keyboard. It has no effect on data received over the datacomm lines. However, the following special circumstances exist:

1. If you enter any control codes (simultaneously pressing the CTRL key and another appropriate key), the code is interpreted as if Foreign Characters mode were off.

2. The special procedure to form mute characters does not exist in Foreign Characters mode. (Rather, you may enter these characters directly by pressing the appropriate key on your keyboard.)

ALTERNATE CHARACTER SETS

The 2627A includes a 64 character line-drawing set as the standard alternate character set. This set is the active alternate character set at power on time and after a hard reset.

You may configure the alternate character set to be either the Line Drawing set or the base character set. (The base character set is the language selected in the Terminal Configuration menu.)

To select the base set as the alternate character set, you must issue the escape sequence:

Esc)@

The assignment of the base set as the alternate character set results in no character distinction between characters displayed when the alternate character set is enabled and when it is not.

To select the Line Drawing set as the alternate character set, you must use the escape sequence:

Esc)B

When the Line Drawing set is selected as the alternate character set, you may access these characters by a "shift-out". From the keyboard, you shift-out from the base set by simultaneously pressing the CTRL and N keys. An executing program may shift-out from the base set by issuing an ASCII ^N code. Once activated, all non-control characters received from the keyboard or over datacomm lines are displayed as line-drawing characters until one of the following conditions occurs:

1. the Base set is "shifted-in" (Control-O)
2. the end of the line is reached
3. a display enhancement (underline, inverse video, blinking, or "half-bright") is encountered

If the Line Drawing set is used in 8-bit mode, the terminal interprets it as a 256 element set. The lower 128 elements, for which 64 unique line-drawing symbols have already been defined, is accessed when Foreign Characters mode is off. (This is the default state.) The upper 128 elements remain undefined. Accessing these characters through the keyboard when Foreign Characters mode is on will print blank characters.

ASCII/EBCDIC CHARACTER CODES

Table C-6 summarizes the complete 128-code ASCII character set, table C-7 presents the decimal, octal, and hexadecimal codes for the ASCII character set, and table C-8 presents the decimal, octal, and hexadecimal codes for the EBCDIC character set.

