2622A

Display Terminal

HEWLETT
PACKARD

National Character Set Option will be available at a later date.

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.

HEWLETT-PACKARD COMPANY
974 EAST ARQUES AVENUE
SUNNYVALE, CALIFORNIA, 94086

Printed in U.S.A. 3/81

Preface

The HP 2622A Display Terminal is a versatile character, line, or block mode CRT terminal.

This reference manual contains detailed information for the HP 2622A which will allow you to configure, test, and use the terminal.

For ease of use, the manual is divided into nine sections and three appendices, as follows:

Section I	General Information
Section II	Configuring the Terminal
Section III	Keyboard Control
Section IV	Display Control
Section V	Printer Control
Section VI	Data Communications
Section VII	Status
Section VIII	Error Messages and Self-Test
Section IX	Terminal Maintenance Procedures
Appendix A	Character Sets
Appendix B	Escape Codes
Appendix C	Programming Examples

HP Computer Museum
www.hpmuseum.net

For research and education purposes only.

Table of Contents

Illustrations

Tables

General Description

SECTION I

INTRODUCTION

The HP 2622A Display Terminal (figure 1-1) is a versatile general purpose CRT terminal that offers the following features:

- CRT screen size of 150 mm (6 inches) by 215 mm (8.5 inches).
- Screen capacity of 24 lines of up to 80 characters each (1920 characters total). Two additional lines for system soft key labels.
- Display enhancements (inverse video, blinking, underlining, and half-bright).
- Character, line, or block mode operation.
- Eight user-programmable function keys (f1 through f8) are provided. The function of these keys is displayed in two rows of labels across the bottom of the display screen. The labels are programmable (up to 16 characters each). The character strings returned when the keys are pressed can be programmed with up to 80-character strings.
- All terminal configuration operations are performed through keyboard entries into formatted menus displayed on the screen, there are no physical straps. (Configuration data is maintained in non-volatile memory.)
- Two-character user-programmable RETURN key.
- Optional line drawing character set.
- Screen labeled system function keys (for selecting operating modes and performing other terminal control functions).

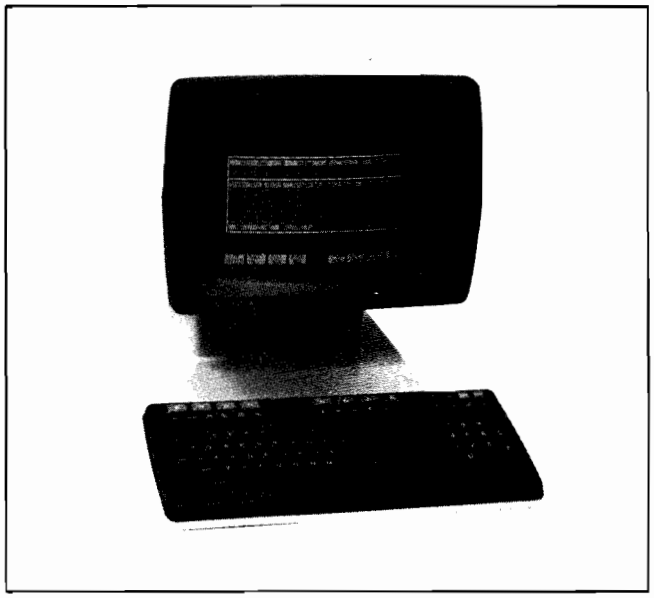


Figure 1-1. HP 2622A Display Terminal

- Full editing capabilities (insert/delete/clear line and insert/delete character).
- Adjustable margins and tab stops.
- Programmatic cursor sensing and addressing.
- Easy-to-use keyboard with separate numeric key pad.
- Extensive self-test capability.

Table 1-1 lists the specifications for the HP 2622A.

Table 1-1. Specifications

GENERAL	
Screen Size:	150 mm (6 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 4.24 mm (0.167 inch)
Character Set:	128 ASCII character set standard
Cursor:	Blinking-underline

Table 1-1. Specifications (Continued)

Display Enhancements:	Inverse, underline, blinking, half-bright
Display Modes:	White on black
Refresh Rate:	60 Hz (50 Hz optional)
Memory:	20 K-byte ROM; 3840 bytes (2 pages) display memory; 256 bytes of RAM for data communications buffer; 128 bytes of non-volatile RAM (battery backup provided)
Keyboard:	Detached, with 1.2 m (4 feet) cable. Full ASCII keyboard; 8 screen labeled keys; auto-repeat; N-key rollover. Numeric keyboard and independent user and cursor keys.
Operating Modes:	Remote; Local; Character, Line, 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
Data Rates:	110, 134.5, 150, 300, 600, 1200, 1800, 2400, 4800, 9600 baud. Integral printer—120 cps maximum.
Parity:	Selectable: Even, odd, zero, one, none.
PHYSICAL CHARACTERISTICS	
Weight:	Display Monitor: 16.8 kg (37 pounds) standard 19.0 kg (42 pounds) with integral printer Keyboard: 2.0 kg (4.4 pounds)
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:	115 V (+10%, −25%) at 50/60 Hz (±5%) 230 V (+10%, −15%) at 50 Hz (±5%)
Power Consumption:	75 watts (standard) 130 watts (with integral printer)

Table 1-1. Specifications (Continued)

OPTIONS	
-001	Swedish/Finnish Keyboard and ROM
-002	Danish/Norwegian Keyboard and ROM
-003	French Keyboard and ROM
-004	German Keyboard and ROM
-005	United Kingdom Keyboard and ROM
-006	Spanish Keyboard and ROM
-013	50 Hz, 240 V Power
-014	60 Hz, 100 V Power
-015	50 Hz, 230 V
-016	50 Hz, 115 V
-050	Integral Printer
-201	Line Drawing Character Set

KEYBOARD

The keyboard of the terminal (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.

Numeric Pad Group—The numeric group of keys is located to the right of the alphanumeric keys. 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.

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).

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.

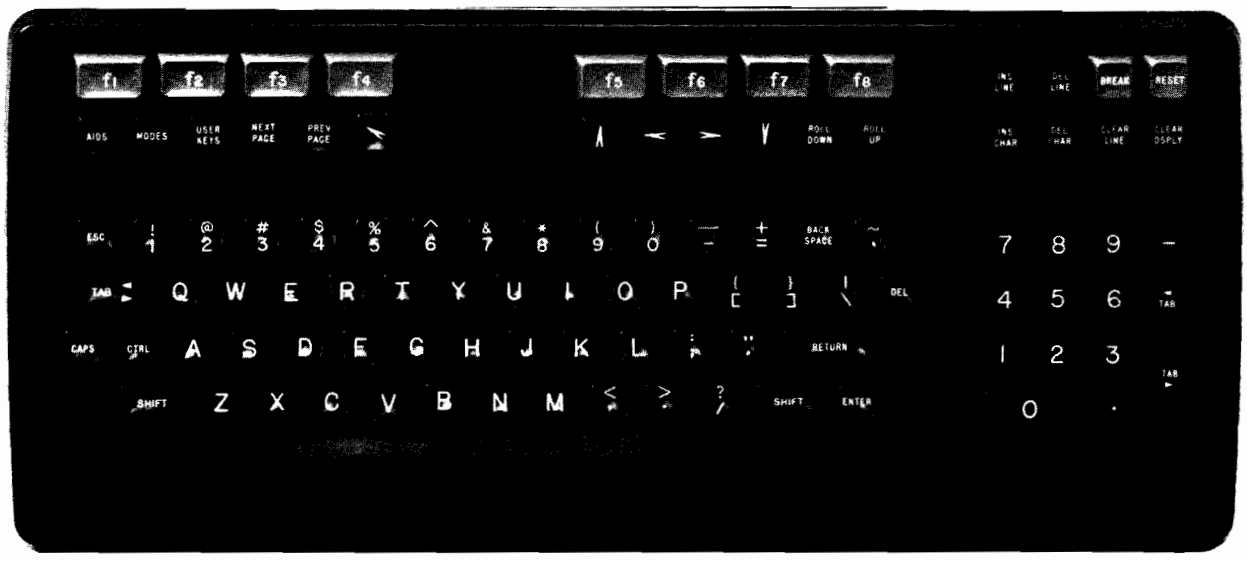


Figure 1-2. HP 2622A Keyboard

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 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 is illustrated in figure 1-5. Pressing AIDS always reinitializes f1 through f8 to the top row of functions (labels) shown in figure 1-5.

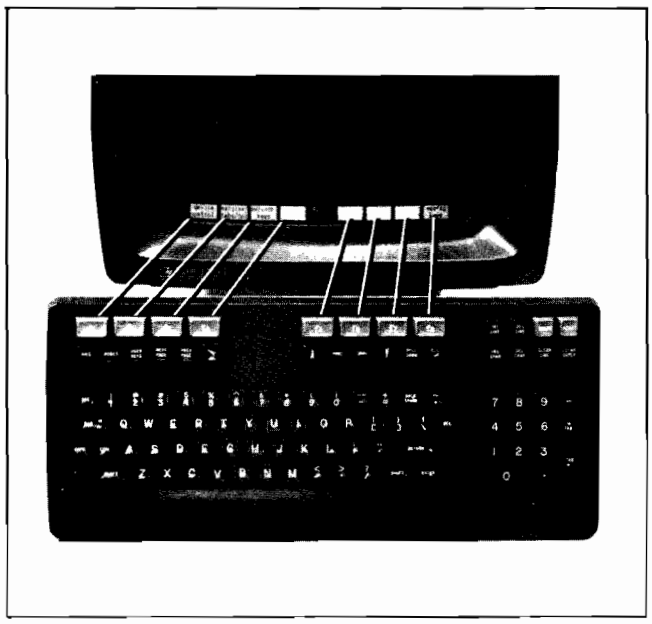


Figure 1-3. Function Keys and Screen Labels

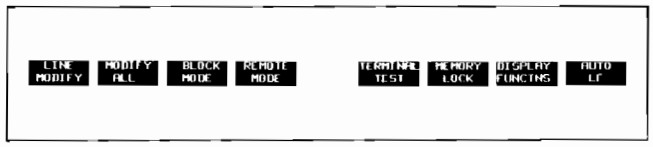


Figure 1-4. Mode Selection Key Labels

	[f1]	[f2]	[f3]	[f4]	[f5]	[f6]	[f7]	[f8]
MODES	LINE MODIFY*	MODIFY ALL *	BLOCK MODE *	REMOTE MODE *	TERMINAL TEST	MEMORY LOCK *	DISPLAY FUNCTNS*	AUTO LF *
AIDS	device control	margins/ tabs/col	service keys					config keys
DEVICE CONTROL	device modes		TO INT PRT*	ADVANCE PAGE	ADVANCE LINE	COPY ALL	COPY PAGE	COPY LINE
DEVICE MODES	device control	TO DISPLAY*	LOG BOTTOM*	LOG TOP *	EXPAND PRINT *	COMPRESS PRINT *	REPORT PRINT *	METRIC PRINT *
MARGINS TABS/COL	START COLUMN	SET TAB	CLEAR TAB	CLR ALL TABS	LEFT MARGIN	RIGHT MARGIN	CLR ALL MARGINS	
CONFIG			datacomm config		terminal config			
CONFIG	SAVE CONFIG	NEXT CHOICE	PREVIOUS CHOICE	DEFAULT VALUES			DISPLAY FUNCTNS*	config keys
SHIFT USER KEYS		NEXT CHOICE	PREVIOUS CHOICE	DEFAULT VALUES			DISPLAY FUNCTNS*	
SERVICE					TERMINAL TEST	IDENTIFY ROMS	DATACOMM TEST	INT PRT TEST

NOTE

Those labels which include an asterisk (*) indicate functions which, when enabled, include the asterisk to indicate that they are enabled.

Figure 1-5. HP 2622A Function Key Hierarchy

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.

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.

Pressing the SHIFT and SIDS keys disables the function keys and removes the key labels from the screen. To re-enable the keys, press SIDS, MODES, or USER KEYS.

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

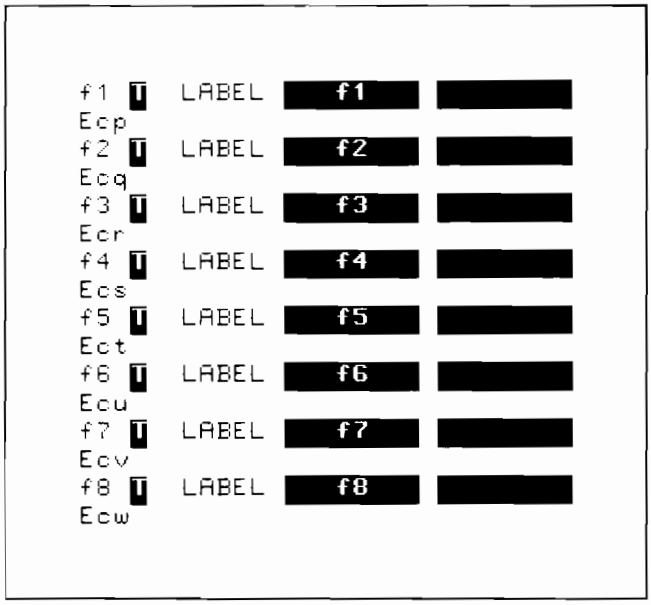


Figure 1-6. User Keys Definition Menu

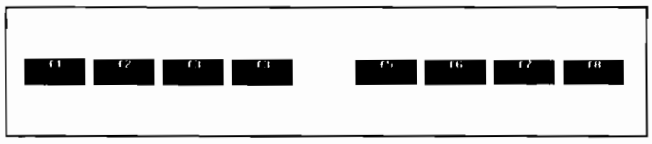


Figure 1-7. Default User Key Labels

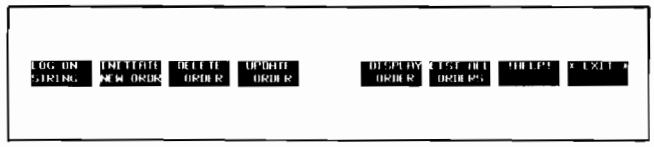


Figure 1-8. Sample User-Supplied User Key Labels

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.

INTEGRAL PRINTER

The optional integral printer is a fast, quiet, bidirectional thermal printing unit that can be used for generating reproducible copy from the terminal's display memory or from a remote computer.

The integral printer can reproduce any character capable of being displayed on the terminal's screen, including the line drawing character set.

Using either the system function keys or escape sequences, you can select the integral 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.

In addition, the integral printer offers both report mode and metric mode. When report mode is enabled, the integral printer output is formatted as a series of 8½ inch X 11 inch pages with a top and bottom margin and a tic mark between successive pages. Metric mode produces similar output except that the pages are longer (report mode generates 60 lines of text per page while metric mode generates 64 lines of text per page).

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 integral printer. With bottom logging, any data added to display memory, either from the keyboard or from a datacomm port, is also directed to the printer.

Refer to Section V for additional information on the integral printer.

DATA COMMUNICATIONS

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)

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.

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

Configuring the Terminal

SECTION II

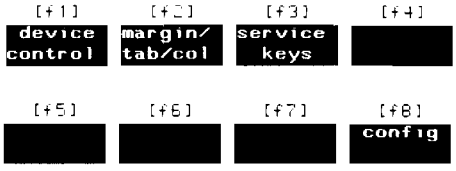
INTRODUCTION

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. Thus altered, the terminal's configuration characteristics are stored in non-volatile memory.

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

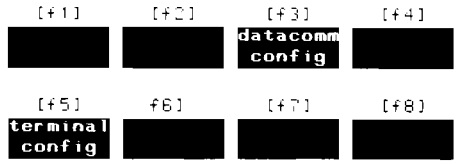
To gain access to the configuration menus through the keyboard, press the SHIFT key. This causes the following label display across the bottom of the screen (f1 , f2 , etc., refer to the function key corresponding to the label):



NOTE

The device control softkey is present only on terminals equipped with the optional integral printer.

Pressing `config. keys` changes the label display to



Pressing `datacomm config` (f3) 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 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 appear 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 had configured it.

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.

