HP 2392A
Display Terminal
Reference Manual

02390-90001

HEWLETT PACKARD

RADIO FREQUENCY INTERFERENCE STATEMENTS

FOR THE UNITED STATES

Federal Communications Commission (FCC)

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

WARNING

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

FOR EUROPE

Verband Deutscher Elektroniker (VDE)

This equipment has been certified to comply with the German standard VDE 0871 level B.

HP 2392A
Display Terminal
Reference Manual

Publications History

Revision 0 (original) ..... April 1984

This manual applies to terminals with serial number prefixes from 2420F onwards. If changes have been made to the terminal since this manual was printed, a yellow "manual updating supplement" defining them is supplied with this manual to explain how to adapt the manual to the modified equipment. The changes will be incorporated in the next revision of this manual.

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.

This document contains proprietary information which is protected by copyright. All rights are reserved. No part of this document may be photocopied, reproduced or translated to another language without the prior written consent of Hewlett-Packard Company.

Copyright © 1984 by HEWLETT-PACKARD COMPANY

Hewlett-Packard
Grenoble Personal Computer Division
5, avenue Raymond Chanas
38320 Eybens – France

Manual Part Number: 02390-90001 **Printed in U.S.A.** 04/1984

Table Of Contents

Section I - General Description Page

INTRODUCTION 1-1
KEYBOARD 1-6
FUNCTION KEYS AND SCREEN LABELS 1-8
STATUS LINE 1-10
CONFIGURING THE TERMINAL 1-10
TERMINAL PORTS 1-10
COMPUTER PORT DATA COMMUNICATIONS 1-12
EXTERNAL PRINTER PORT 1-12
SELF-TEST 1-12

Section II - Configuring The Terminal

INTRODUCTION 2-1
CONFIGURATION FUNCTION KEYS 2-1
TERMINAL CONFIGURATION MENU .. 2-2
Lock/Unlock Configuration Menus 2-11
Setting Terminal Configuration Parameters With Escape Sequence 2-12

Section III - Keyboard Control

INTRODUCTION 3-1
SELECTING THE MODE 3-1
Line Modify Mode 3-1
Modify All Mode 3-2
Character/Block Mode 3-3
Remote/Local Mode 3-4
Smooth Scroll Mode 3-5
Memory Lock Mode 3-5
Display Functions Mode 3-6
Auto Line Feed Mode 3-7
Caps Mode 3-8
Caps Lock Mode 3-8
Send Cursor Position Mode 3-9
Format Mode 3-9
Auto-keyboard Lock Mode 3-10
USER-DEFINABLE KEYS 3-10
Defining Keys Locally 3-11
Defining Keys Programmatically 3-12
Controlling The User Keys Menu Programmatically 3-14
Controlling The Function Key Labels Programmatically 3-14

Page

ENABLE/DISABLE KEYBOARD 3-14
[Print Enter] KEY 3-15
Enter 3-15
Print 3-15
SEND DISPLAY 3-16
[Reset Break] KEY 3-22
Soft Reset 3-22
Hard Reset 3-22
Break 3-23
[Extend char] KEY 3-23
[Select] KEY 3-23
[Stop] KEY 3-23
BELL 3-24
WAIT 3-24
MODEM DISCONNECT 3-24

Section IV - Alphanumeric Display Control

INTRODUCTION 4-1
CURSOR CONTROL 4-2
Home Up 4-2
Home Down 4-3
Move Cursor Up 4-3
Move Cursor Down 4-3
Move Cursor Right 4-3
Move Cursor Left 4-4
Roll Text Up 4-4
Roll Text Down 4-5
Next Page/Previous Page 4-5
MEMORY ADDRESSING SCHEME 4-6
Cursor Sensing 4-8
Cursor Positioning 4-8
Screen Relative Addressing 4-9
Memory Absolute Addressing 4-9
Cursor Relative Addressing 4-10
Combining Absolute And Relative Addressing 4-11
EDIT OPERATIONS 4-12
Insert Line 4-12
Delete Line 4-13
Insert Character 4-13
Delete Character 4-15
Clear Display 4-16
Clear Line 4-17

Table Of Contents (continued)

Page

SETTING AND CLEARING MARGINS .. 4-17
SETING AND CLEARING TABS 4-18
FORWARD TAB 4-19
BACK TAB 4-19
SCREEN BLANKING 4-20
DISPLAY ENHANCEMENTS 4-20
DESIGNING AND USING FORMS 4-23
DRAWING FORMS 4-24
FORMS MODE (FORMAT MODE) 4-26
Protected Fields 4-26
Unprotected Fields 4-26

Section V - External Printer Control

INTRODUCTION 5-1
Record Mode 5-1
DATA LOGGING 5-3
Top Logging 5-3
Bottom Logging 5-3
DISPLAY TO DEVICE ALPHANUMERIC DATA TRANSFERS 5-4
Advance Page 5-4
Advance Line 5-4
Copy All 5-5
Copy Page 5-5
Copy Line 5-5
Copy Display Memory 5-6
Copy Menu 5-6
Device Control Completion Codes 5-6
COMPUTER TO DEVICE DATA TRANSFERS 5-7
CONFIGURING THE EXTERNAL PRINTER 5-8
Printer Cabling 5-8
Filling In The Configuration Menu 5-10