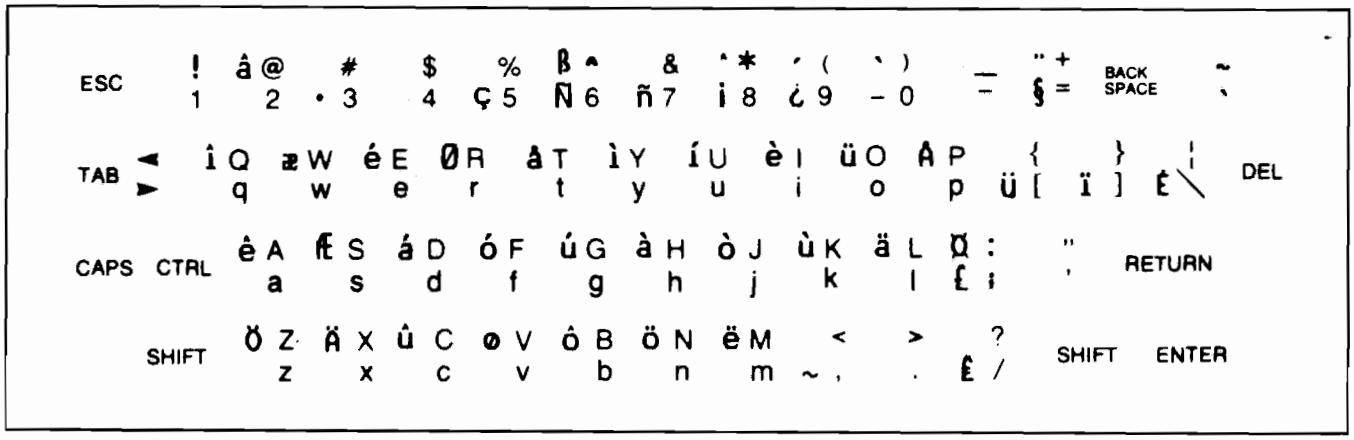


Figure C-2. Accessing Foreign Characters with USASCII Keyboard

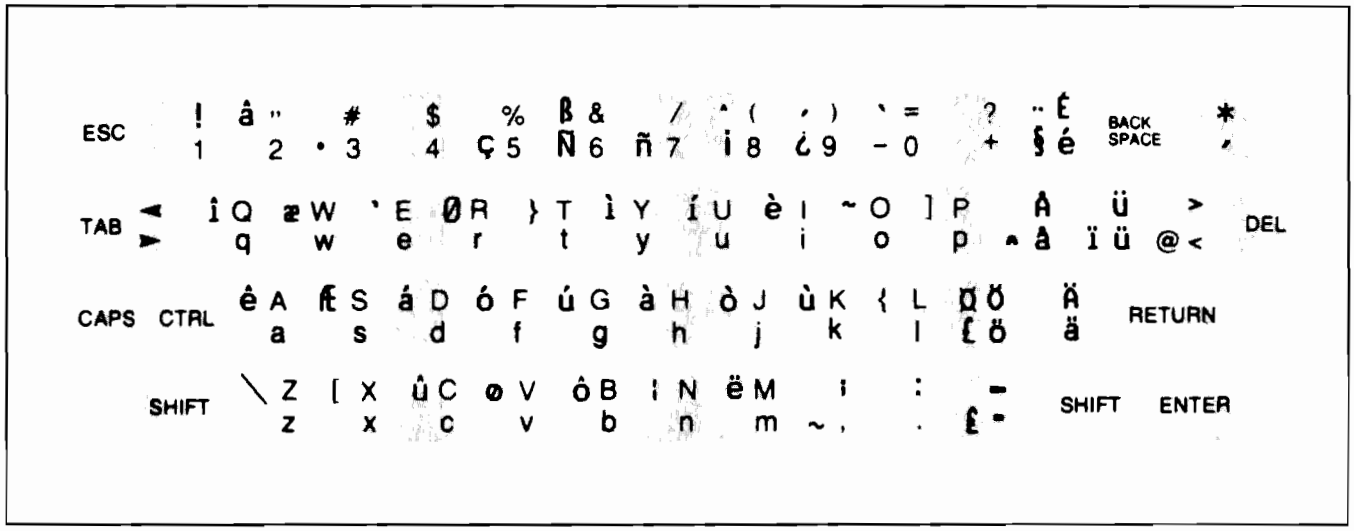


Figure C-3. Accessing Foreign Characters with Swedish/Finnish Keyboard

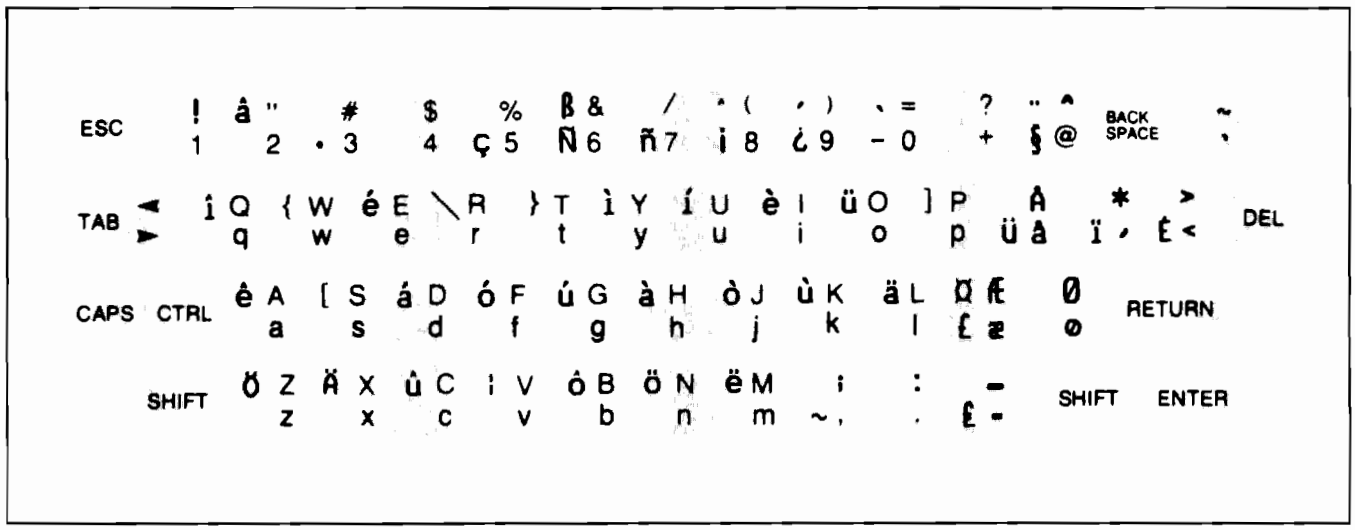


Figure C-4. Accessing Foreign Characters with Danish/Norwegian Keyboard

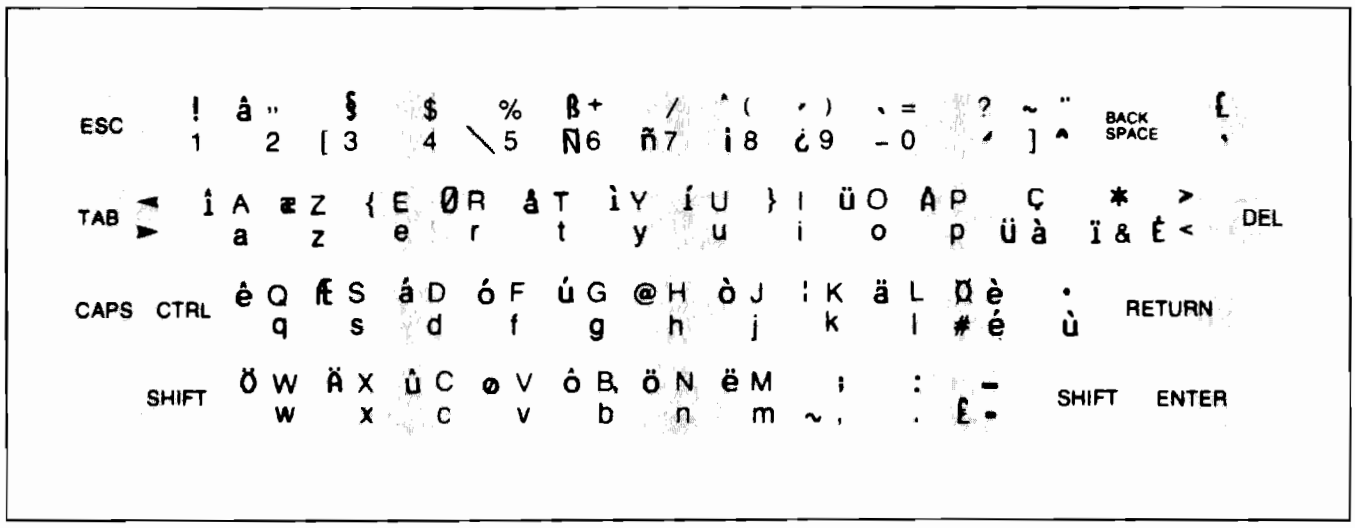


Figure C-5. Accessing Foreign Characters with French (AZERTY) Keyboard

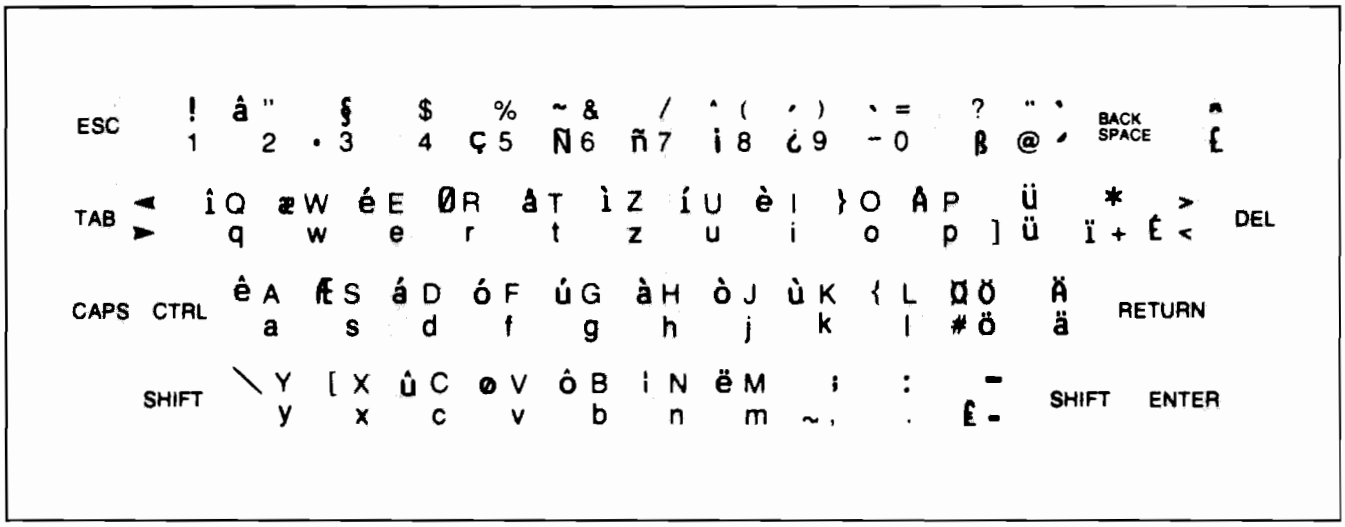


Figure C-6. Accessing Foreign Characters with German Keyboard

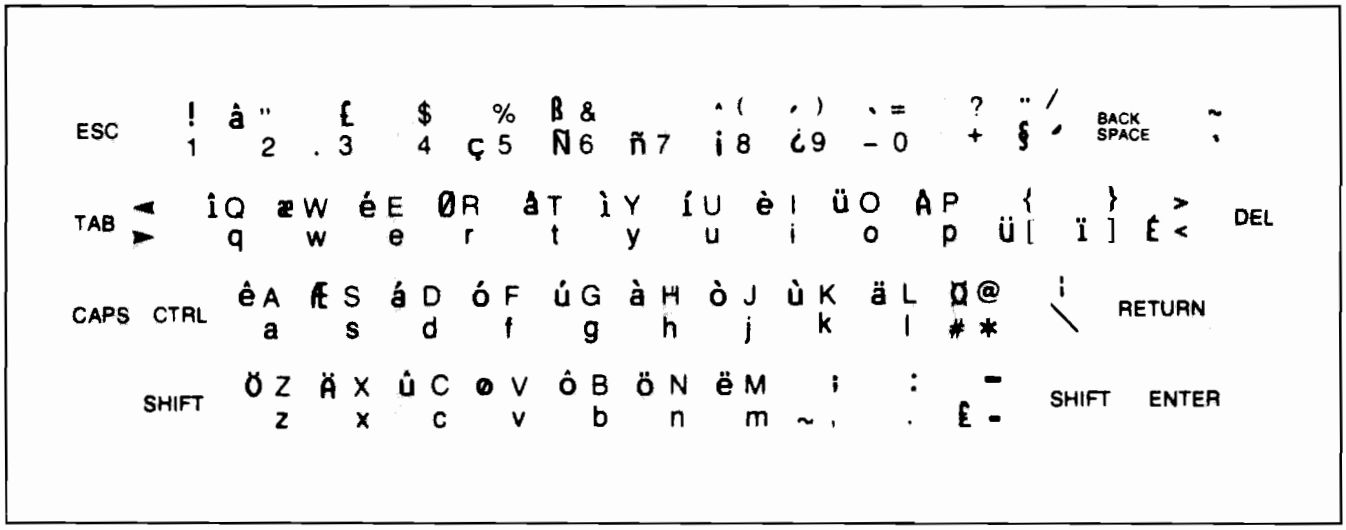


Figure C-7. Accessing Foreign Characters with United Kingdom Keyboard

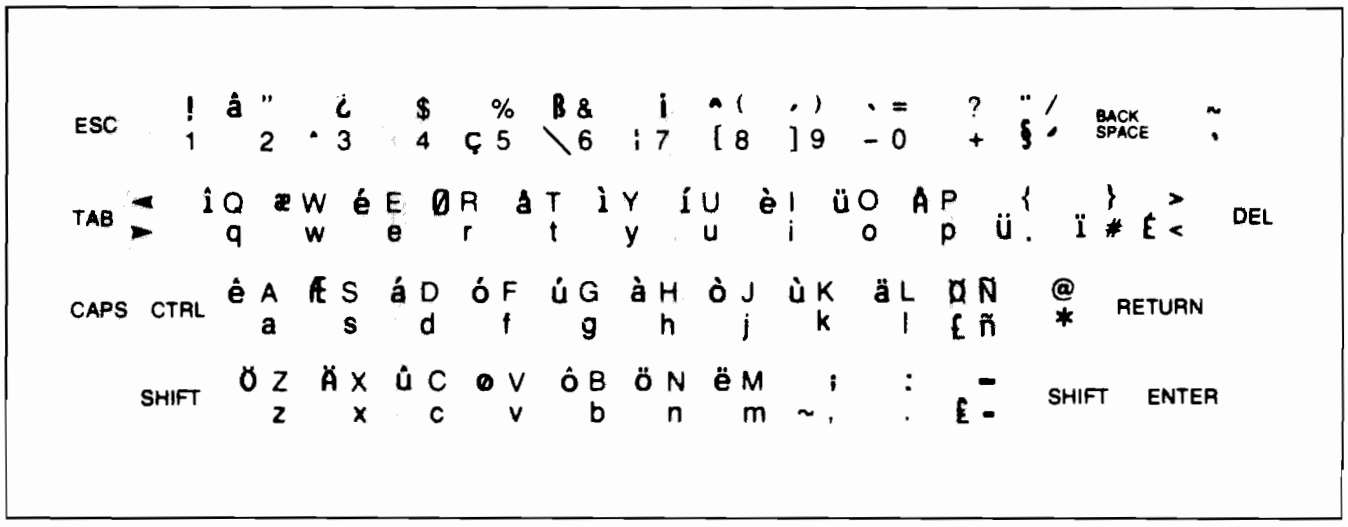


Figure C-8. Accessing Foreign Characters with Spanish Keyboard

Table C-6. ASCII Character Set

DECIMAL VALUE	GRAPHIC	COMMENTS	ALTERNATE CHARACTER	DECIMAL VALUE	GRAPHIC	COMMENTS
0	^@	Null	@^c	64	@	Commercial at
1	^A	Start of heading	A^c	65	A	Uppercase A
2	^B	Start of text	B^c	66	B	Uppercase B
3	^C	End of text	C^c	67	C	Uppercase C
4	^D	End of transmission	D^c	68	D	Uppercase D
5	^E	Enquiry	E^c	69	E	Uppercase E
6	^F	Acknowledge	F^c	70	F	Uppercase F
7	^G	Bell	G^c	71	G	Uppercase G
8	^H	Backspace	H^c	72	H	Uppercase H
9	^I	Horizontal tabulation	I^c	73	I	Uppercase I
10	^J	Line feed	J^c	74	J	Uppercase J
11	^K	Vertical tabulation	K^c	75	K	Uppercase K
12	^L	Form feed	L^c	76	L	Uppercase L
13	^M	Carriage return	M^c	77	M	Uppercase M
14	^N	Shift out	N^c	78	N	Uppercase N
15	^O	Shift in	O^c	79	O	Uppercase O
16	^P	Data link escape	P^c	80	P	Uppercase P
17	^Q	Device control 1 (X-ON)	Q^c	81	Q	Uppercase Q
18	^R	Device control 2	R^c	82	R	Uppercase R
19	^S	Device control 3 (X-OFF)	S^c	83	S	Uppercase S
20	^T	Device control 4	T^c	84	T	Uppercase T
21	^U	Negative acknowledge	U^c	85	U	Uppercase U
22	^V	Synchronous idle	V^c	86	V	Uppercase V
23	^W	End of transmission block	W^c	87	W	Uppercase W
24	^X	Cancel	X^c	88	X	Uppercase X
25	^Y	End of medium	Y^c	89	Y	Uppercase Y
26	^Z	Substitute	Z^c	90	Z	Uppercase Z
27	^[	Escape	[^c	[1] 91	[	Opening bracket
28	^\	File separator	\^c	[2] 92	\	Reverse slant