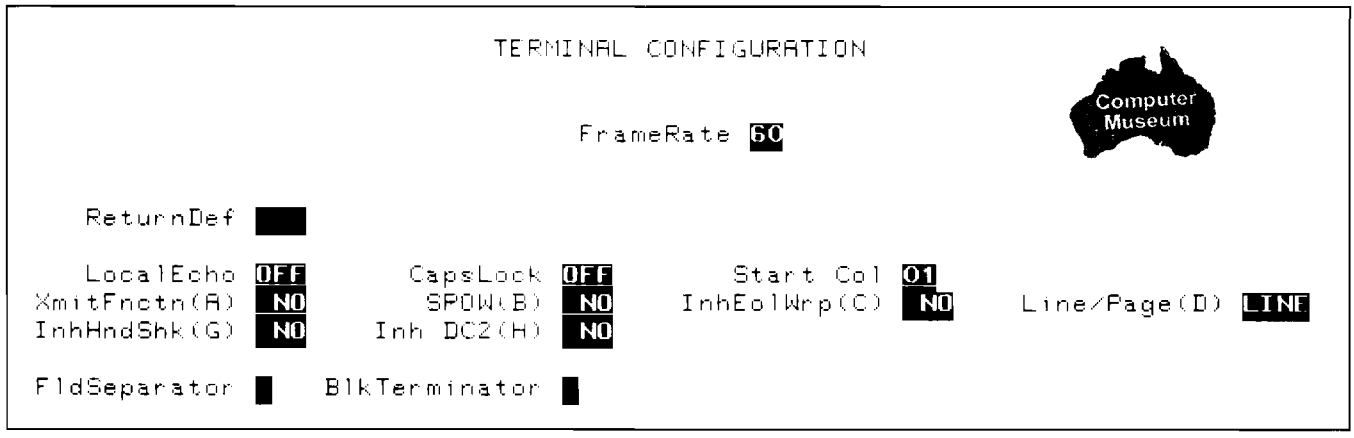


Figure 2-1. Terminal Configuration Menu

The menu contains a set of unprotected fields that you access using the TAB▶ and ◀TAB 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

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: Esc&k 1J = 50 Hz Esc&k 0J = 60 Hz Default: 60
RETURN def	This field specifies the definition of the RETURN key. The default definition is an ASCII CR. The definition may consist of up to two characters. If the second character is a space, it is ignored. Default: CR
LocalEcho	This field specifies whether characters entered through the keyboard are both displayed on the screen and transmitted to the host computer. ON (Esc&k 1L) Characters entered through the keyboard are both displayed on the screen and transmitted to the host computer. OFF (Esc&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 (Esc&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 (Esc&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. When you are operating in MODIFY LINE or MODIFYALL mode and you press ENTER or RETURN, the data transmission from the terminal normally begins at the logical start-of-text pointer in the

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

	particular line. If the line has no logical start-of-text pointer, however, the data transmission begins at the designated start column. This designated start 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 using one of the "`margin/tab/col`" function keys. Values: 1 - 80 Default: 1
`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. Note that display functions will emit Esc Z, Esc Y to a 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 turned 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
`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.

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

	Page (Stats 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 VI of this manual. Default: LINE
InhHndShk(G) and Inh ACK(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. 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, 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 ENQ; terminal transmits block of data. 3. Computer sends ENQ; terminal responds with ACK; computer responds with another ENQ; terminal transmits block of data. NOTE When the ENQ ACK ENQ handshake is enabled and the line/page field of terminal config is selected to be "line", a CR or CR LF is transmitted after the ACK. If "line" is not selected, nothing is transmitted after ACK. In general, the InhHndShk(G) and Inh ACK(H) fields have the following effects: InhHndShk(G) = YES — Eliminates the use of the ENQ handshake (terminal will either use the ENQ/ACK/ENQ handshake or no handshake at all). Inh ACK(H) = YES — Eliminates the use of the ENQ/ACK/ENQ handshake (terminal will either use the ENQ 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 ACK(H) fields. If your terminal is connected to a Hewlett-Packard computer system, you will find that the default settings for these fields (both OFF) 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 ACK(H) fields:

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

	1. ENTER key in block mode; or ENTER or RETURN key in modify mode; or Transmit only (T) user key in block page mode. InhHndShk(G) (ignored) Inh$_2$(H) = NO → $P_1/P_2/P_1$ Inh$_2$(H) = YES → no handshake 2. ENTER key in character mode. InhHndShk(G) = YES Inh$_2$(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 or device status request; or Device control completion code. InhHndShk(G) = NO → P_1 InhHndShk(G) = YES Inh$_2$(H) = NO → $P_1/P_2/P_1$ InhHndShk(G) = YES Inh$_2$(H) = YES → no handshake The G and H selections can be temporarily selected by the following escape sequences: Esc &s 0G disables G Esc &s 0H disables H Esc &s 1G enables G Esc &s 1H enables H Defaults: InhHndShk(G) = NO Inh$_2$(H) = NO
FldSeparator	When you press the ENTER key while the terminal is in block page mode and display memory contains a formatted display, the terminal automatically transmits the specified field separator character at the end of each protected 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: ↵

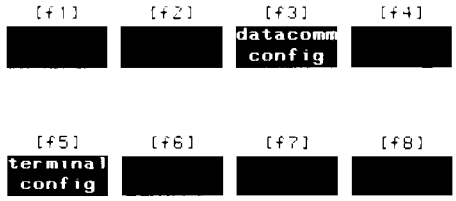
Note that as you alter the fields of a configuration menu on the screen, the selected values do NOT alter the content of non-volatile memory nor do they have any effect on the operation of the terminal.

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, f5, f6, f7, and f8 function keys perform as shown in table 2-2.

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] DISPLAY FUNCTNS	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 DISPLAY FUNCTNS 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: 

Lock/Unlock Configuration Menus

Using an escape sequence, you can "lock" the current configuration menus (terminal 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 (f7), BLOCK MODE (f3), REMOTE MODE (f4), 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	
SPOW Latch	
XmitFnctn(A)	Ec&s
SPOW(B)	
InhEolWrp(C)	
Line/Page(D)	
InhHndShk(G)	
Inh DC2(H)	

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.

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

FrameRate = 60:	Ec&k 0J
FrameRate = 50:	Ec&k 1J
LocalEcho = OFF:	Ec&k 0L
LocalEcho = ON:	Ec&k 1L
Caps Lock = OFF:	Ec&k 0C
Caps Lock = ON:	Ec&k 1C
SPOW Latch:	Ec&k 0N
	Ec&k 1N

You may combine these and other Ec&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:

Ec&k 1l 1C
Ec&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:	Ec&s 0A
XmitFnctn(A) = YES:	Ec&s 1A
SPOW(B) = NO:	Ec&s 0B
SPOW(B) = YES:	Ec&s 1B
InhEolWrp(C) = NO:	Ec&s 0C
InhEolWrp(C) = YES:	Ec&s 1C
Line/Page(D) = LINE:	Ec&s 0D
Line/Page(D) = PAGE:	Ec&s 1D
InhHndShk(G) = NO:	Ec&s 0G
InhHndShk(G) = YES:	Ec&s 1G
Inh%(H) = NO:	Ec&s 0H
Inh%(H) = YES:	Ec&s 1H

You may combine these and other Ec&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%(H) = YES, you could use any of the following escape sequences:

Ec&s 1d 0g 1H
Ec&s 0g 1h 1D
Ec&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 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).

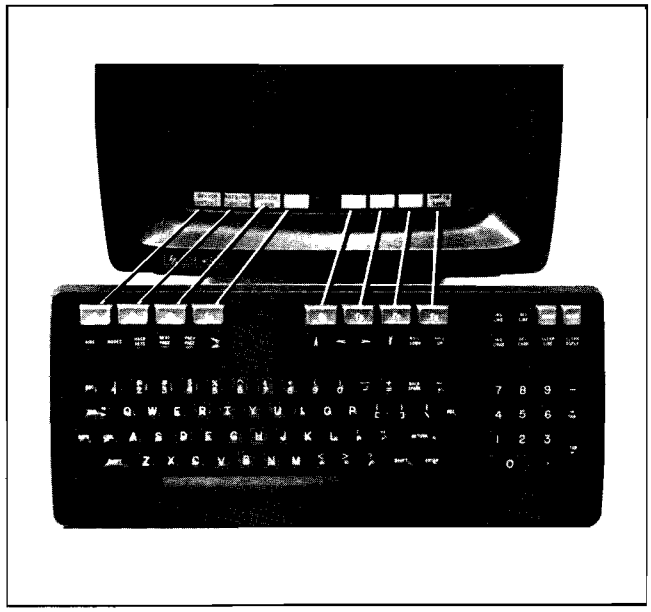


Figure 3-1. Screen-Labeled Function Keys

SELECTING MODES

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

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.

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).

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&k0R
Remote: Esc&k1R

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 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 ^K 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&k 1B
DISABLE: Esc&k 0B

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 key. Line modify mode is automatically disabled 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.

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

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

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

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

When using modify mode you will usually want the data block (NOT a handshake control code) to be sent when you press ENTER or RETURN. The default configuration parameters, however, enable the $P_1/A_2/P_1$ handshake. Therefore, in most cases you will first need to disable the $P_1/A_2/P_1$ handshake before using modify mode. You do so by setting the Inh DC2(H) field in the terminal configuration menu to "YES", or by sending Esc&s 1H. The modify all and modify line functions are ignored if the terminal is in format or block mode.

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 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 "AutoLF" field in both active and non-volatile memory. When you enable or disable auto line feed mode using the escape sequence, however, you only change the content of the "AutoLF" field in active memory.

After a hard reset or turning off the power, the terminal reverts to the mode specified by the "AutoLF" 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 V) 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 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. Position the cursor in the first line of paragraph 1.
c. Enable memory lock mode.
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 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 but the total number of locked rows does not increase.

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 but is not transmitted or displayed.
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 DISPLAY FUNCTNS 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 configuration menu. If XmitFnctn(A) is on, the DISPLAY FUNCTNS key transmits ESC Y, ESC Z. Note that, as a result of this, the receiver of the transmission can never exit display functions.

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 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.

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 ASCII alphanumeric characters and/or control codes (such as ^R or ^F).

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.

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 displayable 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. 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► and TAB◄ 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

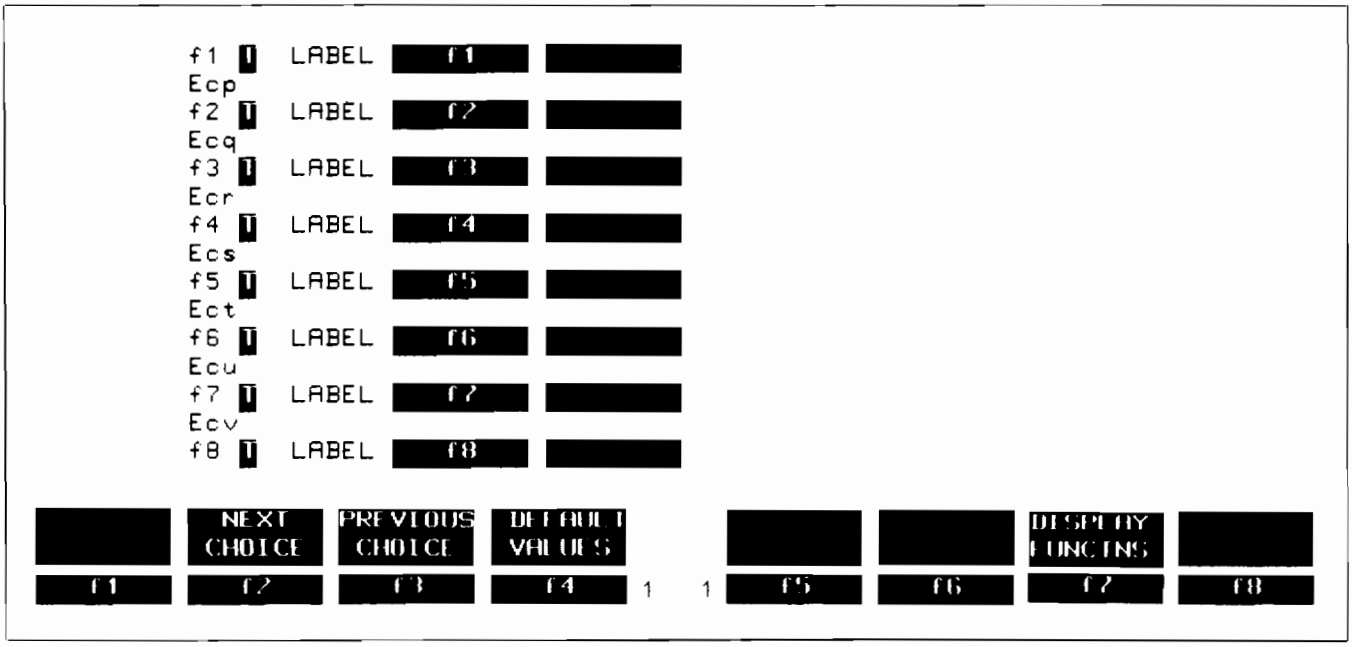


Figure 3-2. User Keys Definition Menu and Default User Key Labels

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.

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 either physically pressed or programmatically triggered. 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 DISPLAY FUNCTNS function key (f7). Note that this implementation of display functions mode is separate from that which is enabled/disabled via the mode selection keys.

The RETURN can be used for including ␍ codes (with display functions mode enabled) in key definitions. If auto line feed mode is also enabled, the RETURN key will generate a ␍␊, otherwise it is considered a cursor movement key.

When the user keys menu is displayed on the screen you may use the INSERT CHAR , DELETE 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 f0 , MODES , or USER KEYS key (in all three cases the user keys menu disappears from the screen). When you press USER KEYS , or enter Ec k , 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:

Ec&f <attribute><key><label length><string length><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 causes field to be erased)	(1 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. You must use an uppercase identifier (A, K, D, or L) for the final parameter and a lowercase identifier (a, k, d, or l) 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 ␍ and ␊, and explicit escape sequence characters) in the label string must not exceed 16, and in the definition string must not exceed 80.

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".

```
Ec&f5k6d19LLOG-ONHELLO USER.ACCOUNT␍
```