Section VI - Data Communications

INTRODUCTION 6-1
HARDWIRE CONNECTIONS 6-2
MODEM CONNECTIONS 6-5

Page

Modem Considerations 6-5
INSTALLING A POINT-TO-POINT CONFIGURATION 6-7
CONFIGURING THE TERMINAL 6-7
POINT-TO-POINT PROGRAMMING INFORMATION 6-11
Start And Stop Bits 6-11
Parity Checking 6-11
Receive Buffer 6-12
Receive Errors 6-12
Local/Remote Mode 6-13
Full-duplex Operation 6-13
Pacing Mechanism 6-14
Transfers And Handshaking 6-15
Data Transfer Priority 6-16

Section VII - Status

INTRODUCTION 7-1
INTERPRETING STATUS 7-2
Terminal ID Status 7-2
Terminal Status 7-3
DEVICE STATUS 7-12

Section VIII - Error Messages And Self-Test

INTRODUCTION 8-1
ERROR MESSAGES 8-1
TERMINAL SELF-TESTS 8-2
Power-on Test 8-2
Terminal Test 8-3
Identify ROMS 8-4
Port 1 Test 8-5
Port 2 Test 8-6

Section IX - Terminal Maintenance Procedures

INTRODUCTION 9-1
CLEANING THE SCREEN AND KEYBOARD 9-1
BATTERY MAINTENANCE 9-1

Table Of Contents (continued)

Page

Appendix A - Escape Codes

TERMINAL CONTROL FUNCTION A-1
CURSOR CONTROL OPERATIONS A-4
CONFIGURATION OPERATIONS A-4
DATA TRANSFER OPERATIONS A-6
FORMAT MODE A-7
FUNCTION KEY AND ERROR MESSAGE OPERATIONS A-7
DISPLAY ENHANCEMENTS OPERATIONS A-9
CONTROL CHARACTERS A-9

Page

Appendix B - Character Set And Keyboards

CHARACTER SETS B-1
7-Versus 8 Bit Transmission To The Computer B-6
Data Transfers To The External Printer ... B-7
KEYBOARDS B-7
Diacriticals Characters B-8
LANGUAGES B-9

Illustrations

Figure		Page
1-1	2392A Display Terminal	1-2
1-2	Standard USASCII Keyboard	1-6
1-3	Function Keys And Screen Labels	1-7
1-4	Mode Selection Labels	1-8
1-5	Function Key Hierarchy	1-9
1-6	User Keys Definition Menu	1-11
1-7	Default User Key Labels	1-11
1-8	Sample User-defined User Key Labels	1-11
1-9	Screen Status Line	1-11
2-1	Terminal Configuration Menu	2-2
3-1	Screen-labeled Function Keys	3-3
3-2	User Keys Definition Menu And Default User Key Labels	3-12
4-1	The "Roll" Data Functions	4-4
4-2	Previous Page And Next Page Concepts	4-6
4-3	Row Addressing	4-7
4-4	Column Addressing	4-7
4-5	Character Insert With Margins	4-14
4-6	Character Delete With Margins	4-15
4-7	Menu After Entering Four Enhancements Escape Sequences	4-22
4-8	Line Drawing Set Elements	4-24
4-9	Sample Data Entry Form	4-25
4-10	Typical Form	4-26
5-1	External Device Port cabling	5-9
5-2	External Device Configuration Menu	5-11
6-1	Terminal Rear View	6-6
6-2	Terminal Cabling	6-6
6-3	Datacomm Configuration Menu	6-8
7-1	Primary Terminal Status Example	7-4
7-2	Secondary Terminal Status Example	7-8
7-3	External Device Status Byte	7-12
8-1	Screen Test Pattern	8-4
8-2	ROM Identification Listing	8-4
9-1	Removing The Batteries	9-2
B-1	Accessing ROMAN 8 And Line Drawing Set	B-1
B-2	Line Drawing Set Elements	B-6
B-3	Mapping Of ROMAN 8 Characters (accessed by the **[Extend char]** key)	B-9
B-4	American USASCII Keyboard	B-9
B-5	Swedish Keyboard	B-10
B-6	Norwegian Keyboard	B-10
B-7	French Keyboard	B-10
B-8	German Keyboard	B-11
B-9	English (UK) Keyboard	B-11
B-10	Spanish (European) Keyboard	B-11
B-11	Canadian (French) Keyboard	B-12
B-12	Canadian (English) Keyboard	B-12
B-13	Italian Keyboard	B-12
B-14	Dutch Keyboard	B-13
B-15	Finnish Keyboard	B-13
B-16	Danish Keyboard	B-13
B-17	Swiss (German) Keyboard	B-14
B-18	Swiss (French) Keyboard	B-14
B-19	Spanish (Latin American) Keyboard	B-14
B-20	Flemish Keyboard	B-15