29	^]	Group separator	]^c	[1] 93	]	Closing bracket
30	^^	Record separator	^^c	[1] 94	^	Circumflex
31	^_	Unit separator	_^c	[2] 95	_	Underscore
32		Space (Blank)		96	`	Grave accent
[1]33	!	Exclamation point		97	a	Lowercase a
34	"	Quotation mark		98	b	Lowercase b
35	#	Number sign		99	c	Lowercase c
36	$	Dollar sign		100	d	Lowercase d
37	%	Percent sign		101	e	Lowercase e
38	&	Ampersand		102	f	Lowercase f
39	'	Apostrophe		103	g	Lowercase g
40	(	Opening parenthesis		104	h	Lowercase h
41	)	Closing parenthesis		105	i	Lowercase i
42	*	Asterisk		106	j	Lowercase j
43	+	Plus		107	k	Lowercase k
44	,	Comma		108	l	Lowercase l
45	-	Hyphen (Minus)		109	m	Lowercase m
46	.	Period (Decimal)		110	n	Lowercase n
47	/	Slant		111	o	Lowercase o
48	0	Zero		112	p	Lowercase p
49	1	One		113	q	Lowercase q
50	2	Two		114	r	Lowercase r
51	3	Three		115	s	Lowercase s
52	4	Four		116	t	Lowercase t
53	5	Five		117	u	Lowercase u
54	6	Six		118	v	Lowercase v
55	7	Seven		119	w	Lowercase w
56	8	Eight		120	x	Lowercase x
57	9	Nine		121	y	Lowercase y
58	:	Colon		122	z	Lowercase z
59	;	Semicolon		[2]123	{	Opening (left) brace
60	<	Less than		[2]124	|	Vertical line
61	=	Equals		[2]125	}	Closing (right) brace
62	>	Greater than		[2]126	~	Tilde
63	?	Question mark		127		Delete

Notes: 1. The equivalent EBCDIC character uses a different graphic.
2. No equivalent character exists in EBCDIC.

Table C-7. ASCII (7-Bit) character Codes

GRAPHIC	DEC	OCT	HEX
NUL	0	0	0
SOH	1	1	1
STX	2	2	2
ETX	3	3	3
EOT	4	4	4
ENQ	5	5	5
ACK	6	6	6
BEL	7	7	7
BS	8	10	8
HT	9	11	9
LF	10	12	A
VT	11	13	B
FF	12	14	C
CR	13	15	D
SO	14	16	E
SI	15	17	F
DLE	16	20	10
DC1	17	21	11
DC2	18	22	12
DC3	19	23	13
DC4	20	24	14
NAK	21	25	15
SYN	22	26	16
ETB	23	27	17
CAN	24	30	18
EM	25	31	19
SUB	26	32	1A
ESC	27	33	1B
FS	28	34	1C
GS	29	35	1D
RS	30	36	1E
US	31	37	1F
SP	32	40	20
!	33	41	21
"	34	42	22
#	35	43	23
$	36	44	24
%	37	45	25
&	38	46	26
'	39	47	27
(	40	50	28
)	41	51	29
*	42	52	2A
+	43	53	2B
,	44	54	2C
-	45	55	2D
.	46	56	2E
/	47	57	2F
0	48	60	30
1	49	61	31
2	50	62	32
3	51	63	33
4	52	64	34
5	53	65	35
6	54	66	36
7	55	67	37
8	56	70	38
9	57	71	39
:	58	72	3A
;	59	73	3B
<	60	74	3C
=	61	75	3D
>	62	76	3E
?	63	77	3F