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␍
```

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. If the softkey labels are in the softkey window when the escape sequence is sent, they should appear in their changed form after the string is processed. If the user defined menu is active when the escape sequence is sent, the label and text string should be totally displayed once all characters have been processed.

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: Ec j
REMOVE MENU: Ec k

Controlling the Function Key Labels Programmatically

From a program executing in a host computer, you can control the function key labels display as follows by using escape sequences:

- 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.

The escape sequences are as follows:

Esc&j@	Disable the function keys and remove all key labels from the screen. Note: If a function key (f1 - f8) is hit while the terminal is in remote, the function key is transmitted (whether or not function key labels are displayed on the screen).
Esc&jA	Enable the mode selection keys.
Esc&jB	Enable the user keys.
Esc&jR	Unlock screen labels.
Esc&jS	Lock screen 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` fields in the terminal configuration menu.

Table 3-1 summarizes the effect of the ENTER key in each of the possible mode/strap combinations.

Table 3-1. ENTER Key Operation

CHARACTER MODE

The cursor is repositioned to column 1.

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, 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

If 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, alternate character set escape sequences, and field definition escape sequences within the field are NOT transmitted.

Table 3-1. ENTER Key Operation (Continued)

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 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, 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.

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

ASCII control codes within the field are transmitted.

Video 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 "line".

Inh DC2(H) = YES

The cursor is repositioned to the "home up" position. All characters through the first subsequent

Table 3-1. ENTER Key Operation (Continued)

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, 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

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

Inh DC2(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, 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)

Inh DC2(H) = NO → $P_1/P_2/P_1$

Inh DC2(H) = YES → no handshake

MODIFY MODE

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 mode 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 Menu 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, 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

Table 3-1. ENTER Key Operation (Continued)

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) is ignored

Inh DC2(H) = NO → $D_1/D_2/D_1$

Inh DC2(H) = YES → no handshake

SEND DISPLAY (Etd)

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:

Etd

This escape sequence can also be executed by typing Etd on the keyboard, or by putting Etd in a local user key.

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 → D_1

   InhHndShk(G) = YES

   Inh DC2(H) = NO → $D_1/D_2/D_1$

   InhHndShk(G) = YES

   Inh DC2(H) = YES → no handshake

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

Note that an Etd 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 Etd, 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, Modify All, or Line Modify is enabled. For more detailed information, refer to Table 3-1 "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: Etb

DISABLE: Etc

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

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. Resets the integral printer, if present.

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:

Etg

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.

More specifically, 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 display 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 or to their default power-on values.
8. Resets certain operating modes and parameters as follows:

   Disables display functions mode, caps mode, report mode, metric mode, data logging, and modify line.

   Resets the left margin to column 80.

   Resets the right margin to column 1.

   Turns off the "insert character" function edit.

   Resets the integral printer, if present.

   Resets the User Keys to default 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.

Display Control

SECTION IV

INTRODUCTION

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. As 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.

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 a left and right margin.
- 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 the inverse video, half bright, underline, and blinking, display enhancements.
- Change from one character set to another (Roman or Line Drawing, if present).

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 display memory.
- Delete the line containing the cursor (subsequent lines are rolled up).
- Change the line containing the cursor to all blanks.
- Delete the character at the current cursor position.
- 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

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 beginning 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 in the bottom line of the screen and rolls the text in display memory up as far as possible so that the last line in display memory appears immediately above the cursor position.

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 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.

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 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.

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).

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).

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, 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.

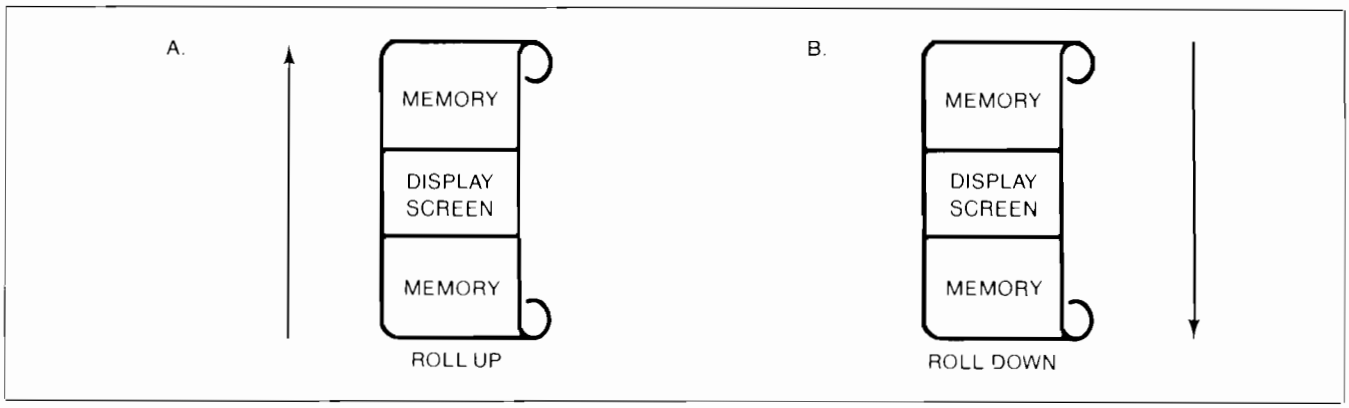


Figure 4-1. The "Roll" Data Functions

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:

Ec S

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, 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:

Ec T

Next Page/Previous Page

The data in display memory can be accessed (displayed on the screen) in blocks that are known as "pages". A page consists of 24 lines of data. The current page is that sequence of lines which appears on the screen at any given time. The previous page is the preceding 24 lines in display memory. The next page is the succeeding 24 lines in display memory.

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

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

Ec 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:

Ec V

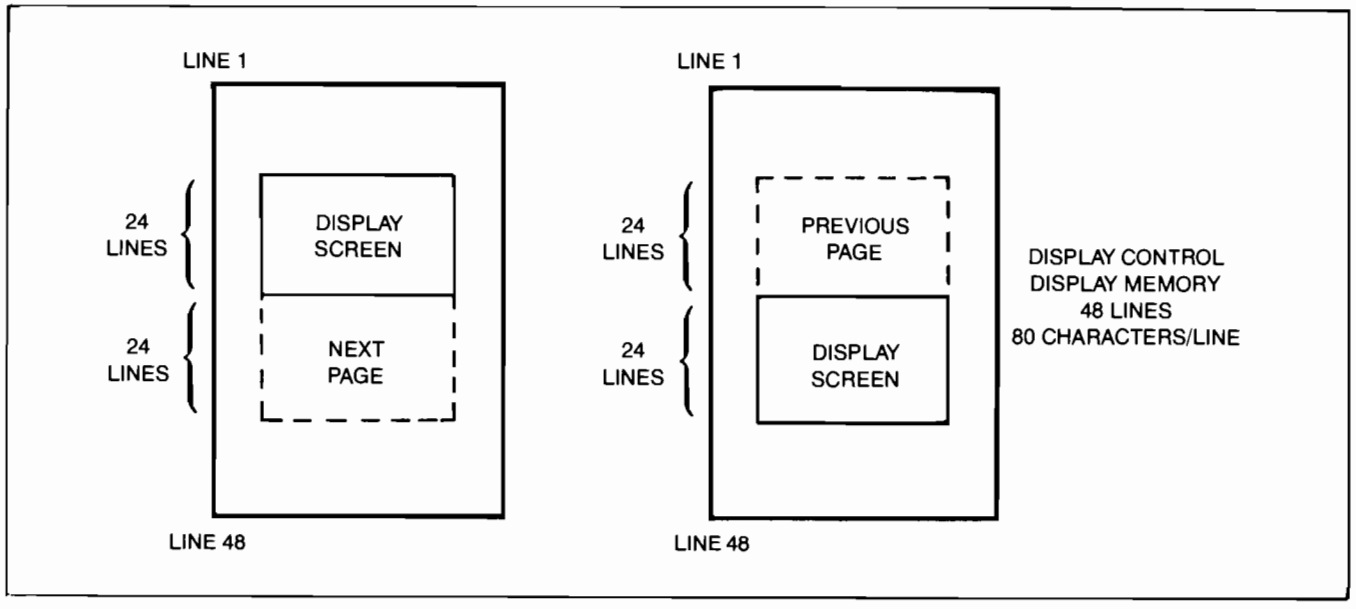


Figure 4-2. Previous Page and Next Page Concepts

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.

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). There can be as many as 48 lines of 80 characters (2 screens). The amount of memory in the terminal can be determined from byte 0 of the primary terminal status (refer to Section VI). 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 available screen positions (0 to 79 in figure 4-4a 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.

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 ESC a

Example: The cursor is at column 20, row 40.
computer: ESC a
terminal: ESC & a 020c 040R

Relative Sensing ESC '

Example: The cursor is again at column 20, row 40, but screen row 0 begins at row 35 of display memory.
computer: ESC '
terminal: ESC & a 020c 005Y

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.

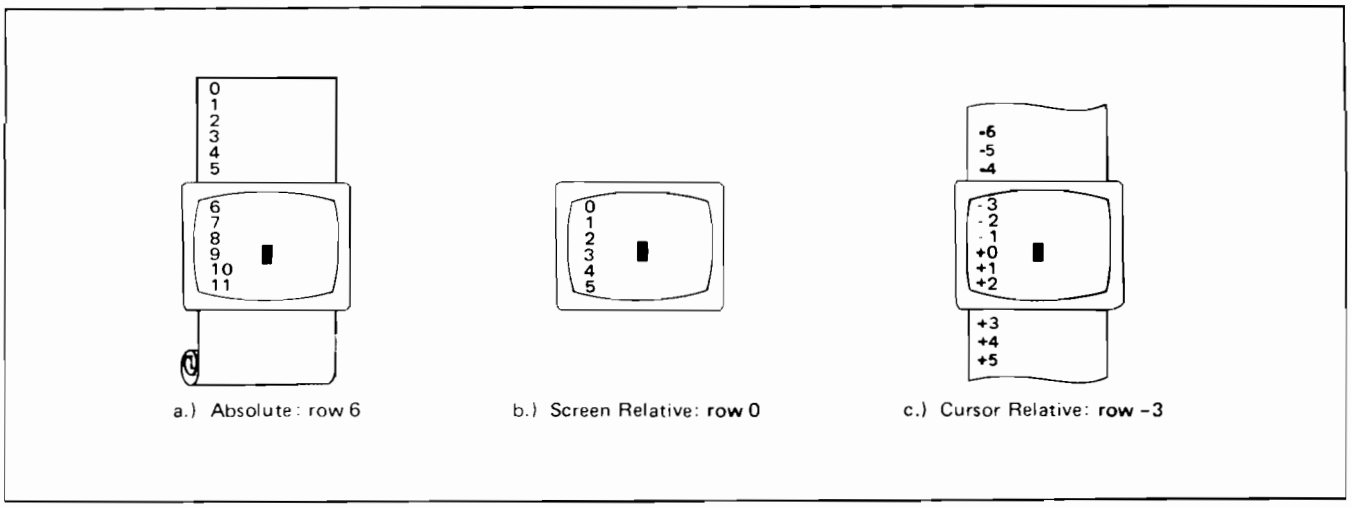


Figure 4-3. Row Addressing

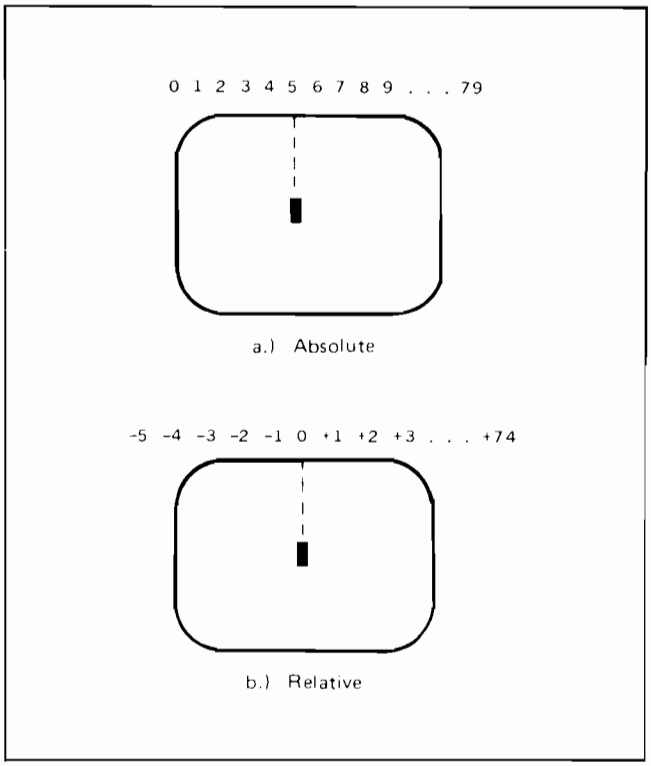


Figure 4-4. Column Addressing

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 to which you wish to move the cursor. Zero specifies the leftmost column.
<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.

When using the above escape sequences, the data on the screen always remains unchanged.

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 6y 19C
```

Absolute Addressing

You can specify the location of any character within display memory by supplying absolute row and column coordinates. (Note that this function is disabled when memory lock mode is on.) 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.
<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.

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.

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. (Note that this function is disabled when memory lock mode is on.) 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.
- 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.

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:

```
Ec&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.

```
Ec&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.

```
Ec&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.

```
Ec&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 INSERT 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 display memory (48 rows) is 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:

ESC L

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 DELETE 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:

ESC M

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.

When format mode is off, any unprotected, alternate character set, 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.

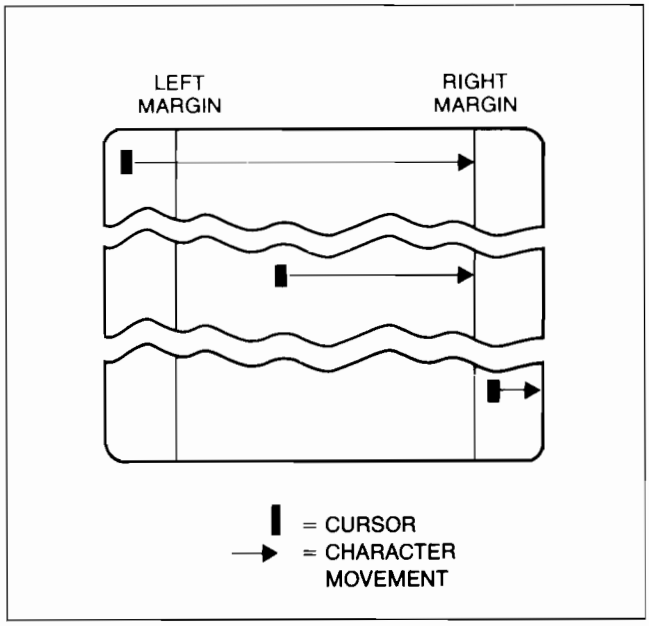


Figure 4-5. Character Insert with Margins

From the keyboard, you enable and disable the insert character editing function using the INSERT 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.

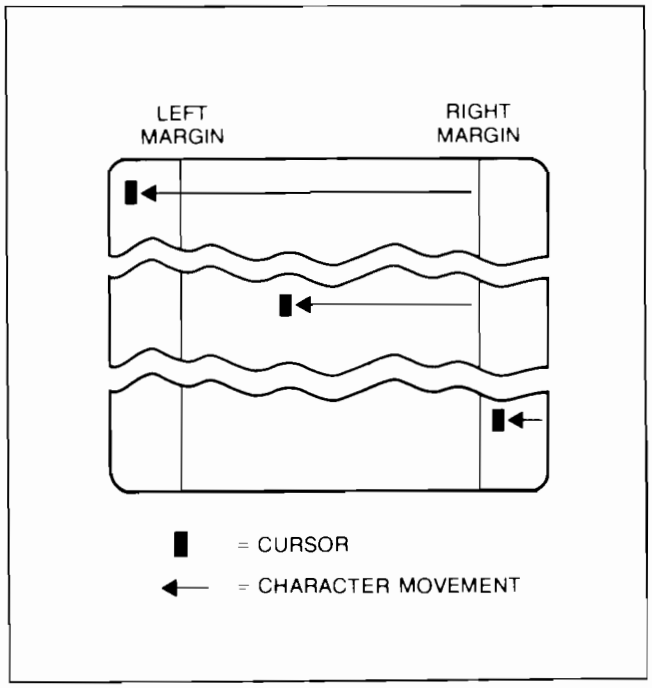


Figure 4-6. Character Delete with Margins

When format mode is off, any unprotected, alternate character set, 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 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 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 a protected field, the delete character function has no effect.

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, each time you press the DELETE 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 DISPLY key deletes all displaying and non-displaying characters from the current cursor position through the end of display memory.

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

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 from the current cursor position through the end of the current line.

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 all unprotected video enhancements from the current cursor position through the end of the current 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 a configuration menu.

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

ESC t 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 (1–80).

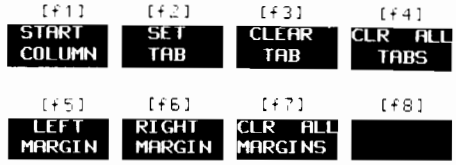
When you are entering data through the keyboard and the cursor reaches the right margin, it automatically moves to the left margin in the next lower line (note that this operating characteristic can be disabled through the use of the "InhEolWrp" terminal configuration parameter; refer to Section II). 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 CR 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/tab/col set of function keys. To get to that set, use the following keystroke sequence:

[AIDS] [f2] margins/ tab/col

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: ESC t 4
SET RIGHT MARGIN: ESC t 5
CLEAR ALL MARGINS: ESC t 9

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/tab/col set of function keys. To get to that set, use the following keystroke sequence:

[AIDS] [f2] margins/ tabs/col

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: Esc 1
CLEAR TAB: Esc 2
CLEAR ALL TABS: Esc 3

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. In format mode, tab moves the cursor to the beginning of the next unprotected field. The last field wraps around the beginning of the first field. Tab acts similarly in the user softkeys definition menu and the configuration menu.

From a program executing in a host computer, you can move the cursor ahead to the next tab stop issuing either an ASCII HT control code (decimal 9; Control "I") or 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 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.

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.

DISPLAY ENHANCEMENTS

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

- Inverse Video—black characters are displayed against a white background.
- Underline Video—characters are underscored.
- Blink Video—characters blink on and off.
- Half Bright—characters (or background for inverse video) are displayed at half intensity.

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 an @ or one of the uppercase letters A through O specifying the desired enhancement(s) as follows:

Enhancement Character

	@	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
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" (Esc&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.

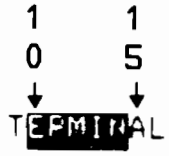
Step 1. Position the cursor at column 10 in line 5.

Step 2. Enter Esc&d C.

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

Step 4. Enter Esc&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.



DESIGNING AND USING FORMS

With the terminal, you can design elaborate data entry forms constructed of varying line types 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. 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 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

In order to draw lines with your terminal, you need option 202 of the HP 2622A terminal. Option 202 is equipped with two character sets: the standard USASCII character set and the Line Drawing character set.

With the terminal configured to its default state, the standard 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 keyboard are illustrated in figure 4-7.

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 standard USASCII character sets.

When you are designing a form through the keyboard, you use ASCII ^N (control-N) and ^O (control-O) codes to switch back and forth between the Line Drawing and USASCII character sets.

You switch from the Base set (USASCII) to the Line Drawing set by issuing an ASCII ^N code (control-N) and you switch from the Line Drawing set back to the Base set by issuing an ASCII ^O code (control-O). Note that the ^N code

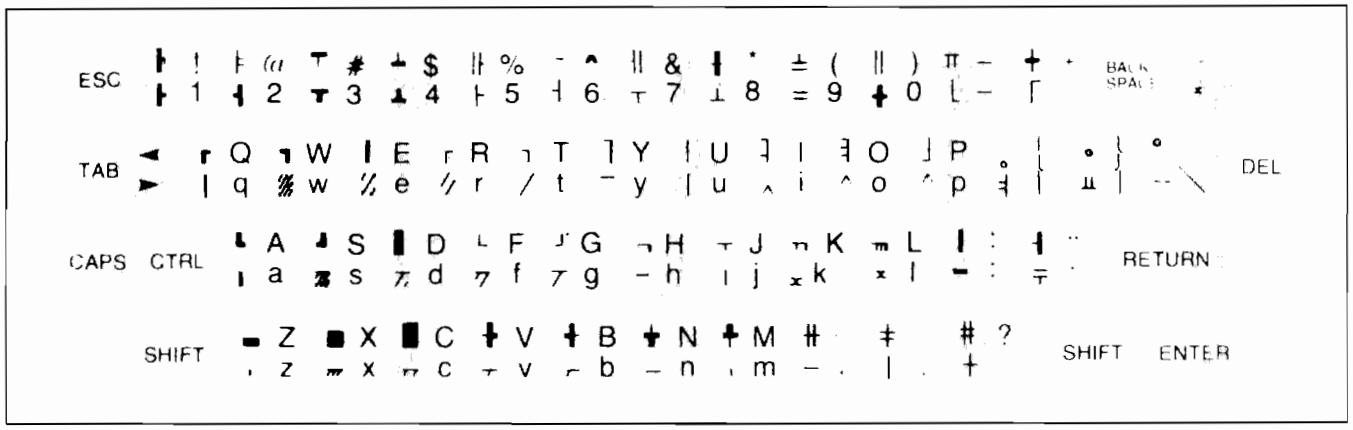


Figure 4-7. Line Drawing Set Elements

affects only those characters from the current cursor position through the next ESC 6 or ESC 5 code or through the end of the line, 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 ESC 6 code at the start of the second line.

Figure 4-8 illustrates a sample form and identifies the keystrokes used for generating the various different types of line segments. Figure 4-9 shows the same form as it actually appears on the terminal's screen. Figure 4-10 shows the Base set equivalent characters for the entire form structure.

One approach to generating a form structure through the keyboard is to load two of the user keys with the ESC 6 and ESC 5 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

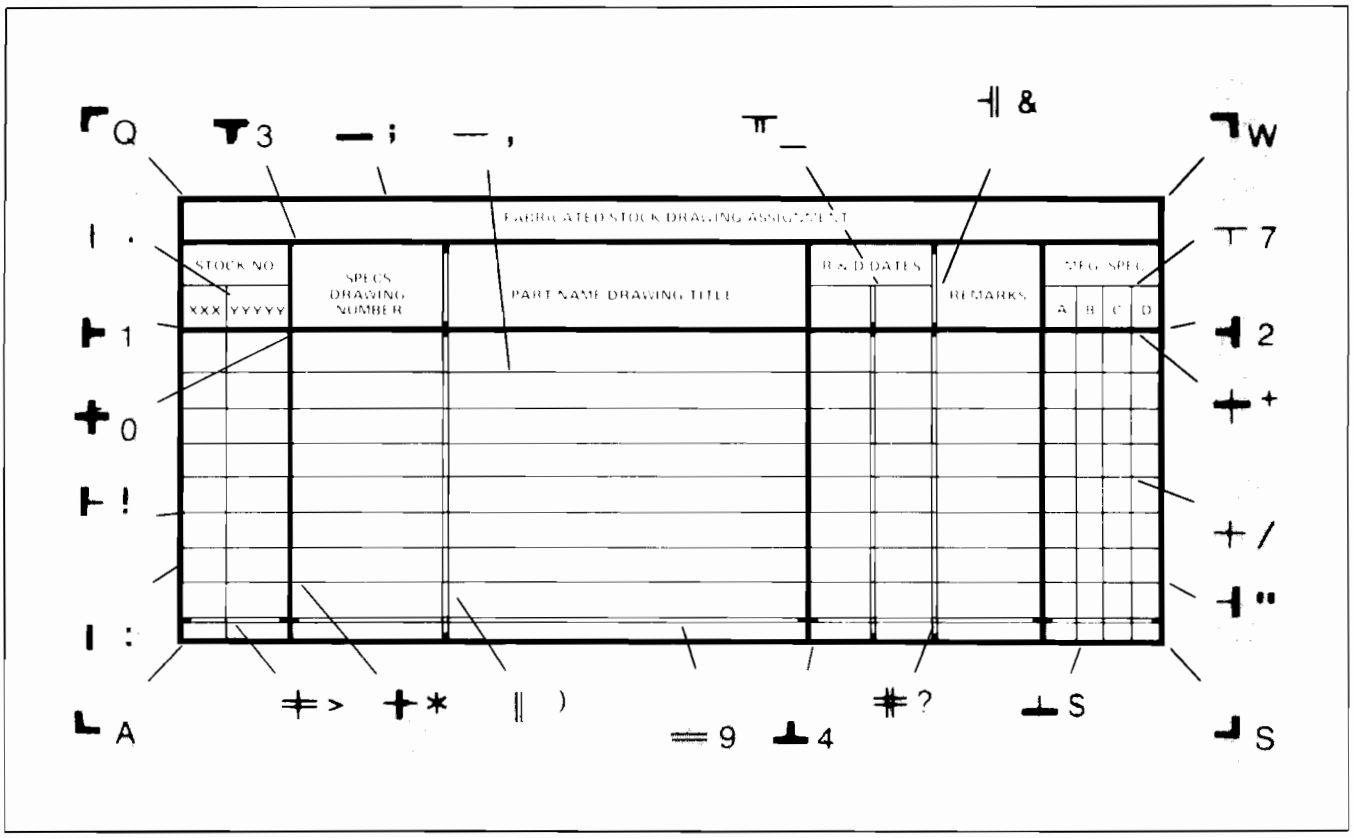
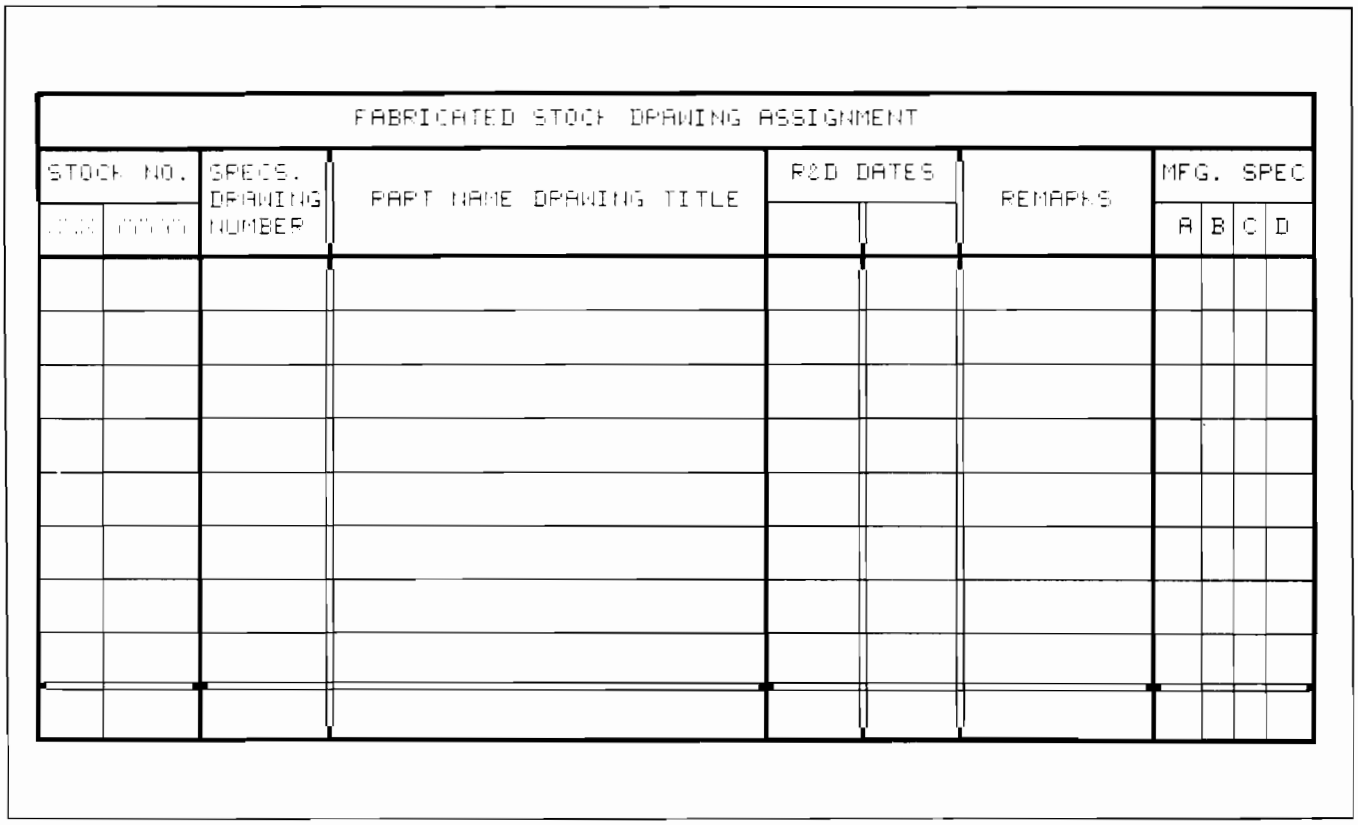


Figure 4-8. Sample Data Entry Form



FABRICATED STOCK DRAWING ASSIGNMENT

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

Figure 4-9. Completed Data Entry Form

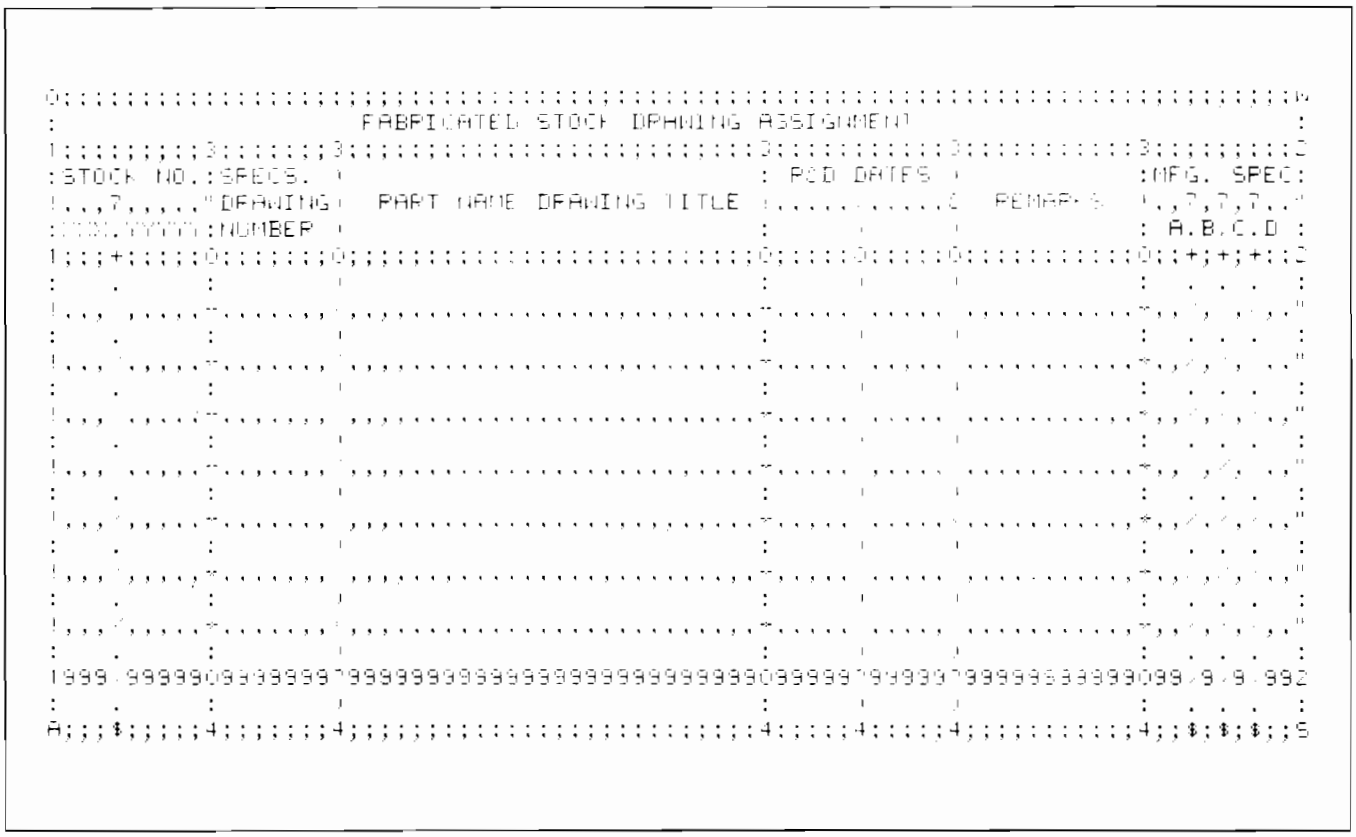


Figure 4-10. Base Set Equivalents for Data Entry Form Illustrated in Figure 4-9.

set as illustrated in figure 4-10. 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 4-11 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 definitions (such as the second and third lines in figure 4-11) into user keys to speed up the drawing of the main body of the form.

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 ESC)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 ESC W (the cursor is homed to the beginning of the first unprotected field). Normal operation is returned with ESC X (the cursor remains in its present position).

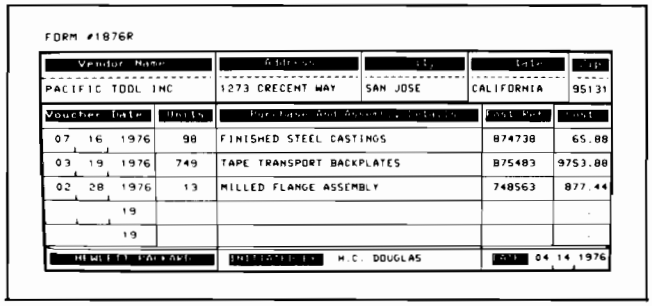
Protected Fields

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. ESC i causes the cursor to be positioned at the beginning of the previous unprotected field. Fields are defined as "unprotected" by using ESC [at the start of the field. ESC] 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.

Figure 4-11. Use of Shift-In and Shift-Out Codes

Printer Control

SECTION V

INTRODUCTION

As an option, your terminal may include an integral printer.

With the integral printer present, you may do any of the following:

- Print one or more lines from the display.
- Print the current content of the screen.
- Enable data logging (to occur either from the top or bottom of the display, as designated by you when you enable logging).
- Perform a line feed (advance the paper one line).
- Perform a form feed (advance the paper to the top of the next page when in either report or metric format; advance the paper one line when in continuous forms mode).

With the optional integral printer, you may also select continuous forms mode, report format (60 text lines per page), or metric format (64 text lines per page).

The above printer control functions can be initiated either locally by operator keystrokes or remotely by escape sequences sent from a host computer, if the printer option is present.

SELECTING PRINTER MODES

At power-on time or after a hard reset, the integral printer is automatically reset to print in continuous forms mode using normal-size characters (80 characters per line, 10 characters to the inch).

To enable or disable the various printer modes (expanded characters, compressed characters, report format, metric format, or data logging), you must get to the "device modes" set of system function keys. One way of doing so is the following keystroke sequence:

This changes the function key labels to the following:

The use of f3 - f8 "device modes" keys are described in the next few topics below. The "TO DISPLAY" key (f2) is used to select the display as a destination device. It is also used for eliminating the display as a destination device after it has been selected by way of device control (ESC&p) escape sequence.

Expanded Characters

The integral printer can print expanded characters in which each print line contains up to 40 characters spaced five to the inch (see figure 5-1).

From the keyboard, you enable and disable expanded character printing by pressing the EXPAND PRINT key (f5). When enabled, an asterisk appears in the key label.

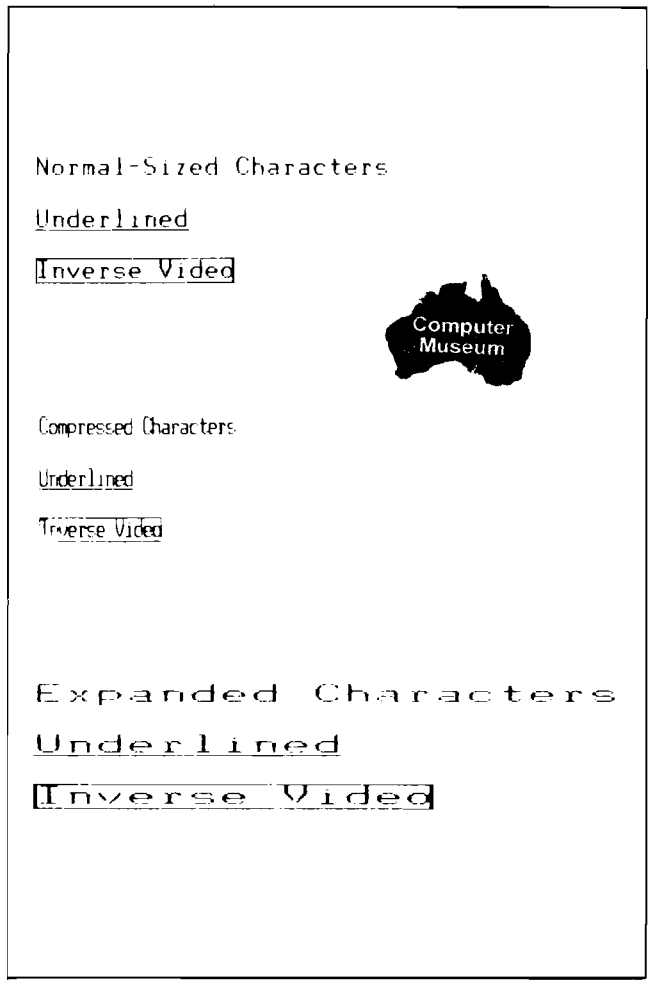


Figure 5-1. Character Sizes and Enhancements as Printed on the Integral Printer

From a program operating in a host computer, you enable and disable the printing of expanded characters using the following escape sequences:

ENABLE: ESC&k1S
DISABLE: ESC&k0S

Once the printing of expanded characters is enabled, it remains enabled until explicitly disabled, until compressed characters are enabled, until a hard reset is performed, or until the power is turned off.

Compressed Characters

The integral printer can print compressed characters in which each print line contains up to 132 characters spaced 16.2 to the inch (see figure 5-1).

From the keyboard, you enable and disable compressed character printing using the COMPRESS key (f6). When enabled, an asterisk appears in the key label.

From a program executing in a host computer, you enable and disable the printing of compressed characters using the following escape sequences:

ENABLE: ESC&k2S
DISABLE: ESC&k0S

Once the printing of compressed characters is enabled, it remains enabled until explicitly disabled, until expanded characters are enabled, until a hard reset is performed, or until the power is turned off.

Report Format

The integral printer normally operates in continuous forms mode without regard for page boundaries. You can, however, enable report format in which printed output is treated as a series of 66-line pages (a 3-line top margin, 60 lines of text, and a 3-line bottom margin). The margins and text area together form an 8½ inch by 11 inch page. The printer uses a small tic mark to mark the end of one page and the beginning of the next. Report forms mode is shown in figure 5-2.

From the keyboard, you enable and disable report format using the REPORT PRINT key (f7). This key alternately enables and disables report format. When enabled, an asterisk appears in the key display; the asterisk in f8, if on, is turned off; and the printer skips 3 lines, prints a page break, and skips 3 lines.