GRAPHIC	DEC	OCT	HEX
@	64	100	40
A	65	101	41
B	66	102	42
C	67	103	43
D	68	104	44
E	69	105	45
F	70	106	46
G	71	107	47
H	72	110	48
I	73	111	49
J	74	112	4A
K	75	113	4B
L	76	114	4C
M	77	115	4D
N	78	116	4E
O	79	117	4F
P	80	120	50
Q	81	121	51
R	82	122	52
S	83	123	53
T	84	124	54
U	85	125	55
V	86	126	56
W	87	127	57
X	88	130	58
Y	89	131	59
Z	90	132	5A
[	91	133	5B
\	92	134	5C
]	93	135	5D
^	94	136	5E
_	95	137	5F
`	96	140	60
a	97	141	61
b	98	142	62
c	99	143	63
d	100	144	64
e	101	145	65
f	102	146	66
g	103	147	67
h	104	150	68
i	105	151	69
j	106	152	6A
k	107	153	6B
l	108	154	6C
m	109	155	6D
n	110	156	6E
o	111	157	6F
p	112	160	70
q	113	161	71
r	114	162	72
s	115	163	73
t	116	164	74
u	117	165	75
v	118	166	76
w	119	167	77
x	120	170	78
y	121	171	79
z	122	172	7A
{	123	173	7B
|	124	174	7C
}	125	175	7D
~	126	176	7E
■	127	177	7F

Table C-8. EBCDIC Character Codes

GRAPHIC	DEC	OCT	HEX
NUL	0	0	0
SOH	1	1	1
STX	2	2	2
ETX	3	3	3
PF	4	4	4
HT	5	5	5
	6	6	6
DEL	7	7	7
	8	10	8
	9	11	9
	10	12	A
VT	11	13	B
FF	12	14	C
CR	13	15	D
SO	14	16	E
SI	15	17	F
DLE	16	20	10
DC1	17	21	11
DC2	18	22	12
TM	19	23	13
RES	20	24	14
NL	21	25	15
BS	22	26	16
IL	23	27	17
CAN	24	30	18
EM	25	31	19
CC	26	32	1A
CU1	27	33	1B
IFS	28	34	1C
IGS	29	35	1D
IRS	30	36	1E
IUS	31	37	1F
DS	32	40	20
SOS	33	41	21
FS	34	42	22
	35	43	23
BYP	36	44	24
LF	37	45	25
ETB	38	46	26
ESC	39	47	27
	40	50	28
	41	51	29
SM	42	52	2A
CU2	43	53	2B
	44	54	2C
ENQ	45	55	2D
ACK	46	56	2E
BEL	47	57	2F
	48	60	30
	49	61	31
SYN	50	62	32
	51	63	33
PN	52	64	34
RS	53	65	35
UC	54	66	36
EOT	55	67	37
	56	70	38
	57	71	39
	58	72	3A
CU3	59	73	3B
DC4	60	74	3C
NAK	61	75	3D
	62	76	3E
SUB	63	77	3F

GRAPHIC	DEC	OCT	HEX
SP	64	100	40
	65	101	41
	66	102	42
	67	103	43
	68	104	44
	69	105	45
	70	106	46
	71	107	47
	72	110	48
	73	111	49
	74	112	4A
.	75	113	4B
<	76	114	4C
(	77	115	4D
+	78	116	4E
ББ	79	117	4F
&	80	120	50
	81	121	51
	82	122	52
	83	123	53
	84	124	54
	85	125	55
	86	126	56
	87	127	57
	88	130	58
	89	131	59
!	90	132	5A
$	91	133	5B
*	92	134	5C
)	93	135	5D
;	94	136	5E
¬	95	137	5F
-	96	140	60
/	97	141	61
	98	142	62
	99	143	63
	100	144	64
	101	145	65
	102	146	66
	103	147	67
	104	150	68
	105	151	69
¦	106	152	6A
,	107	153	6B
%	108	154	6C
_	109	155	6D
>	110	156	6E
?	111	157	6F
	112	160	70
	113	161	71
	114	162	72
	115	163	73
	116	164	74
	117	165	75
	118	166	76
	119	167	77
	120	170	78
	121	171	79
:	122	172	7A
#	123	173	7B
@	124	174	7C
'	125	175	7D
=	126	176	7E
"	127	177	7F

Table C-8. EBCDIC Character Codes (Continued)

GRAPHIC	DEC	OCT	HEX
	128	200	80
a	129	201	81
b	130	202	82
c	131	203	83
d	132	204	84
e	133	205	85
f	134	206	86
g	135	207	87
h	136	210	88
i	137	211	89
	138	212	8A
	139	213	8B
	140	214	8C
	141	215	8D
	142	216	8E
	143	217	8F
	144	220	90
j	145	221	91
k	146	222	92
l	147	223	93
m	148	224	94
n	149	225	95
o	150	226	96
p	151	227	97
q	152	230	98
r	153	231	99
	154	232	9A
	155	233	9B
	156	234	9C
	157	235	9D
	158	236	9E
	159	237	9F
	160	240	A0
~	161	241	A1
s	162	242	A2
t	163	243	A3
u	164	244	A4
v	165	245	A5
w	166	246	A6
x	167	247	A7
y	168	250	A8
z	169	251	A9
	170	252	AA
	171	253	AB
	172	254	AC
[	173	255	AD
	174	256	AE
	175	257	AF
	176	260	B0
	177	261	B1
	178	262	B2
	179	263	B3
	180	264	B4
	181	265	B5
	182	266	B6
	183	267	B7
	184	270	B8
	185	271	B9
	186	272	BA
	187	273	BB
	188	274	BC
]	189	275	BD
	190	276	BE
	191	277	BF

GRAPHIC	DEC	OCT	HEX
{	192	300	C0
A	193	301	C1
B	194	302	C2
C	195	303	C3
D	196	304	C4
E	197	305	C5
F	198	306	C6
G	199	307	C7
H	200	310	C8
I	201	311	C9
	202	312	CA
	203	313	CB
	204	314	CC
	205	315	CD
	206	316	CE
	207	317	CF
}	208	320	D0
J	209	321	D1
K	210	322	D2
L	211	323	D3
M	212	324	D4
N	213	325	D5
O	214	326	D6
P	215	327	D7
Q	216	330	D8
R	217	331	D9
	218	332	DA
	219	333	DB
	220	334	DC
	221	335	DD
	222	336	DE
	223	337	DF
\	224	340	E0
	225	341	E1
S	226	342	E2
T	227	343	E3
U	228	344	E4
V	229	345	E5
W	230	346	E6
X	231	347	E7
Y	232	350	E8
Z	233	351	E9
	234	352	EA
	235	353	EB
	236	354	EC
	237	355	ED
	238	356	EE
	239	357	EF
0	240	360	F0
1	241	361	F1
2	242	362	F2
3	243	363	F3
4	244	364	F4
5	245	365	F5
6	246	366	F6
7	247	367	F7
8	248	370	F8
9	249	371	F9
	250	372	FA
	251	373	FB
	252	374	FC
	253	375	FD
	254	376	FE
	255	377	FF

Programming Examples

APPENDIX D

This appendix presents several utility programs, written in BASIC, that you may find useful when implementing an applications program to drive the terminal.

FORMIO1

This program reads the content of display memory into the host computer and generates the sequence of PRINT statements necessary to recreate the data.

Figure D-1 shows a source listing of FORMIO1.