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

ENABLE: ESC&p17C
DISABLE: ESC&p19C

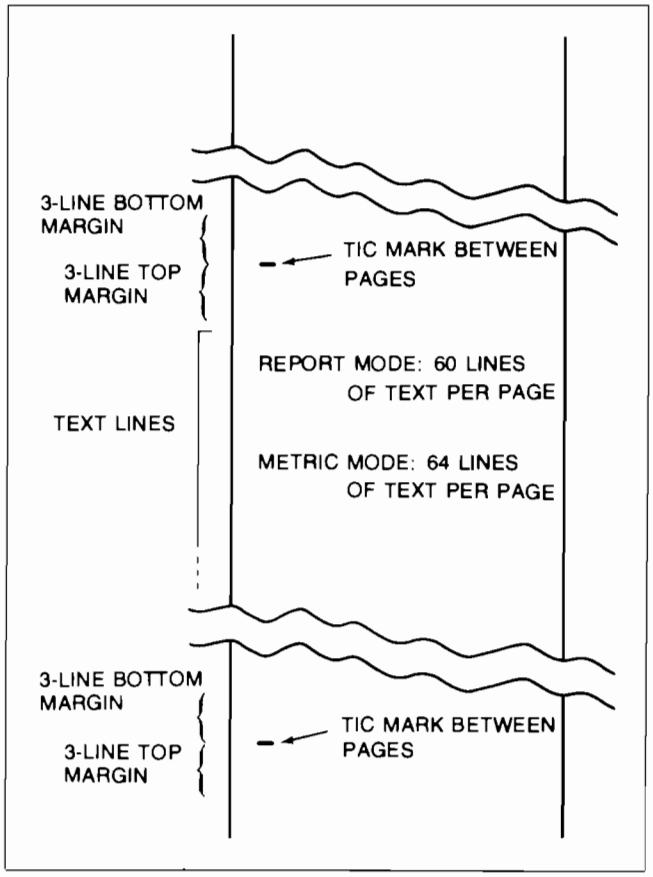


Figure 5-2. Report and Metric Formats

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

Metric Format

The integral printer normally operates in continuous forms mode without regard for page boundaries. You can, however, enable metric format in which printed output is treated as a series of 70-line pages (a 3-line top margin, 64 lines of text, and a 3-line bottom margin). The printer uses a small tic mark to mark the end of one page and the beginning of the next. Metric format is illustrated in figure 5-2.

From the keyboard, you enable and disable metric format using the METRIC PRINT key (f8). This key alternately enables and disables metric format. When enabled, an asterisk appears in the key display; the asterisk in f7, if on, is turned off; and the printer skips 3 lines, prints a page break, and skips 3 lines.

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

ENABLE: ESC&p18C
DISABLE: ESC&p19C

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

DATA LOGGING

The terminal includes a mechanism called "data logging" whereby data can be automatically routed to the integral printer and/or 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 selected printer can function. (For individual lines being logged sporadically on the integral printer, 1200 baud may work; for a series of successive lines, however, you will probably have to drop to 600 baud.)

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.

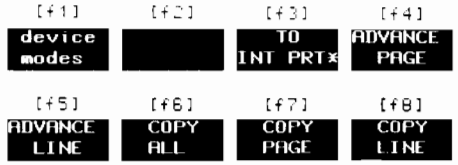
Once either form of data logging is enabled, it remains enabled until explicitly disabled, until the other form of data logging is enabled, until a hard reset is performed, or until the power is turned off.

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 DATA TRANSFERS

The display is defined as the "from" device in data transfers to the integral printer. If you define the integral printer as destination ("to") devices, you can use the "device control" set of system function keys to print one or more lines of data from the display. One way of getting to the "device control" keys is the following keystroke sequence:

This changes the function key labels to the following:



Press f3 for the integral printer.

Programmatically, you can define the "to" devices by using a device control escape sequence (ESC&p). See Appendix B.

Copy Line

When the internal printer is selected as a 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 on an empty line, COPY LINE should not cause anything to print.

From the keyboard, you copy one line of data using the COPY LINE 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 internal 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 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 internal 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 a destination ("to") device, you can copy all of display memory to it by issuing 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 a destination ("to") device and a configuration 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 integral printer is selected as a 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 integral printer by using the following device control escape sequence:

INTEGRAL PRINTER: ESC&p 1c 6u 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.

Skip Page

When the integral printer is selected as a 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.

When the integral printer is selected as a "to" device, this control function causes a true form feed only if report format or metric format is enabled. In all other cases, the advance page function merely causes the integral printer to advance the paper one line.

Programmatically, you can cause a form feed on the printer by using the following device control escape sequence:

INTEGRAL PRINTER: ESC&p 0c 6U

Note that the values 2 through 10 may also be used with the "c" parameter (instead of the zero); this will also initiate one form feed.

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 ␍ (or a ␍␊ if auto line feed mode is enabled) following the completion code. In block page mode, it sends a block terminator character.

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 VI.

COMPUTER TO TERMINAL DATA TRANSFERS

When the 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 characters (bytes) in <record>. This is an optional parameter. If present, then the record is terminated when the specified number of characters have been transmitted; otherwise the record is terminated when the 256th data byte after the "W" is transmitted or by the first ASCII ␊ code, whichever occurs first. If the record it terminated by an ␊, the ␊ is also passed to the printer.
<record>	is the data record to be transmitted.

Example: ESC&p 15W
Send the next 15 data bytes from the computer to all "to" devices.

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 display (3d) and the integral printer (6d).

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 ␊ or the 256th data byte terminates the record. In binary transfers, all eight bits received are passed to the printer. Priority 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 immediately following the "W" and preceding the data record 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 optional leading Enq-Ack handshake, ALL characters received are treated as data (including <ENQ>, <ACK>, <NULL>, and <DEL>) regardless of the setting of the EnqAck 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", "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 ␍ (or a ␍␊ 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.

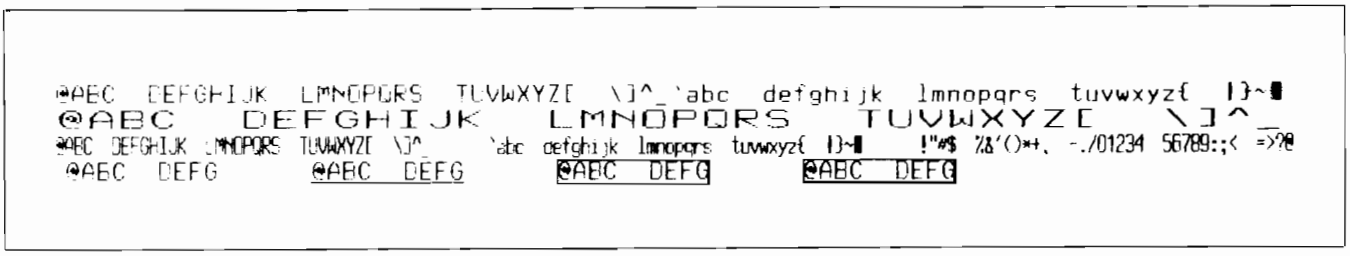


Figure 5-3. Integral Printer Self-Test Output

PRINTER SELF-TEST

The terminal includes a printer self-test feature that exercises the integral printer to verify that it is functioning properly.

From the keyboard, you initiate the printer self-test using the following keystroke sequence:

If the printer is present and functioning properly, it generates the test pattern shown in figure 5-3.

If an error condition is detected while the test is being executed, the message "INTEGRAL PRINTER ERROR" appears across the bottom of the screen. To clear the message, press RETURN. Note that the error condition may be either of the following, in which case you could correct it yourself:

1. Out of paper.
2. The metal latch (under the plastic printer lid) is not pressed down securely. See Section IX, figure 9-3.

The printer self-test cannot be initiated programmatically.

Note that while the printer self-test is in progress the terminal's interrupt mechanism is disabled. If you initiate this test while data is being received over a datacomm port, some of that data could get lost.

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 data 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 data link.
4. The final part provides programming reference material for someone who is writing a datacomm driver or controller program to communicate with 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.

Data 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 data link. The terminal is designed for use with a point-to-point data link.

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 data link.

Full Duplex: A data 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" data link). Point-to-point is supported by most computers. At any given time, it accomodates only one terminal per data link; it may, however, operate in either character mode or block mode.

A major consideration in selecting which type of connection to use is the anticipated distance between the terminal and the computer. If the terminal will be located in the vicinity of the computer system, you may use a hardwired connection. The Electrical Industry Association (EIA) Standard RS-232-C (to which the terminal conforms) limits cable lengths to a maximum of 15 meters (50 feet).

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.

HARDWIRED CONNECTIONS

If you decide on a point-to-point hardwired connection, the only decision that remains to be made is the type of cable to be used. The available cables are summarized in table 6-1. The pin assignments for each are illustrated in Appendix C of this manual. 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.

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)
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)
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)

Table 6-2. Modems

MODEM	Data Rate (Bits/Sec)	Duplex Full/Half	Dialed/ Leased	Reverse Channel
HP 13265A (see note 1)	300	F	D	No
Bell 103A	300	F/H	D/L	No
Bell 202T Bell 202D	1200 (see note 2)	F/H	L	Option
Vadic VA3400 (see notes 1, 3)	1200	F	D	No

NOTES:
1. Can be configured for either asynchronous or synchronous operation. With the HP 2622A, however, it must be configured for asynchronous operation.
2. C2 line conditioning allows operation at 1800 bits per second.
3. Must include the internal clock option.

Modem Considerations

If you are communicating with the host computer through a modem, 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 Bell 103A, Bell 212A, or equivalent type of modem.

Whenever the modem line (Data Set Ready) is active, an asterisk appears between the foruth and fifth screen label at the bottom of the screen. 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 compatible connector provided on the back of the terminal (see figure 6-1).

HP 2622A CABLING

The HP 13222 cables listed in table 6-1 all have a male 50-pin connector on one end and either a male or female RS-232-C 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 6-2.

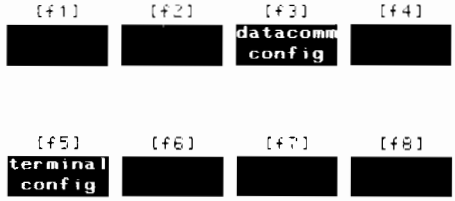
You may also connect either an HP 13265A Modem or an HP 13266A Current Loop Converter to the connector as illustrated in figure 6-3.

CONFIGURING THE TERMINAL

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:

This changes the function key labels to the following:



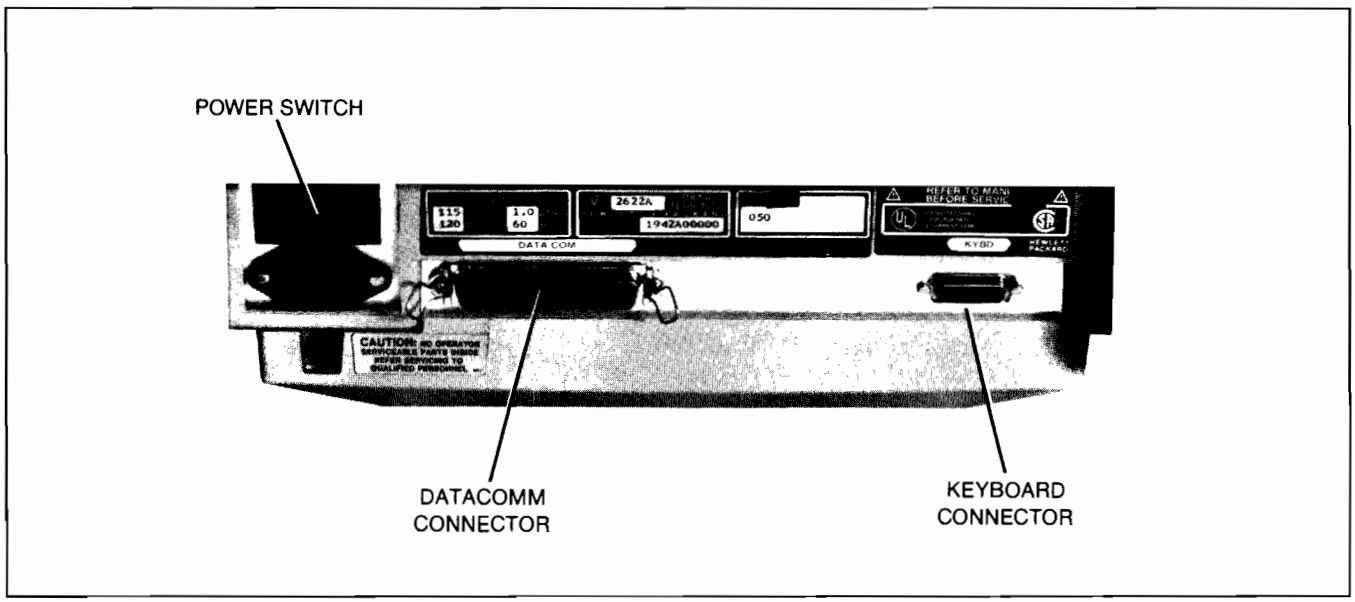
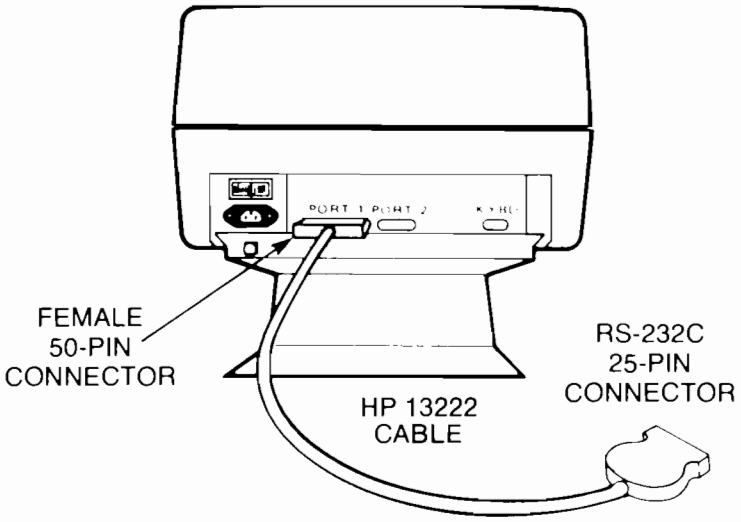


Figure 6-1. HP 2622A Display Terminal, Rear View



1. Interface cable to host computer.
2. Multiplexer panel on host computer.
3. Modem.
4. Modem bypass cable (HP 13232U).

Figure 6-2. HP 2622A Cabling (HP 13222 Cables)

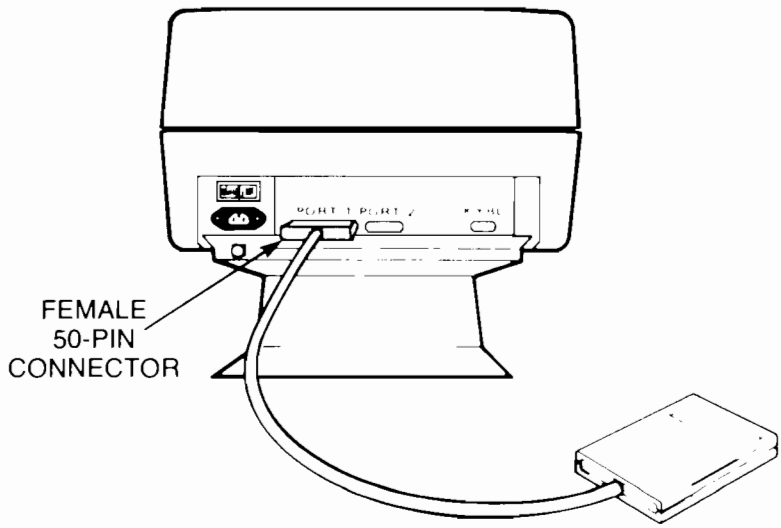


Figure 6-3. HP 2622A Cabling (HP 13265A Modem or HP 13266A Current Loop Converter)

To access the data communications configuration set of menus, press the **datacomm config** (f3) function key. When you press this 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]
		DISPLAY FUNCTNS	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 6-4).

The menu contains a set of unprotected fields that you access using the TAB► and TAB◄ keys. 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.

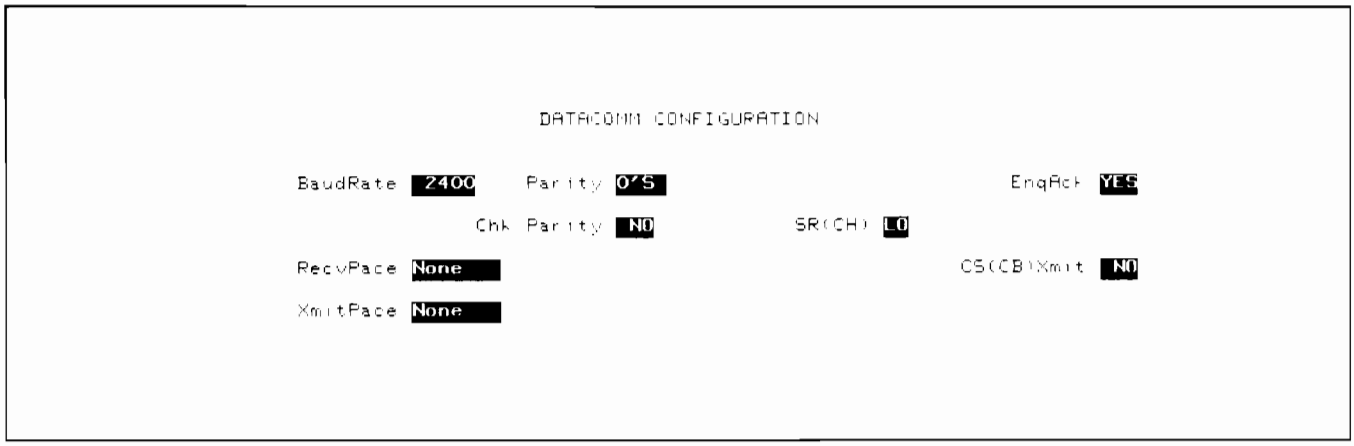


Figure 6-4. Datacomm Configuration Menu

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, 134.5, 150, 300, 600, 1200, 1800, 2400 (default), 4800, 9600 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 transmitted 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)

Table 6-3. Datacomm Configuration Menu Fields (Continued)

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 Detector (DTE Source). It is normally used on dual speed modems to select the appropriate speed (single speed modems merely ignore this line). Values: HI LO (default)
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/P_2 handshaking. Therefore, if XmitPace is enabled, InhHndShk (G) and Inh P_2 (H) in terminal configuration must both be set to "ON", which disables D_1/P_2 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

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.

[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".

The HP 2622A 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 character (■) is placed in the datacomm queue and later it is displayed on the terminal screen.

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)

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.
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.

Note that this type of transmit pacing can only be used in a hardwired configuration.

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."

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)

Status

SECTION VII

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 four 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
   SPOW
   Inh Eol Wrp
   Line/Page
   Inh Hndshk
   Inh DC2
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 integral printer.

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:	$_R$ or $_R\,{}^{L}{}_{F}$
Block Line Mode:	$_R$ or $_R\,{}^{L}{}_{F}$
Block Page Mode:	<BlkTermnator>

In either character mode or block line mode, the $_R\,{}^{L}{}_{F}$ is used if auto line feed mode is enabled. In block page mode, the block terminator is <Blk Termnator> ($_S$).

The type of handshaking used is determined by the setting of the InhHndShk and Inh $_2$ fields of the configuration menu as follows:

InhHndShk(G)	= YES	No handshake
Inh DC2(H)	= YES	
InhHndShk(G)	= NO	$_1$
Inh DC2(H)	= YES or NO	
InhHndShk(G)	= YES	$_1/_2/_1$
Inh DC2(H)	= NO	

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 7-1.

For a terminal ID request, the terminal returns the 5-character ASCII string "2622A".

Terminal ID Status

You request the terminal ID status by issuing the following escape sequence:

$_{E}{}_{Z}$

The terminal responds by sending back the following five-character string:

2622A

Table 7-1. 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:

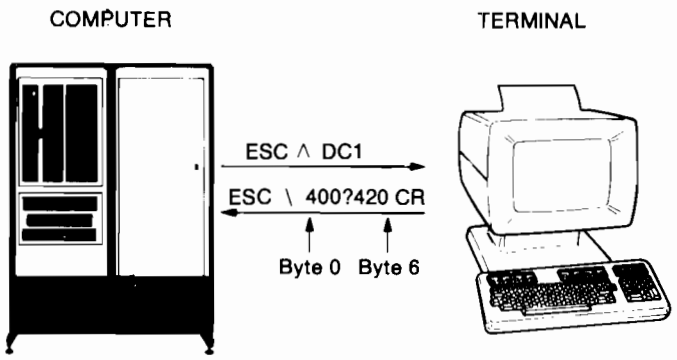
ESC ^

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 7-1. 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 7-2. The example assumes that the DC1 handshake is being used and that the appropriate terminator is a CR.

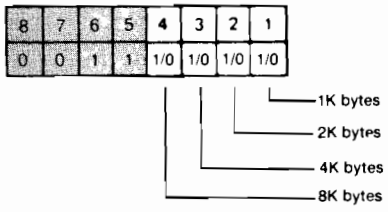


BYTE	ASCII	BINARY	STATUS
0	4	0011 0100	3840 bytes of display memory
1	0	0011 0000	Function key transmission disabled Space overwrite latch disabled Cursor wraparound disabled Strapped for line
2	0	0011 0000	Disable handshake Inhibit DC2 disabled
3	?	0011 1111	Upper case only Block mode Auto line feed Terminal sends secondary status
4	4	0011 0100	ENTER key has been hit
5	2	0011 0010	No data comm errorst Last Self-Test ok No device errors (no integral printer error)
6	0	0011 0000	No devise status pending No device operation status pending

Figure 7-1. Primary Terminal Status Example

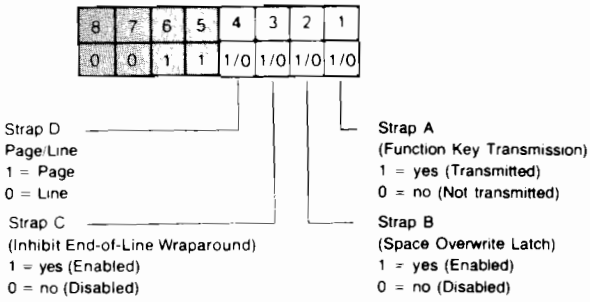
PRIMARY STATUS BYTES

BYTE 0 DISPLAY MEMORY SIZE



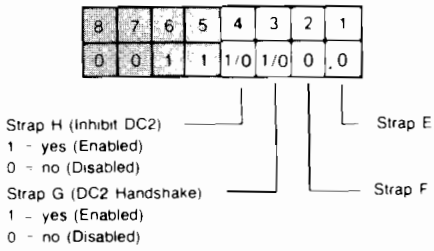
This byte specifies the amount of display memory available in the terminal. Note that "1111" specifies 15K or more bytes.

BYTE 1 CONFIGURATION STRAPS A–D



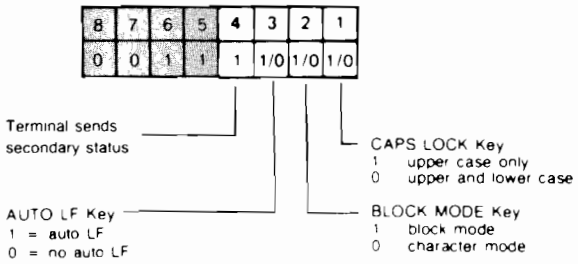
Refer to Section II for a detailed description of configuration straps A–D.

BYTE 2 CONFIGURATION STRAPS E–H

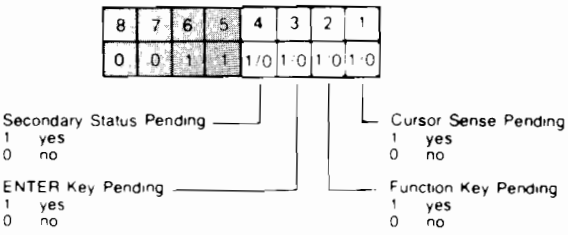


Refer to Section II for a detailed description of configuration straps G and H (straps E and F do not apply to the terminal).

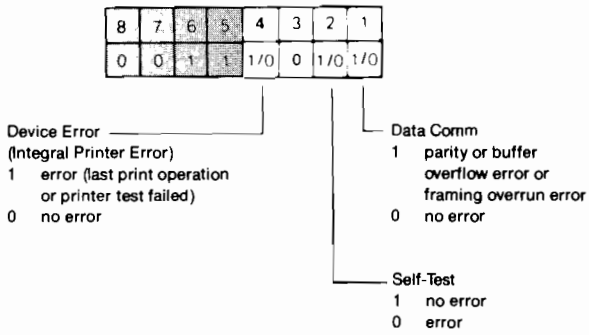
BYTE 3 LATCHING KEYS



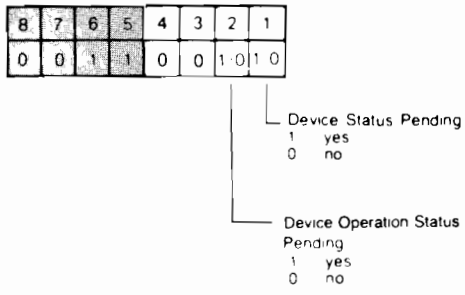
BYTE 4 TRANSFER PENDING FLAGS

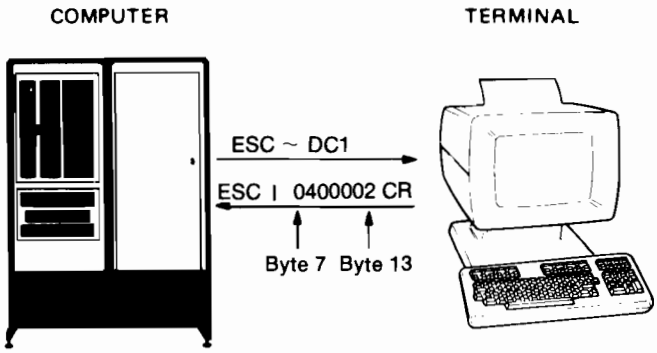


BYTE 5 ERROR FLAGS



BYTE 6 DEVICE TRANSFER PENDING FLAGS





BYTE	ASCII	BINARY	STATUS
7	0	0011 0000	
8	4	0011 0100	I/O firmware installed Terminal identifies self
9	0	0011 0000	
10	0	0011 0000	
11	0	0011 0000	
12	0	0011 0000	
13	2	0011 0010	Memory not locked

Figure 7-2. 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.

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, integral printer (Option 50) present
0 = No APL Firmware

APL Firmware does not apply.

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 CONFIGURATION STRAPS N–R
(always zero)

8	7	6	5	4	3	2	1
0	0	1	1	0	0	0	0

Strap R
Strap N
Strap Q
Strap P

Straps N–R do not apply to the 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 when option 050 (the integral printer) is installed.

The status of the integral 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 7-3.

You request device status by issuing the following escape sequence:

^{E}t&p <device code>^

where <device code> is either 4 or 6. Both 4 and 6 are interpreted as the integral printer.

If <device code> is any value other than 4 or 6, the escape sequence is ignored.

The terminal responds with the sequence ^{E}t\p <device code>, followed by three status bytes followed by a terminator. A typical device status request and response are illustrated in figure 7-4.

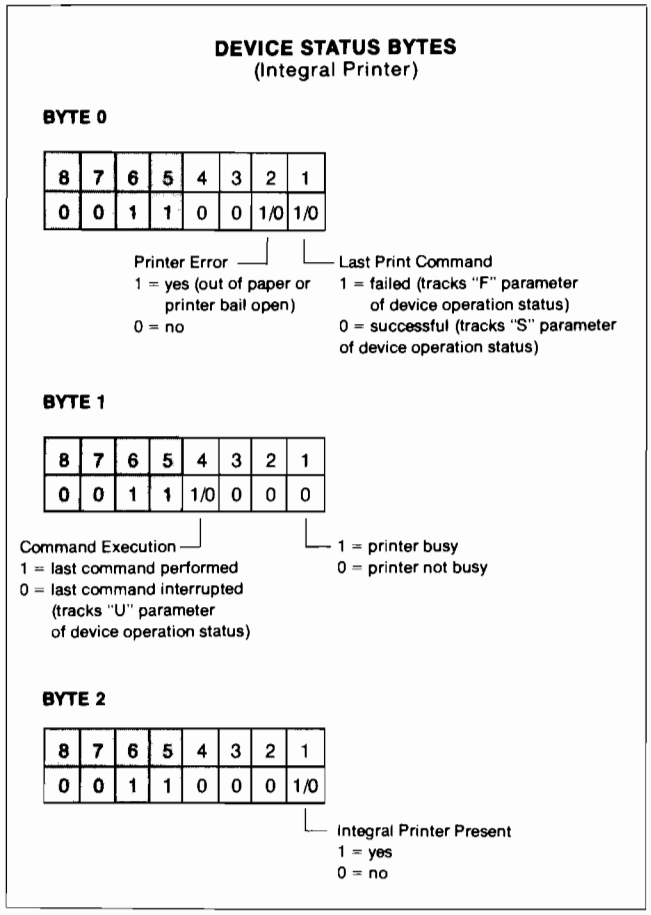


Figure 7-3. Device Status Bytes

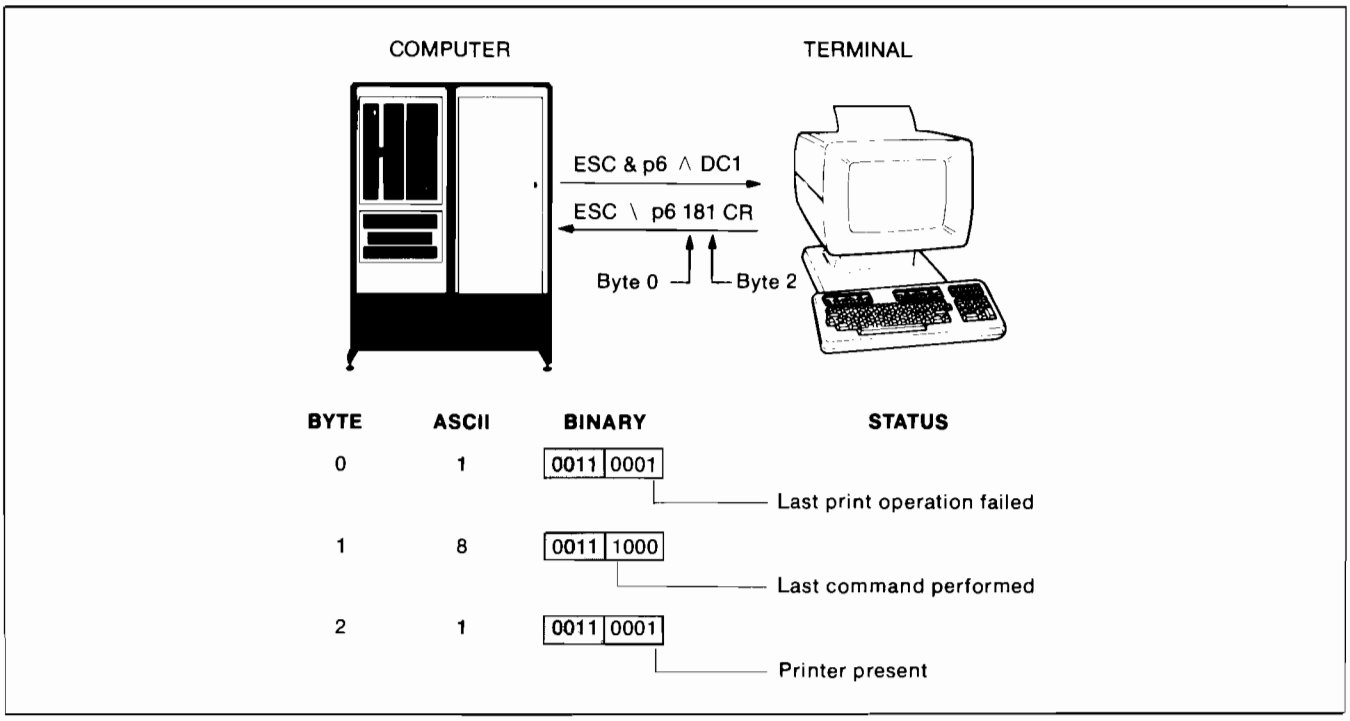


Figure 7-4. Device Status Example

Error Messages and Self-Test

SECTION VIII

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.

`Integral printer error`
`Press RETURN to clear`

Something is wrong with the integral printer. It may just be out of paper or the metal latch (under the plastic printer lid) may not be pressed down securely.

`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 and/or the integral printer as the "to" device, and then retry the copy operation.

TERMINAL SELF-TESTS

The terminal includes six types of self-tests:

- Power-On Test
- Manufacturing Test
- Terminal Test
- Identify ROMs
- Datacomm Test
- Internal Printer Test

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 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 15 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.

Manufacturing Test

Because the datacomm test is performed as part of the manufacturing test, you must connect a loopback test hood (test connector, part no. 02620-60056) to the datacomm port. Datacomm errors will result if the datacomm test connector is not used. The datacomm baud rate test performed during manufacturing test is performed on all possible baud rates (as opposed to the datacomm test which performs the baud rate test only on the currently selected baud rate).

Manufacturing test consists of the following items performed in the sequence listed:

1. The screen is filled with inverse video. This pattern outlines the display area boundaries, enabling adjustment of display area width and height, tilt, and centering. The test is then halted until the RETURN key is pressed.
2. After a carriage return, the screen is filled with @ signs to enable a focus alignment, if necessary. Again, a carriage return is necessary to continue the test.
3. After a carriage return, the following tests are repeated until stopped with a carriage return or a hard reset:
   a. Firmware ROM tests:
      1) Presence check for each required firmware ROM.
      2) Check of identification byte.
      3) Check of configuration address.
      4) Verify the ROM by running a cyclic redundancy check (CRC).
   b. Integral printer test (if printer installed).
   c. Non-destructive RAM test (display memory).
   d. Datacomm test.
   e. Destructive test of 8049 chip on integral printer PCA (if installed).
   f. Print character pattern.

To initiate the manufacturing test, press and hold the CTRL and left SHIFT keys, then simultaneously press the right SHIFT and ENTER keys.

Terminal Test

This test does the following:

1. Displays the message "TESTING" at the bottom of the screen (on the same line where the function key labels normally appear).
2. Verifies the integrity of all firmware ROM chips within the terminal.
3. Non-destructively verifies the integrity of all RAM chips within the terminal (including the one used for non-volatile memory).
4. Displays the test pattern shown in figure 8-1 or 8-2 (depending on whether you have the optional Line Drawing character set).