Once FORMIO1 is stored as a BASIC program file within the host computer, you use it as follows:

1. Switch the terminal to local mode, home the cursor, and clear the display.
2. Using the line drawing set, design the data entry form. When you are done, leave the cursor positioned at the start of a line anywhere BELOW the form.
3. Switch the terminal back to remote mode and, within the BASIC Interpreter, run FORMIO1.
4. FORMIO1 will ask you what starting line number and

```
>list
FORMIO1
   10 FILES *,*
   15 SYSTEM X1,"PURGE FDATA"
   20 SYSTEM X1,"BUILD FDATA;rec=-132,,f,ascii"
   30 SYSTEM X1,"FILE X=$stdin;rec=-256"
   40 ASSIGN "FDATA",1,A1
   50 ASSIGN "X",2,A1,WR
   60 DIM A$[255],A1$[6],C$[3]
   70 PRINT CTL(208),'27"F"'27"a";
   80 ENTER 255,X,A$
   90 CONVERT A$[8;3] TO R
   95 SYSTEM X1,'27";"
  100 PRINT "This program creates basic statements that define the"
  110 PRINT "FORM or other data in this terminal's memory.";LIN(3)
  120 INPUT "Starting statement number, increment  ?",A,B
  130 SYSTEM T9,'27+";"
  150 PRINT '27"h";
  160 PRINT #1;"scr";END
  170 FOR I=1 TO R
  180   PRINT '27"d";
  190   LINPUT #2;A$
  200   IF UPS$(A$[1,3])="RUN" THEN 500
  210   IF UPS$(A$[1,4])=">RUN" THEN 500
  220   CONVERT A TO A1$
  230   REM compensate for imbedded "  marks
  240   C=-4
  250   IF C+5>LEN(A$) THEN 310
  260   C1=POS(A$[C+5],'34)
  270   IF NOT C1 THEN 310
  280   C=C1+C+4
  290   A$=A$[1,C]+"'34"+'34+A$[C+1]
  300   GOTO 250
  310   REM spaces >=7 are converted to direct cursor addresses
  320   FOR C=1 TO LEN(A$)
  330     IF A$[C,C+6]="       " THEN DO
  340       FOR C1=C+7 TO LEN(A$)
  350         IF A$[C1,C1]<>" " OR LEN(A$)=C1 THEN DO
  360           CONVERT C1-C TO C$
  370           A$[C]='27"&a+"+DEB$(C$)+"C"+A$[C1]
  380           GOTO 310
  390         DOEND
  400       NEXT C1
  410     DOEND
  420   NEXT C
  430   REM output form record as a BASIC print statement
  440   PRINT #1;" "+A1$+" print ctl(208),&";END
  450   PRINT #1;'34+A$[1,LEN(A$) MIN 127];"&";END
  460   IF LEN(A$)<128 THEN PRINT #1;'34;END
  470   IF LEN(A$)>=128 THEN PRINT #1;A$[128]+'34;END
  480   A=A+B
  490 NEXT I
  500 PRINT '27"FNow type 'XEQ FDATA' then 'LIST'.";LIN(1)
  510 PRINT "These statements will reproduce your terminal's memory--"
  520 PRINT "modify, NAME, RENUM, and SAVE as you wish....."
  530 SYSTEM T9,'27+":"
  550 END
```

Figure D-1. FORMIO1 Source Listing

increment you wish used. For example, if you want the PRINT statements numbered starting with 10 and proceeding in increments of 10, type "10, 10" and then press RETURN.

5. FORMIO1 homes the cursor and reads each line until the cursor reaches the line containing the RUN command.
6. Enter "XEQ FDATA" and then press RETURN. FOMRIO1 is now replaced in the BASIC Interpreter work space by the PRINT statements which, when executed, will recreate the data that was read from display memory.

At this point you may do with the PRINT statements as you like (LIST them, RUN them, NAME and SAVE them, add more statements to them, and so forth).

LRGLINE

This program accepts characters entered through the keyboard and replaces them on the screen with large characters created by the line drawing set.

Figure D-2 shows a source listing of LRGLINE.

Once LRGLINE is stored as a BASIC program file within the host computer, you use it as follows:

1. Within the BASIC Interpreter, run LRGLINE.
2. Move the cursor with the space bar, then type the desired string of characters and then press RETURN. Starting at the cursor position, LRGLINE recreates the string using the line drawing set and then leaves the cursor positioned at the left margin in the third line below the large character string.
3. Repeat step #2 for as many character strings as desired.
4. To stop the program, enter a control-Y and then type "ABORT".

With the terminal in local mode, use the edit keys (insert character, delete character, insert line, and delete line) to reposition the characters as desired and then, in remote mode, use the FORMIO1 program to generate the PRINT statements necessary to recreate the large character string(s).

```
LRGLINE
   10 PRINT LIN(4)
   20 LINPUT L1$
   30 IF LEN(L1$)=0 THEN GOTO 150
   40 L1$=UPS$(L1$)
   50 J$=DEB$(L1$)
   60 IF LEN(J$)=0 THEN GOTO 150
   70 I=1
   80 IF L1$[I,I]<>" " THEN GOTO 110
   90 I=I+1
  100 GOTO 80
  110 PRINT CTL(208),'27"&a-1r";I-1;"C"+'27"4";
  120 L1$=DEB$(L1$)
  130 IF 6*LEN(L1$)>78-I THEN L1$=L1$[1,INT((78-I)/6)]
  140 GOSUB 9000
  150 PRINT CTL(208),'10'10'10'10'10'10'10'27"&a0C";
  160 PRINT CTL(208),'27"9";
  170 GOTO 20
 9000 REM SUBROUTINE TO PRINT STRING IN L1$ IN LINE DRAWING SET
 9010 REM USES VARIABLES L1$ THRU L4$,L1 THRU L3
 9020 DIM L1$[80],L2$[11],L3$[128,75],L4$[1]
 9025 INTEGER L4
 9030 IF UND(L1)=1 THEN DO
 9040   L3$[32]="      \      \      \      \      \"
 9050   L3$[33]="  c   \  c   \  c   \      \   x  \"
 9060   L3$[34]=" c c  \      \      \      \      \"
 9070   L3$[35]=" c c \zczcz\ c c \zczcz\ c c \"
 9100   L3$[39]="  c   \ "'27"&dBz"'27"&d@    \ '    \      \      \"
 9160   L3$[44]="      \      \      \  c   \ "'27"&dBz"'27"&d@   \"
 9180   L3$[46]="      \      \      \      \  c  \"
 9200   L3$[48]="x"'27"&dBzzz"'27"&d@x\c    c\c    c\c    c\"'27"&dBz"'27&
    "&d@xxx"'27"&dBz\"'27"&dBz\"
 9210   L3$[49]=" "'27"&dBz"'27"&d@c   \  c   \  c   \  c   \xxcxx\"
 9220   L3$[50]="c"'27"&dBzzz"'27"&d@c\     c\c"'27"&dBzzzz\c     \cxxxc\"
 9230   L3$[51]="c"'27"&dBzzz"'27"&d@c\     c\ "'27"&dBzzz"'27&
    "&d@c\     c\cxxxc\"
 9240   L3$[52]="c    c\c    c\"'27"&dBzzzz"'27"&d@c\     c\     c\"
 9250   L3$[53]="c"'27"&dBzzzz\c     \"'27"&dBzzzz"'27"&d@x\     c\cxxxc\"
 9260   L3$[54]="c"'27"&dBzzz"'27"&d@c\c     \c"'27"&dBzzz"'27&
    "&d@x\c    c\cxxxc\"
 9270   L3$[55]="c"'27"&dBzzz"'27"&d@c\     c\     c\     c\     c\"
 9280   L3$[56]="x"'27"&dBzzz"'27"&d@x\c    c\z"'27"&dBzzz"'27&
    "&d@z\c    c\"'27"&dBz"'27"&d@xxx"'27"&dBz\"
 9290   L3$[57]="x"'27"&dBzzz"'27"&d@x\c    c\"'27"&dBxzzz"'27&
    "&d@c\     c\     c\"
 9300   L3$[58]="      \  c   \      \  c   \      \"
 9310   L3$[59]="      \  c   \      \  c   \ "'27"&dBzx"'27"&d@   \"
 9360   L3$[66]="c"'27"&dBzzz"'27"&d@x\c    c\c"'27"&dBzzz"'27&
    "&d@z\c    c\cxxx"'27"&dBz\"
 9370   L3$[65]="x"'27"&dBzzz"'27"&d@x\c    c\cxxxc\c    c\c    c\"
```