To initiate the Terminal Test press the following keys in the sequence shown:

[AIDS] [f3] service keys [f5] TERMINAL TEST

or

[MODES] [f5] TERMINAL TEST

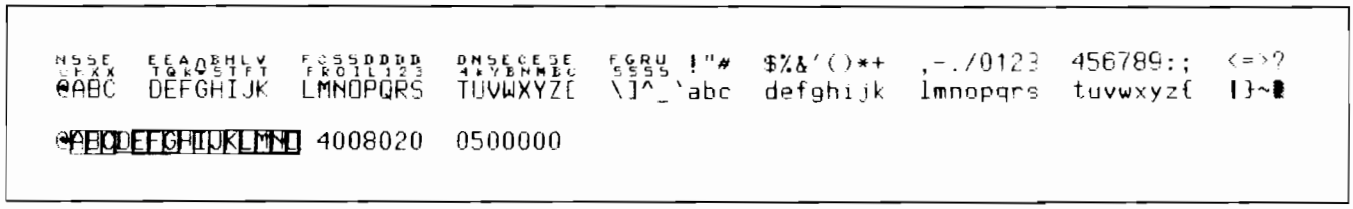


Figure 8-1. Screen Test Pattern, Standard Terminal

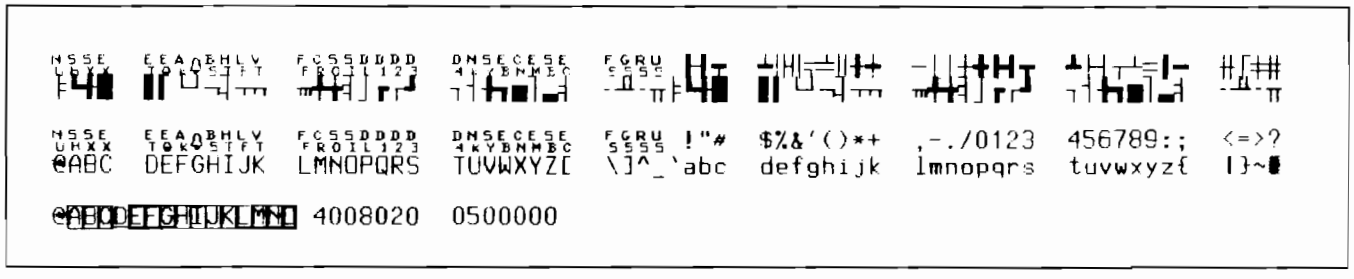


Figure 8-2. Screen Test Pattern, Option 202 (Line Drawing Character Set)

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 8-3 is displayed on the screen.

Datacomm Test

To enable the data communications (datacomm) self-test press the following keys in the sequence shown:

A test hood (HP part no. 02620-60056) is needed to perform the datacomm test. `DatacommError 1` will appear if no hood is used.

```
FIRMWARE ROMS

1818-1255 2004
1818-1256 2004
1818-1257 2004
```

Figure 8-3. ROM Identification Listing

The datacomm test consists of a loopback test, which requires one of the following test hoods:

02620-60062 or 02620-60062 — a male RS-232-C test hood.

02645-60004 — a female RS-232-C test hood for use on an HP 13222 or 13242 datacomm cable.

These test hoods are available as a set which can be ordered separately as the HP 13259A Data Comm Self-Test Connector Kit.

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.

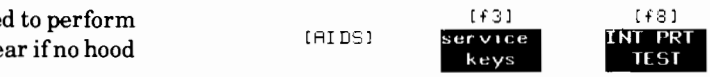
While the test is executing, "Testing" is displayed on the terminal screen. If no errors are found, the terminal beeps and displays the softkeys. If an error is found, an error message will appear on the terminal screen, similar to a ROM error.

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

Printer Test

The Printer Test, which is usable only if the optional integral printer is present, exercises all the features of the integral printer to verify that it is functioning properly. To initiate this test press the following keys in the sequence shown:



If the printer is functioning properly it generates the test pattern shown in figure 8-5.

Note that if your terminal does NOT include the integral printer the [f8] key label in the "`service keys`" set of function keys will be blank and pressing that key will have no effect.

If an error condition is detected while the test is being executed, the message

```
INTEGRAL PRINTER ERROR
Press RETURN to clear
```

appears across the bottom of the screen. To clear the message, press RETURN. Note that the error condition may be either of the following, in which case you could correct it yourself:

1. Out of paper.
2. The metal latch (under the plastic printer lid) is not pressed down securely.

If the error is not due to one of these conditions, call your nearest HP Service Representative.

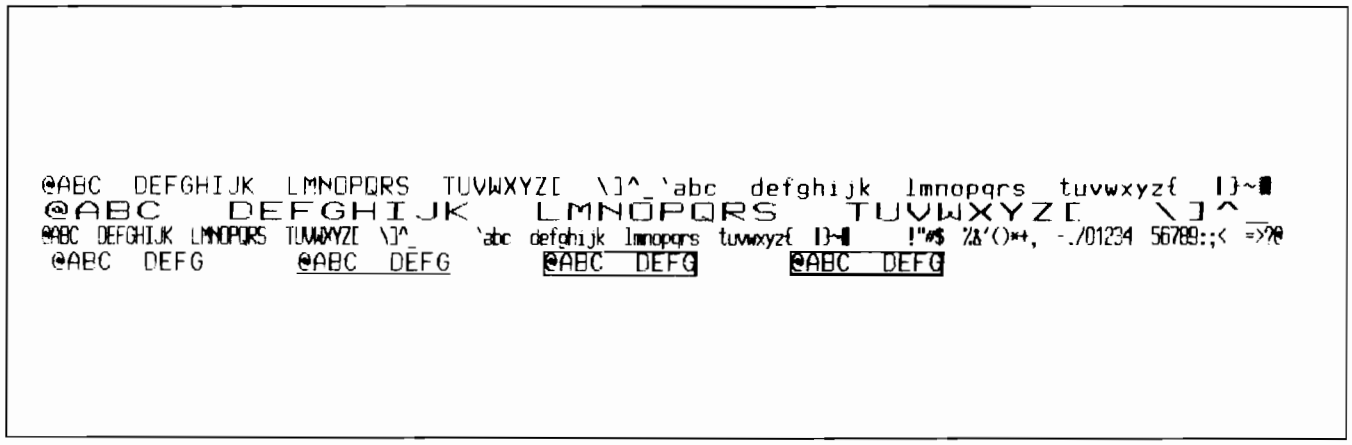


Figure 8-5. Integral Printer Self-Test Output

Terminal Maintenance Procedures

SECTION IX

INTRODUCTION

This section provides information on preventive maintenance for your terminal, such as cleaning the screen and keyboard; and routine maintenance such as replacing paper in the integral printer.

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 most conventional spray cleaners. Avoid spraying 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 a battery that is located just above the rear panel of your terminal. Figure 9-1 shows the rear panel and the location of the battery.

The battery requires no special care or maintenance. It should, however, be replaced with a new battery every 12 months. You may purchase a replacement battery through conventional retail stores. When doing so, request a Mallory Battery, Type TR133. You may also order replacement batteries through your local HP Sales and Service Office using the following nomenclature and part number:

HP 2622A Battery, HP Part No. 1420-0259

If your terminal includes the optional thermal printer you may wish to record the various configuration menus on paper before removing the old battery. To do so, perform the following:

1. Use the "`config keys`" set of function keys to display the particular menu of the screen.
2. Press the ENTER key.

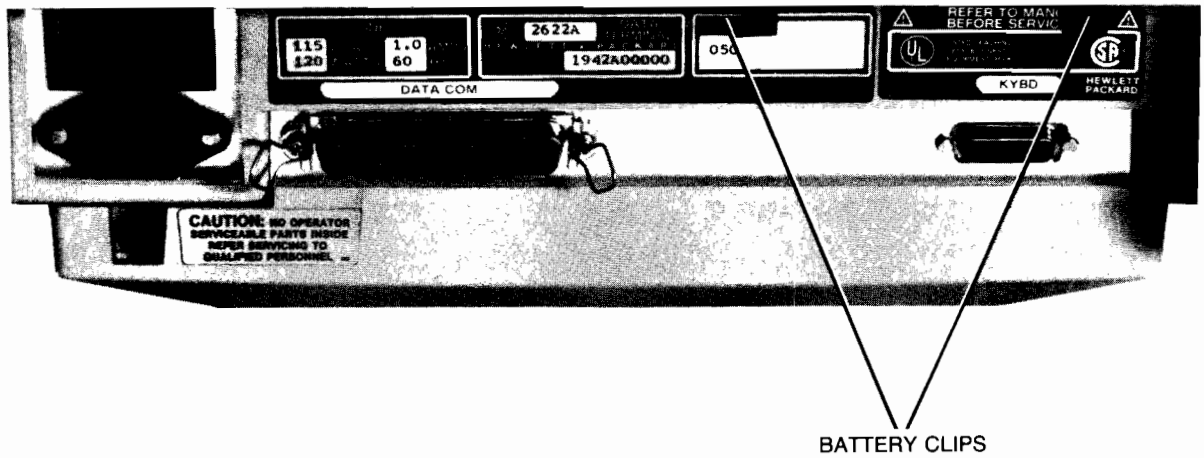


Figure 9-1. Battery Location, Rear Panel

Replace the battery as follows:

1. Grasp the battery support at points A and B as shown in figure 9-2.
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.
3. Gently pull the support downward until it is completely free from the terminal housing.
4. Remove the old battery from the support.
5. Install the new battery in the support making sure that the positive end of the battery matches the positive end of the support (+ to + and − to −).
6. Reinsert the battery support into the terminal. A slotted guide along one side of the battery support ensures that the support is inserted correctly. The slotted guide must be facing away from the terminal case when you reinsert the support (otherwise the support will not fit back into the terminal).

THERMAL PRINTER PAPER

The optional thermal printer mechanism uses a thermal printing paper that is manufactured specifically for use by the HP 262x family of terminals. You can purchase it through your local HP Sales and Service Office using the following nomenclature and HP part number:

1 Box (24 rolls) Thermal Paper, HP Part No. 9270-0638

It is recommended that you only use HP Thermal Paper in your terminal. If you have an HP Warranty and Service Contract, you MUST use only HP Thermal Paper in order to maintain a valid contract. HP Warranty and Service Contracts are available through your local HP Sales and Service Office.

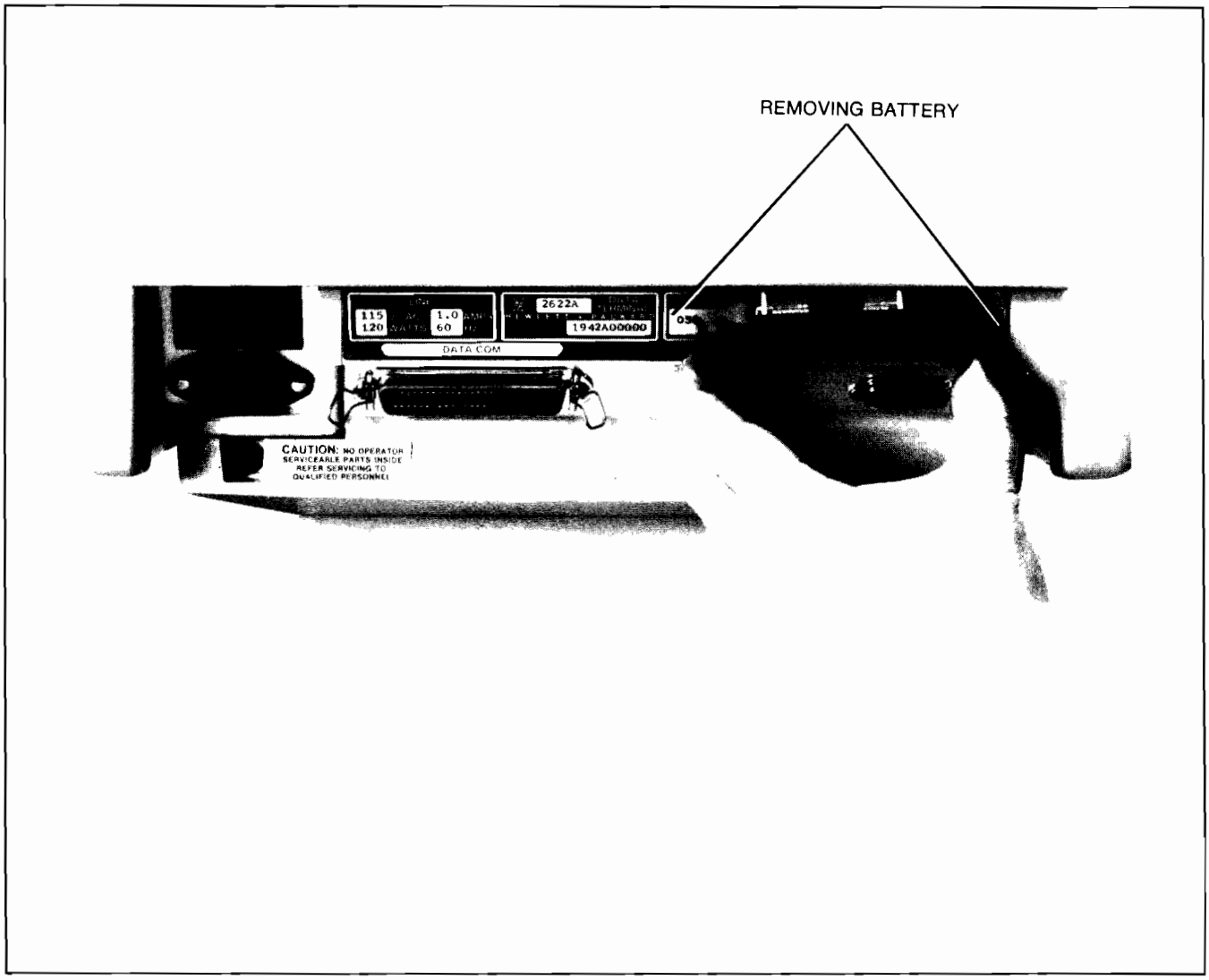


Figure 9-2. Removing the Battery

Paper Loading

The printer mechanism is shown in figure 9-3.

Load a roll of thermal paper into the printer as follows:

1. Lift the top cover of the printer mechanism. An illustration of the correct paper position and flow is embossed on the underside of this cover.
2. Press the latch (figure 9-3) toward the front of the terminal to release the latching frame. Lift the hinged latching frame to its forward position.
3. Remove any paper remaining in the printer.
4. The center paper core is held in place by a metal rod inserted through the center of the core. Grasp the core and lift forward and upward along the guide slots to remove the core and rod.
5. Remove the rod from the old core and insert the rod through the core of a new roll of paper.

   The HP Thermal Paper is coated with print material on one side and must be inserted into the printer correctly to produce the print image. The paper must feed toward the front of the terminal from the underside of the paper roll (see the embossed illustration on the top cover).
6. Place the ends of the metal rod into the guide slots on either side of the print mechanism and press downward and then toward the back of the terminal until the rod snaps into place.
7. Feed the leading edge of the paper through the latching frame (between the latching frame and the clear plastic guide window). Be careful not to sharply strike the print head because damage may result.
8. Lower the latching frame without locking it into place.
9. Align the sides of the paper with the guide lines embossed on each side of the guide window.
10. Each new roll of HP Thermal Paper has a glue spot near the leading edge of the roll that holds the paper roll intact during shipment. You must not allow the print head to come in contact with this glue spot.

    Feed approximately 12 inches of paper through the latching frame so that the glue spot is beyond (outside) the print head and guide window.
11. Press the latch down until it locks into place with an audible click.
12. Tear off the excess paper using the edge of the guide window as a cutting edge.
13. Close the top cover securely and press RETURN.

Note that if subsequent printer operations produce no image on the paper, the paper has probably been installed with the wrong side facing the print head. An image can be printed only on one side of HP Thermal Paper.

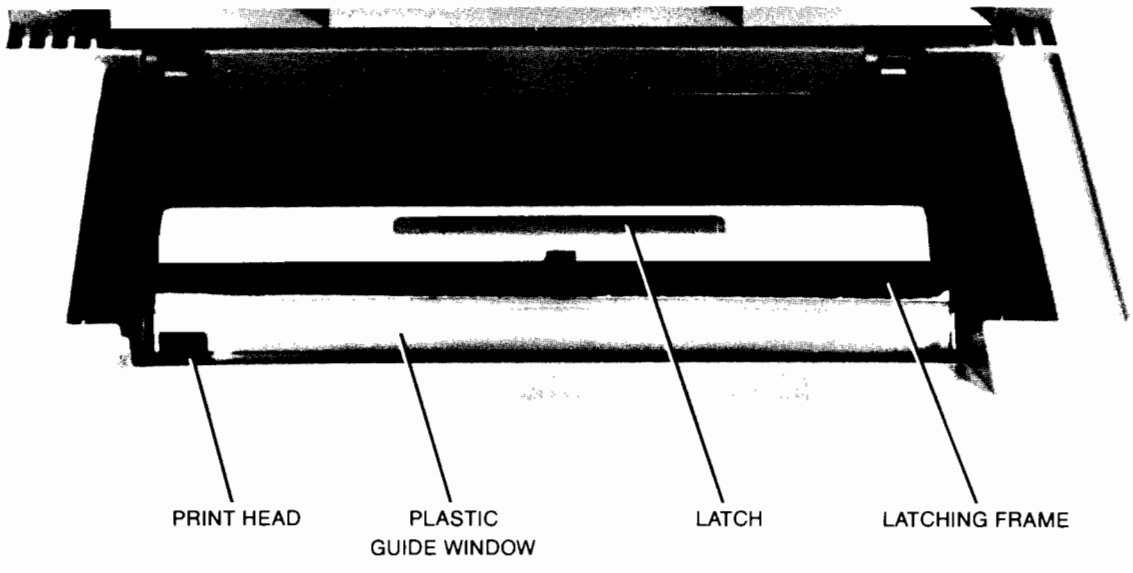


Figure 9-3. Printer Mechanism

Character Sets

APPENDIX
A

ASCII CHARACTER SET

Table A-1 contains the entire 128-code ASCII character set.