Figure D-2. LRGLINE Source Listing

```
9380   L3$[67]="x"'27"&dBzzz"'27"&d@x\c    "'27"&dBx\c     \c    x\"'27&
    "&dBz"'27"&d@xxx"'27"&dBz\"
9390   L3$[68]="c"'27"&dBzzz"'27"&d@x\c    c\c    c\c    c\cxxx"'27"&dBz\"
9400   L3$[69]="c"'27"&dBzzzz\c     \c"'27"&dBzz"'27"&d@   \c     \cxxxx\"
9410   L3$[70]="c"'27"&dBzzzz\c     \cxxx \c     \c     \"
9420   L3$[71]="x"'27"&dBzzz"'27"&d@x\c    "'27"&dBx\c xxx\c    c\"'27&
    "&dBz"'27"&d@xxx"'27"&dBz\"
9430   L3$[72]="c    c\c    c\c"'27"&dBzzz"'27"&d@c\c    c\c    c\"
9440   L3$[73]="'27"&dBzz"'27"&d@c"'27"&dBzz\  "'27&
    "&d@c   \   c   \   c   \xxcxx\"
9450   L3$[74]="     c\     c\     c\x    c\"'27"&dBz"'27"&d@xxx"'27"&dBz\"
9460   L3$[75]="c    c\c   c"'27"&dBx\cccz \c   c \c   cc\"
9470   L3$[76]="c     \c     \c     \c     \cxxxx\"
9480   L3$[77]="cczcc\c c c\c "'27"&dBz"'27"&d@ c\c    c\c    c\"
9490   L3$[78]="c    c\cc   c\c c c\c   cc\c    c\"
9500   L3$[79]="x"'27"&dBzzz"'27"&d@x\c    c\c    c\c    c\"'27"&dBz"'27&
    "&d@xxx"'27"&dBz\"
9510   L3$[80]="c"'27"&dBzzz"'27"&d@x\c    c\c"'27"&dBzzz"'27&
    "&dBx\c     \c     \"
9520   L3$[81]="x"'27"&dBzzz"'27"&d@x\c    c\c    c\c "'27"&dBz"'27&
    "&d@zc\"'27"&dBz"'27"&d@xxx"'27"&dBz\"
9530   L3$[82]="c"'27"&dBzzz"'27"&d@x\c    c\cxxx"'27"&dBz\c   c \c   cc\"
9540   L3$[83]="x"'27"&dBzzz"'27"&d@x\c     \"'27"&dBz"'27&
    "&d@xxxz\     c\"'27"&dBz"'27"&d@xxx"'27"&dBz\"
9550   L3$[84]="'27"&dBzz"'27"&d@c"'27"&dBzz\  "'27&
    "&d@c   \   c   \   c   \   c   \"
9560   L3$[85]="c    c\c    c\c    c\c    c\"'27"&dBz"'27"&d@xxx"'27"&dBz\"
9570   L3$[86]="c    c\c    c\c    c\"'27"&dBx"'27"&d@c c"'27"&dBx\ "'27&
    "&dBx"'27"&d@c"'27"&dBx"'27"&d@ \"
9580   L3$[87]="c    c\c    c\c x c\c c c\cc"'27"&dBx"'27"&d@cc\"
9590   L3$[88]="c    c\ c c \   c   \ c c \c    c\"
9600   L3$[89]="c    c\c    c\"'27"&dBz"'27"&d@c c"'27"&dBz\   c   \   c   \"
9610   L3$[90]="'27"&dBzzzz"'27"&d@c\    c \   c   \ c    \cxxxx\"
9800 DOEND
9810 PRINT CTL(208);'27")B";
9820 FOR L1=1 TO LEN(L1$)
9830   PRINT CTL(208),'27"4";
9840   L3=1
9850   L2=NUM(L1$[L1,L1])
9851   IF POS(L3$[L2,1,75],"\")=0 THEN L2=32
9860   FOR I=1 TO 5
9870     L4=POS(L3$[L2,L3,75],"\")
9875     IF L4=0 THEN RETURN
9880     PRINT CTL(208);'14+L3$[L2,L3,L3+L4-2]+'15+'27"&d@ ";
9890     L3=L4+L3
9900     PRINT CTL(208);'15'13'10;
9910   NEXT I
9920   PRINT CTL(208),'27"&a-5r+6C"+'27"4";
9930 NEXT L1
9940 RETURN
```

Figure D-2. LRGLINE Source Listing (Continued)

LANDDEMO

Figure D-3 provides a listing of a BASIC program that shows a geological cross section of a typical oil well and its adjacent subterranean composition.

```
LANDDEMO
    5 PRINT '27"*e0B"
   10 PRINT '27"*dacfL"
   20 PRINT '27"*m2a3B"
   25 REM  &v01Sturn on dithered blue for filling top of ocean
   30 PRINT '27"*m0g6h0,.5,1v1w5x1Y"
   35 REM  &v02Sdata points for top of ocean
   40 PRINT '27"*pas 378,389 131,389 52,230 459,230Z"
   50 PRINT '27"*m0g6h0,.5,1v1w0x1Y"
   55 REM  &v05Sdata points for right ,and left side of ocean top water
   60 PRINT '27"*pas 459,230 459,190 286,190 286,270Z"
   70 PRINT '27"*pas 286,270 52,230 52,190 286,190Z"
   75 REM  &v05SBLACK LINE FOR SEPARATING TOP FROM SIDE OF WATER
   76 PRINT '27"*pa  459,230 286,270 52,230Z"
   79 REM  &v05Sdithered green-blue set
   80 PRINT '27"*m0g3h1,1,.5v1w5x1Y"
   85 REM  &v06Sdata points for water ocean middle right side
   90 PRINT '27"*pas 459,190 447,197 423,195 384,201 351,211 ";
   91 PRINT "343,211 326,202 314,205 306,209 301,215 296,217 ";
   92 PRINT "288,221 286,221 286,148 459,148Z"
   95 REM  &v06Sdata points for water ocean middle left side
  100 PRINT CTL(208),'27"*pas 286,221 282,222 275,220 260,216 ";
  101 PRINT "245,210 215,213 179,204 155,206 116,197 93,200 ";
  102 PRINT "52,195 52,193 52,148 286,148Z"
  109 REM  &v07Sdithered yellow for bottom of ocean floor
  110 PRINT '27"*m0g3h.5,.5,0v1w5x1Y"
  115 REM  &v06Sdata points for bottom of ocean right
  120 PRINT '27"*pas 459,148 454,154 430,163 425,161 419,154 ";
  121 PRINT "397,173 371,171 352,161 326,156 286,170 286,108 459,108Z"
  125 REM  &v06Sdata points for bottom of ocean left
  126 PRINT '27"*pas 286,171 268,165 252,168 215,172 189,165 ";
  127 PRINT "141,157 134,153 122,156 105,155 92,157 60,151 ";
  128 PRINT "52,148 52,108 286,108Z"
  135 REM  &v02Sdithered gold set
  140 PRINT '27"*m0g3h1,.5,0v1w5x1Y"
  145 REM &v03S data points for second level of land left
  150 PRINT '27"*pas 286,68 52,68 52,121 98,115 179,120 ";
  151 PRINT "233,137 247,131 270,135 286,147Z"
  152 REM &v03S data points for second level of land right
  155 PRINT '27"*pas 286,148 314,133 332,131 360,128 391,149 ";
  156 PRINT "397,150 439,129 459,124 459,68 286,68Z"
  158 REM  &v02Sset color to black
  159 PRINT '27"*m0g0h0,0,0v1w5x1Y"
  160 REM &v03S data points for oil
  161 PRINT '27"*pas 173,123 161,120 152,114 133,109 ";
  162 PRINT "89,103 84,99 83,93 86,79 100,76 337,71 ";
  163 PRINT "352,103 354,109 353,117 350,119 335,122 ";
  164 PRINT "317,114 301,116 287,130 271,113 251,109 ";
  165 PRINT "215,115 175,124Z"
  169 REM  &v01Sdithered brown for third level of land
  170 PRINT '27"*m0g3h1,.5,.5v1w5x1Y"
  175 REM  &v07Sdata points for third level of land left
  180 PRINT '27"*pas 286,102 254,83 239,78 166,91 79,74 52,71 ";
  181 PRINT "52,26 52,0 286,40Z"
  185 REM  &v07Sdata points for third level of land right
  190 PRINT '27"*pas 286,102 294,98 347,103 377,86 410,83 ";
  191 PRINT "414,90 444,81 459,85 459,0 286,40Z"
  230 REM  &v02Sset color to black
  235 REM PRINT '27"*m0g0h0,0,0v1w5x1Y"
  250 REM &v03S data points for oil
```

Figure D-3. LANDDEMO Source Listing

```
255 REM PRINT '27"*pas 157,125 162,132 166,134 172,136 ";
256 REM PRINT "187,140 196,136 211,115 215,95 243,101 ";
257 REM PRINT "252,99 254,95 253,91 244,85 191,28 117,28 ";
258 REM PRINT "130,81 147,102 155,111 152,116Z"
270 REM  &v01Sdithered yellow for fourth level of land
275 PRINT '27"*m0g3h1,0,.5v1w5x1Y"
280 REM  &v06Sdata points for fourth level of land left
285 PRINT '27"*pas 286,67 259,62 253,56 218,51 200,34 161,40 ";
286 PRINT "124,29 63,37 52,33 52,0 286,40Z"
290 REM  &v06Sdata points for fourth level of land right
295 PRINT '27"*pas 286,67 328,61 361,45 422,42 432,31 ";
296 PRINT "459,31 459,0 286,40Z"
300 REM  &v02Sset color to black
305 PRINT '27"*m0g0h0,0,0v1w5x1Y"
309 REM  &v02Sdraw corner where v meets
310 PRINT '27"*pas 286,270 286,40 287,40 287,270Z"
320 REM  &v05Sdata points for platform
325 PRINT '27"*pas 165,305 139,279 188,279 214,305Z"
330 REM  &v06Sset color for platform legs
331 PRINT '27"*m0g6h0,0,0v1w0x1Y"
335 REM  &v06Sdata points for legs of platform
336 PRINT '27"*pas 214,305 214,296 214,270 210,270 210,293 ";
337 PRINT "192,275 192,254 189,253 187,254 187,274 ";
338 PRINT "146,274 146,246 142,245 141,246 141,274 ";
339 PRINT "139,274 139,280 189,280Z"
340 REM PRINT '27"*pas 169,270 166,270 166,275 169,275Z"
345 REM  &v01Sset color for tower white
346 PRINT '27"*m0g7h1,1,1v1w0x1Y"
354 REM  &v01Sdata points for tower on platform
355 PRINT '27"*pas 164,289 181,289 191,295 184,333 ";
356 PRINT "181,333 173,333Z"
360 REM  &v03Soutline for tower struts
365 PRINT '27"*m0g0h0,0,0v1w0x1Y"
370 PRINT '27"*pas 182,290 182,333 183,333 183,290Z"
380 PRINT '27"*pa 166,296 183,296 190,301Z"
390 PRINT '27"*pa 188,307 183,305 168,305Z"
400 PRINT '27"*pa 169,313 183,313 187,315Z"
410 PRINT '27"*pa 186,321 183,320 171,320Z"
420 PRINT '27"*pa 171,326 183,326 185,327Z"
430 PRINT '27"*pa 174,331 182,331 182,333 173,333Z"
440 PRINT '27"*pas 176,331 171,289 172,289 177,331Z"
450 REM  &v01Sdrilling shaft
460 PRINT '27"*m0g6h0,0,0v1w0x1Y"
470 PRINT '27"*pas 175,274 181,274 181,121 175,121Z"
495 END
```

Figure D-3. LANDDEMO Source Listing (Continued)

APHADEMO

Figure D-4 provides a source listing of a BASIC program that shows a typical data entry form used for inventory control purposes.

```
APHADEMO
    5 PRINT '27"h"'27"J"
   10 PRINT '27"*ddelT"
   15 PRINT '27"&j@"
   16 PRINT '27"c"
   17 PRINT '27"h"
   20 REM set default color pairs
   30 REM
   40 PRINT CTL(208),'27"&v1a1b1c0I"'27"&v1a1I"'27"&v1b2I"'27"&v1a1b3I"&
      '27"&v1c4I"'27"&v1a1c5I"'27"&v1b1c6I"'27"&v1x1y7I"
   50 REM
   60 REM define green on black as color pair 1
   70 REM and blue on white as color pair 4.
   80 REM
   90 PRINT CTL(208),'27"&1bI"
  100 PRINT CTL(208),'27"&v1c1x1y1z4I"
  110 PRINT CTL(208),'27"&a+11C "
  115 PRINT '27"h";
  120 PRINT CTL(208),'27"&a+17C"'27"&v04S"'27&
      "&dBCOLOR MAKES DATA ENTRY FORMS MORE READABLE"
  130 PRINT CTL(208),""
  140 PRINT CTL(208),"  "'27"&v04S"'27")B"'14&
      "Q;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;&
;;;;;;;;;;;W"'15
  150 PRINT CTL(208),"  "'27"&v04S"'27")B"'14":"'27"&v02S"'27"&dB"'15&
      "HEWLETT-PACKARD / "'27"&dJDATA TERMINALS DIVISION "'27"&v04S"'27&
      "&d@"'27"&a+8C"'27"&v02S"'27"&dBOrder #"'27"&v01S"'27"[ 92355"'27&
      "]-"'27"[6412-2215  "'27"&v04S"'27"&d@"'27")B"'14":"
  160 PRINT CTL(208),"  "'27"&v04S"'27")B"'14&
      "1;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;3;3;;;;;;;;;;;;;;;;;;;;;;;;;;&
;;;;;;;;;;;2"'15
  170 PRINT CTL(208),"  "'27"&v04S"'27")B"'14":"'27"&v02S"'27"&dB"'15&
      "Bill to:"'27"&dN"'27"[ ERGONOMIC FORMS INC.      "'27"] "'27&
      "&v04S"'27"&d@"'27")B"'14": :"'27"&v02S"'27"&dB"'15"Ship to:"'27&
      "&dN"'27"[ HI-VOL DATA OPERATIONS    "'27"] "'27"&v04S"'27"&d@"'27&
      ")B"'14":"
  180 PRINT CTL(208),"  "'27"&v04S"'27")B"'14":"'27"&v02S"'27"&dB"'15&
      "Address:"'27"&dN"'27"[ 1235 BIRCHMEAD PL."'27"&a+8C"'27"] "'27&
      "&v04S"'27"&d@"'27")B"'14": :"'27"&v02S"'27"&dB"'15"Address:"'27&
      "&dN"'27"[ 3450 LONG BEACH DR. BLDG 3"'27"] "'27"&v04S"'27"&d@"'27&
      ")B"'14":"
  190 PRINT CTL(208),'27"&v01S  "'27"&v04S"'27")B"'14":"'27"&v02S"'27&
      "&dB"'15"City"'27"&dN"'27"[ SANTA CARLA  "'27"] "'27"&dBState"'27&
      "&dN"'27"[CA"'27"] "'27"&dBZip"'27"&dN"'27"[98089"'27"] "'27&
      "&v04S"'27"&d@"'27")B"'14": :"'27"&v02S"'27"&dB"'15"City"'27"&dN"&
      '27"[ PASAROBLES   "'27"] "'27"&dBState"'27"&dN"'27"[CA"'27"] "'27&
      "&dBZip"'27"&dN"'27"[97238"'27"] "'27"&v04S"'27"&d@"'27")B"'14":"
  200 PRINT CTL(208),"  "'27"&v04S"'27")B"'14&
      "1;;;;;;;;#;;;;;;;;;;#;;;;;#;;;;;;;;;;;4;4;;;;;;;;;;;;;#;;;;;;;;;;;;&
#;;;;;;;;;;2"'15
  210 PRINT CTL(208),'27"&v02S  "'27"&v04S"'27")B"'14":"'27"&v02S"'27&
      "&dB"'15" Item   "'27"&v04S"'27"&d@"'27")B"'14"."'27"&v02S"'27&
      "&dB"'15" Product  "'27"&v04S"'27"&d@"'27")B"'14"."'27"&v02S"'27&
      "&dB"'15" Qty "'27"&v04S"'27"&d@"'27")B"'14"."'27"&v02S"'27"&dB"&
      '15'27"&a+7CDescription"'27"&a+7C"'27"&v04S"'27"&d@"'27")B"'14"."&
      '27"&v02S"'27"&dB"'15"   Units $  "'27"&v04S"'27"&d@"'27")B"'14"."&
      '27"&v02S"'27"&dB"'15" Total $  "'27"&v04S"'27"&d@"'27")B"'14":"&
      '15
  220 PRINT CTL(208),"  "'27"&v04S"'27")B"'14&
      "1;;;;;;;;+;;;;;;;;;;+;;;;;+;;;;;;;;;;;;;;;;;;;;;;;;;;+;;;;;;;;;;;;&
+;;;;;;;;;;2"'15
  230 PRINT CTL(208),"  "'27"&v04S"'27")B"'14":"'27"&v02S"'15'27&
      "[ 0001  "'27"] "'27"&v04S"'27")B"'14"."'27"&v02S"'15'27&
      "[938927A  "'27"] "'27"&v04S"'27")B"'14"."'27"&v02S"'15'27"[10  "&
      '27"] "'27"&v04S"'27")B"'14"."'27"&v02S"'15'27"[FIELD UPGRADE KIT"&
      '27"&a+7C"'27"] "'27"&v04S"'27")B"'14"."'27"&v02S"'15'27&
      "[    1000.00"'27"] "'27"&v04S"'27")B"'14"."'27"&v02S"'15'27&
      "[10,000.00"'27"] "'27"&v04S"'27")B"'14":"'15
  240 PRINT CTL(208),"  "'27"&v04S"'27")B"'14":"'27"&v02S"'15'27&
      "[ 0002  "'27"] "'27"&v04S"'27")B"'14"."'27"&v02S"'15'27&
      "[13242G   "'27"] "'27"&v04S"'27")B"'14"."'27"&v02S"'15'27"[10  "&
      '27"] "'27"&v04S"'27")B"'14"."'27"&v02S"'15'27"[PRINTER CABLE"'27&
      "&a+11C"'27"] "'27"&v04S"'27")B"'14"."'27"&v02S"'15'27&
      "[      80.00"'27"] "'27"&v04S"'27")B"'14"."'27"&v02S"'15'27&
      "[   800.00"'27"] "'27"&v04S"'27")B"'14":"'15
```