LINE DRAWING SET

If your terminal contains the line drawing character set option, you can construct forms and tables by combining different types of line segments. Each individual type of line segment is associated with one of the alphanumeric or symbol keys as shown in figure A-1. Figure A-2 contains the keystrokes used for generating a sample data entry form.

Note that you use the "shift out" (SO) and "shift in" (SI) control codes to shift from the standard ASCII character set to the line drawing set and back again. For example:

1. To shift from ASCII to the line drawing set, use SO by pressing the N key while holding the CTRL key down.
2. To shift from the line drawing set back to ASCII, use SI by pressing the O key while holding the CTRL key down.

See Appendix C for further information on using the line drawing set.

Table A-1. ASCII Character Set

DECIMAL VALUE	GRAPHIC	COMMENTS	ALTERNATE CHARACTER	DECIMAL VALUE	GRAPHIC	COMMENTS
0		Null	@[c]	64	@	Commercial at
1		Start of heading	A[c]	65	A	Uppercase A
2		Start of text	B[c]	66	B	Uppercase B
3		End of text	C[c]	67	C	Uppercase C
4		End of transmission	D[c]	68	D	Uppercase D
5		Enquiry	E[c]	69	E	Uppercase E
6		Acknowledge	F[c]	70	F	Uppercase F
7		Bell	G[c]	71	G	Uppercase G
8		Backspace	H[c]	72	H	Uppercase H
9		Horizontal tabulation	I[c]	73	I	Uppercase I
10		Line feed	J[c]	74	J	Uppercase J
11		Vertical tabulation	K[c]	75	K	Uppercase K
12		Form feed	L[c]	76	L	Uppercase L
13		Carriage return	M[c]	77	M	Uppercase M
14		Shift out	N[c]	78	N	Uppercase N
15		Shift in	O[c]	79	O	Uppercase O
16		Data link escape	P[c]	80	P	Uppercase P
17		Device control 1 (X-ON)	Q[c]	81	Q	Uppercase Q
18		Device control 2	R[c]	82	R	Uppercase R
19		Device control 3 (X-OFF)	S[c]	83	S	Uppercase S
20		Device control 4	T[c]	84	T	Uppercase T
21		Negative acknowledge	U[c]	85	U	Uppercase U
22		Synchronous idle	V[c]	86	V	Uppercase V
23		End of transmission block	W[c]	87	W	Uppercase W
24		Cancel	X[c]	88	X	Uppercase X
25		End of medium	Y[c]	89	Y	Uppercase Y
26		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 A-2. 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

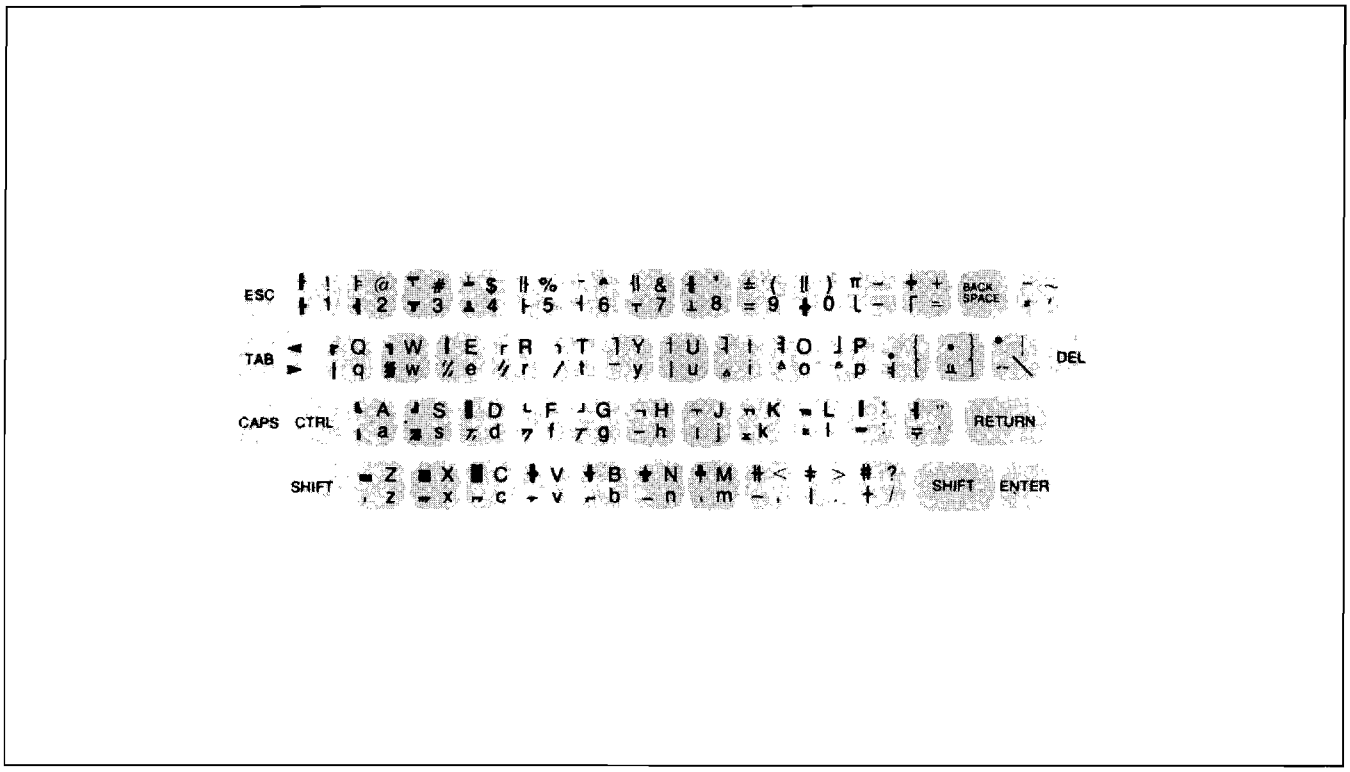


Figure A-1. Line Drawing Set Elements

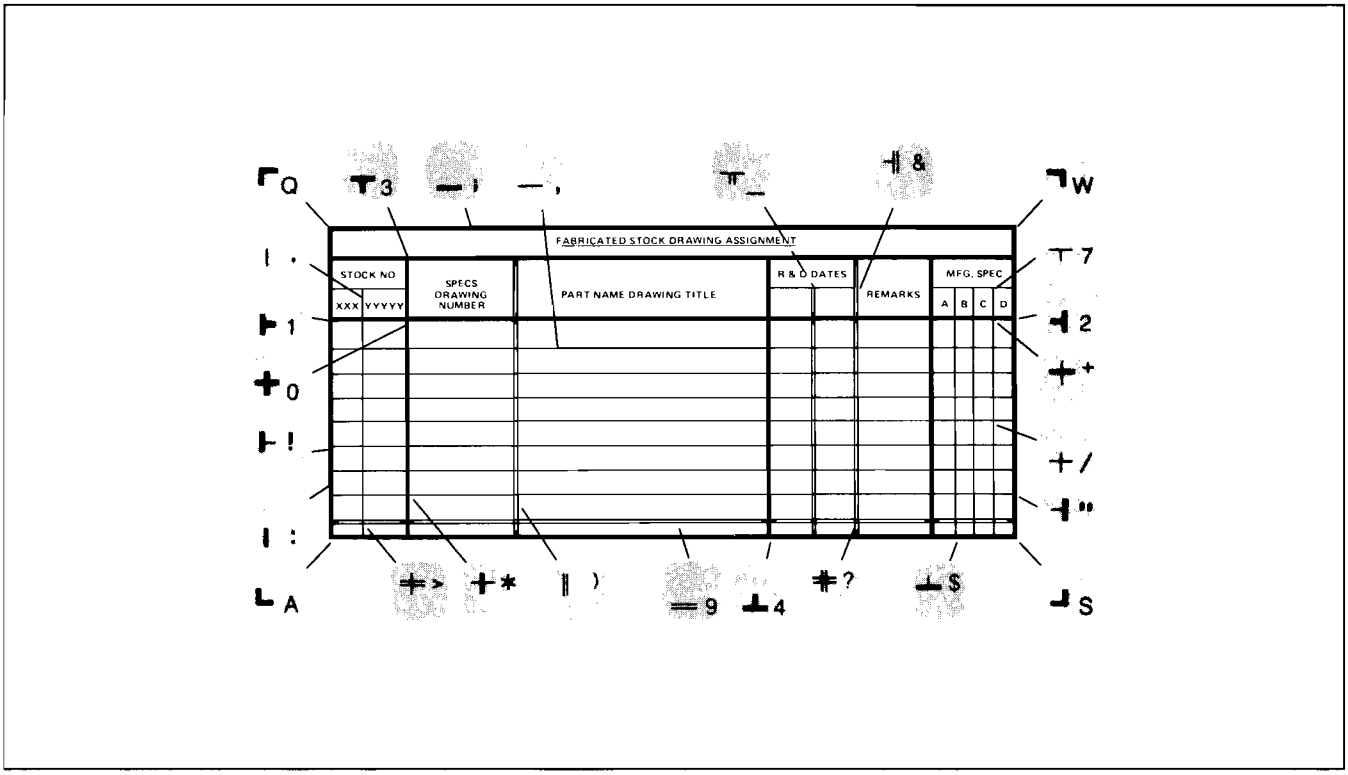


Figure A-2. Sample Data Entry Form

Escape Codes

APPENDIX B

KEY(S)			CODE	FUNCTION
TERMINAL CONTROL FUNCTION				
	(as used in Local mode)		↖ 0	Copy memory to destination(s)
	margins / tabs /col	SET TAB	↖ 1	Set tab
	margins tabs /col	CLEAR TAB	↖ 2	Clear tab
	margins / tabs /col	CLR ALL TABS	↖ 3	Clear all tabs
	margins / tabs /col	LEFT MARGIN	↖ 4	Set left margin
	margins / tabs /col	RIGHT MARGIN	↖ 5	Set right margin
	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

KEY(S)		CODE	FUNCTION
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
	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
INSERT LINE		↖ L	Insert line
DELETE 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
NEXT PAGE		↖ U	Next page
PREV PAGE		↖ V	Previous page

KEY(S)	CODE	FUNCTION
TERMINAL CONTROL FUNCTION (continued)		
	↯ W	Format mode on
	↯ X	Format mode off
, DISPLAY FUNCTION	↯ Y	Display Functions mode on
, DISPLAY FUNCTNS •	↯ Z	Display Functions mode off
	↯ f	Start unprotected field
	↯]	End unprotected/transmit-only 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

KEY(S)	CODE	FUNCTION
	↯ h	Cursor home up
+	↯ i	Backtab
+	↯ j	Begin User Key Definition mode
or or	↯ k	End User Keys Definition mode
MEMORY LOCK	↯ l	Begin Memory Lock mode
MEMORY LOCK •	↯ M	End Memory Lock mode
f1	↯ p	Default definition for user definable function key f1
f2	↯ q	Default definition for user definable function key f2
f3	↯ r	Default definition for user definable function key f3
f4	↯ s	Default definition for user definable function key f4
f5	↯ t	Default definition for user definable function key f5
f6	↯ u	Default definition for user definable function key f6
f7	↯ v	Default definition for user definable function key f7
f8	↯ w	Default definition for user definable function key f8

KEY(S)	CODE	FUNCTION
TERMINAL CONTROL FUNCTION (continued)		
service keys	TERMINAL TEST	↯ z Initiate terminal self test
TERMINAL TEST		
		↯ ~ Secondary terminal status request

CURSOR CONTROL OPERATIONS

NOTE

Columns and rows are numbered starting with 0 as the leftmost column and the top row.

↯ &a <col >c <rows >Y	Moves the cursor to column "col" and screen row "row" or the screen (screen relative addressing).
↯ &a <col >c <row >R	Moves the cursor to column "col" and row "row" in memory (absolute addressing).
↯ &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.
↯ &a ± <col >c ± <row > R	Moves the curso r 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

↑ &q 0L	Unlock configuration.
↑ &q 1L	Lock configuration.

ESCAPE SEQUENCE	MENU FIELD	ENTRY VALUE	X
↑ &k <x>A	AUTO LF	OFF	x=0
		ON	x=1
↑ &k <x>B	BLOCK	OFF	x=0
		ON	x=1
↑ &k <x>C	Caps Lock	OFF	x=0
		ON	x=1
↑ &k <x>J	FrameRate	60	x=0
		50	x=1
↑ &k <x>L	LocalEcho	OFF	x=0
		ON	x=1
↑ &k <x>M	MODIFY ALL	OFF	x=0
		ON	x=1
↑ &k <x>N	SPOW		
↑ &k <x>P	Caps Mode	OFF	x=0
		ON	x=1
↑ &k <x>R	REMOTE	OFF	x=0
		ON	x=1
↑ &s <x>A	xmitFnctn(A)	NO	x=0
		YES	x=1
↑ &s <x>B	SPOW(B)	NO	x=0
		YES	x=1
↑ &s <x>C	InhEolWrp(C)	NO	x=0
		YES	x=1
↑ &s <x>D	Line/Page(D)	LINE	x=0
		PAGE	x=1
↑ &s <x>G	IndHndShk(G)	NO	x=0
		YES	x=1
↑ &s <x>H	Inh DC2(H)	NO	x=0
		YES	x=1

DATA OPERATIONS

The following escape sequences control data transfer to and from the integral and external printers and display memory. Only effective if printer present.

ESC &p <a>d <b>d <c>d <Y>

Copies "Y" amount of data to destination devices "a", "b", and "c". As many destinations as desired can be specified.

a, b, and c	DEVICE
3	Display.
6	Integral printer.

y	ACTION
B	Copy the Line in which the cursor is located.
F	Copy the display screen from the line in which the cursor is located (cursor line) to the last displayed line.
M	Copy the contents of display memory from the cursor line to the end of memory.

ESC &p <x>^

Requests the status of device "x".

x	DEVICE
6	Integral printer.

ESC &K <x>S

Enables Expanded, Compressed, or Normal Character mode for the integral printer as designated by the character "X".

x	ACTION
0	Disable both Expanded and Compressed Character modes.
1	Initiate Expanded Character mode.
2	Initiate Compressed Character mode.

ESC &p <x>p <y>u <z>c

Performs the action specified by "z" on device "y".

z	ACTION
0	Generates 1 form feed.
1	Space "x" lines.
2-10	Generates 1 form feed.
11	Turn on Log Bottom mode.
12	Turn on Log Top mode.
13	Turn off any logging mode.
14	Print normal characters.
15	Print expanded characters.
16	Print compressed characters.
17	Turn on Report mode.
18	Turn on Metric Report mode.
19	Turn off any Report mode.

y	DEVICE
4	Internal or external printer depending on the PRINTER-CODE4 entry on the Terminal Configuration menu.
6	Integral printer.
3	Display.

DATA OPERATIONS (continued)

`ESC &p>x>W >data string>`	Transfers "x" bytes of the data string from the computer to the selected destination device in binary form.
`ESC &p W >data string>`	Transfers the data string, in ASCII form, from the computer to the printer selected as the destination device. The string is terminated either by the 256th byte or by an ASCII line feed character.

FORMAT MODE

`ESC [`	Starts a field.
`ESC ]`	Ends the field.

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.)
@	Disable the function keys and remove the function key labels from the screen.

To enable or disable the Function Control keys:

S	Disables the [SIDS], [MODES], and [USER KEYS] keys.
R	Enables the [SIDS], [MODES], and [USER KEYS] keys.

To define functions for the [RETURN] and function keys:

`ESC &f <attribute>a <key>k <label length>d <string length>l <label> <string>`

TERM	SYMBOL	MEANING	DEFAULT
Attribute	0	Normal (N)	0
	1	Local only (L)	
	2	Transmit only (T)	
Key	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	0 thru 16	Number of characters in the label.	0
String length	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.	

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

esc) B Selects line drawing as alternate character set.

Programming Examples

APPENDIX C

This appendix presents several utility programs, written in BASIC, that you may find useful when implementing an applications program to drive an HP 2622A Display 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 C-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 increment you wish used. For example, if you want the

```
>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 C-1. FORMIO1 Source Listing

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. FORMIO1 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 C-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 C-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    \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$[81]="x"'27"&dBzzz"'27"&d@x\c   c\c   c\c "'27"&dBz"'27&
    "&d@zc\"'27"&dBz"'27"&d@xxx"'27"&dBx\"
9530   L3$[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$[85]="c   c\c   c\c   c\"'27"&dBz"'27"&d@xxx"'27"&dBz\"
9570   L3$[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 C-2. LRGLINE Source Listing (Continued)

Index

K

L

M

N

O

P

R

S

T

U

V

X