Figure D-4. APHADEMO Source Listing

```
250 PRINT CTL(208),'27"&v02S  "'27"&v04S"'27")B"'14":"'15'27"["'27&
   "&a+7C"'27"] "'27")B"'14"."'15'27"["'27"&a+9C"'27"] "'27")B"'14"."&
   '15'27"[     "'27"] "'27")B"'14"."'15'27"["'27"&a+24C"'27"] "'27&
   ")B"'14"."'15'27"["'27"&a+11C"'27"] "'27")B"'14"."'15'27"["'27&
   "&a+9C"'27"] "'27")B"'14":"'15
260 FOR A=1 TO 4
270   PRINT CTL(208),'27"&v02S  "'27"&v04S"'27")B"'14":"'15'27"["'27&
   "&a+7C"'27"] "'27")B"'14"."'15'27"["'27"&a+9C"'27"] "'27")B"'14"."&
   '15'27"[     "'27"] "'27")B"'14"."'15'27"["'27"&a+24C"'27"] "'27&
   ")B"'14"."'15'27"["'27"&a+11C"'27"] "'27")B"'14"."'15'27"["'27&
   "&a+9C"'27"] "'27")B"'14":"'15
280 NEXT A
290 PRINT CTL(208),'27"&v02S  "'27"&v04S"'27")B"'14":"'15'27"["'27&
   "&a+7C"'27"] "'27")B"'14"."'15'27"["'27"&a+9C"'27"] "'27")B"'14"."&
   '15'27"[     "'27"] "'27")B"'14"."'15'27"["'27"&a+24C"'27"] "'27&
   ")B"'14"."'15'27"["'27"&a+11C"'27"] "'27")B"'14"."'15'27"["'27&
   "&a+9C"'27"] "'27")B"'14":"'15
300 PRINT CTL(208),'27"&v02S  "'27"&v04S"'27")B"'14&
   "199999999(9999999999(99999(9999999999999999999999999(9999999999999&
(99999999992"'15
310 PRINT CTL(208),"  "'27"&v04S"'27")B"'14":"'27"&v02S"'27"&dB"'15&
   "Entered by "'27"&dJ"'27"[ PAM CALOUBEAN      "'27"] "'27&
   "&v02SDate "'27"[09"'27"]/"'27"[01"'27"]/"'27"[82"'27&
   "] Total Price $ "'27"&dJ"'27"[ 10,800.00      "'27"] "'27"&v04S"&
   '27"&d@"'27")B"'14":"'15
320 PRINT CTL(208),'27"&v02S  "'27"&v04S"'27")B"'14&
   "A;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;&
;;;;;;;;;;;S"'15
330 PRINT CTL(208),""
340 PRINT CTL(208),"  "'27"&v01S"'27&
   "&dB >> RED FIELDS INDICATE ERROR << "
350 PRINT CTL(208),'27"h";
355 PRINT CTL(208),'27"b";
360 PRINT '27"h";
```

Figure D-4. APHADEMO Source Listing (Continued)

Handshaking Protocol

APPENDIX E

A handshake's primary purpose is to control the start and stop of data transfers between two locations. The HP 2627A recognizes three types of block-transfer handshakes:

Type 1: No Handshaking. The terminal responds immediately.

Type 2: DC1 Handshaking. Before responding, the terminal waits until the host sends a DC1 (ASCII decimal value 17).

Type 3: DC2 Handshaking. The terminal waits until the host sends a DC1. The terminal acknowledges with a DC2 (ASCII decimal value 18). If the `Line/Page` field of the terminal configuration menu is set to "`LINE`", the terminal also sends the host a ↵ (ASCII decimal value 13). After the host sends a second DC1, the terminal begins the data transfer.

A block transfer is any data transfer that involves multiple characters. Such transfers may be initiated by the ENTER key, an Esc d sequence (transmit a block of text), a transmit-only softkey, a status request, or by the ENTER or RETURN key when in Modify Mode.

The following table summarizes the types of handshaking for each of the above functions based upon the settings of the `InhHndShk(G)`, `Inh DC2(H)`, and the `Line/Page(D)` fields in the terminal configuration menu, and whether Block Mode (accessed by the modes selection keys) is enabled or disabled.

The following symbols are used:

- Y "YES" field setting. Used in `InhHndShk(G)` and `Inh DC2(H)`.
- N "NO" field setting. Used in `InhHndShk(G)` and `Inh DC2(H)`.
- L "LINE" field setting in the `Line/Page(D)` field.
- P "PAGE" field setting in the `Line/Page(D)` field.
- * Block Mode enabled.
- (blank) Block Mode disabled.
- 1 Type 1 handshake (No Handshake).
- 2 Type 2 handshake (DC1 Handshake).
- 3 Type 3 handshake (DC2 Handshake).

Table E-1. How Terminal Configuration Determines Handshake Protocol

"Strap" Settings				Terminal Function				
G	H	B	D	ENTER	Modify Mode	Status	Transmit Softkey	Esc d
N	N		L	1	1	2	2	2
N	N		P	1	1	2	2	2
N	N	*	L	3	X	2	2	2
N	N	*	P	3	X	2	3	2
N	Y		L	1	1	2	2	2
N	Y		P	1	1	2	2	2
N	Y	*	L	1	X	2	2	2
N	Y	*	P	1	X	2	1	2
Y	N		L	3	3	3	3	3
Y	N		P	3	3	3	3	3
Y	N	*	L	3	X	3	3	3
Y	N	*	P	3	X	3	3	3
Y	Y		L	1	1	1	1	1
Y	Y		P	1	1	1	1	1
Y	Y	*	L	1	X	1	1	1
Y	Y	*	P	1	X	1	1	1

Note: "X"—Modify Mode is ignored when Block Mode is enabled.

Index