Tables

Table		Page
1-1	Specifications	1-3
2-1	Terminal Configuration Menu Fields	2-3
2-2	Configuration Function Keys [f4], [f7], [f8]	2-11
3-1	[Enter] Key Operation	3-17
4-1	Display Enhancement Character	4-21
5-1	External Device Port Data Communications Cables	5-9
5-2	External Device Configuration Menu Fields	5-12
6-1	Data Communications Cables And Test Connectors	6-3
6-2	Datacomm Configuration Menu Fields	6-8
6-3	DC1/DC2 Handshake Used In Data Transfers	6-16
7-1	Status Characters	7-3
B-1	ROMAN 8 Characters Supported By 2392A	B-2
B-2	Non-displayable Control Character Set	B-3
B-3	National Characters Replaced By USASCII Characters In 7-bit Mode	B-4
B-4	Diacritical Marks In 8-bit Operation	B-5
B-5	Extended Roman Data Transfers To Printer	B-7

Preface

The HP 2392A Display Terminal is a versatile, ergonomic, character, line, or block mode CRT terminal.

This reference manual contains detailed information for configuring, testing, and using the terminal.

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

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

General Description

SECTION

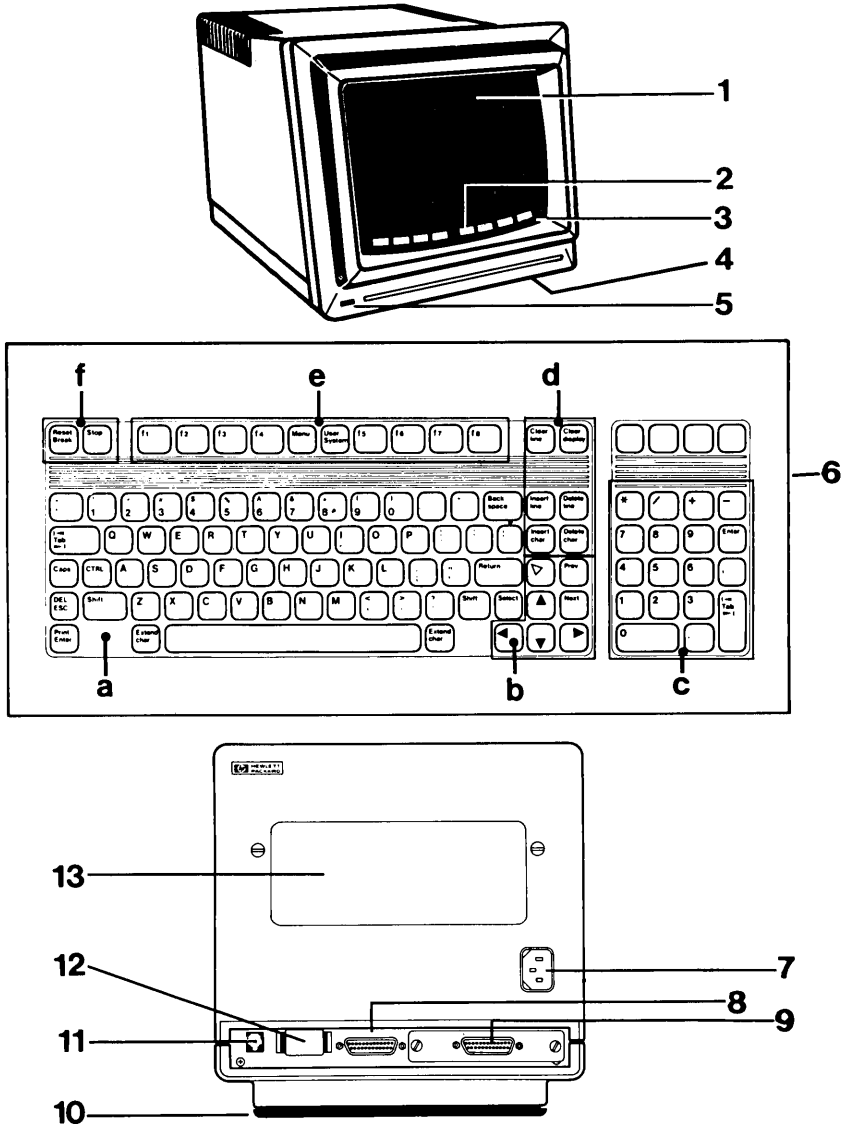
I

INTRODUCTION

The HP 2392A Display Terminal (figure 1-1) is a versatile general purpose alphanumeric CRT terminal that offers the following features:

- CRT screen size of 170 mm (6.7 inches) by 250 mm (9.8 inches).
- Screen capacity of 24 lines of up to 80 characters each (1920 characters total). Plus three additional lines; two for function key labels, and one for terminal status.
- Display enhancements (inverse video, blinking, underlining, half-bright and security).
- Character, line, or block mode operation.
- Eight user-definable 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 definable (up to 16 characters each). The character strings returned when the keys are pressed can be defined with up to 80 characters.
- 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-definable [**Return**] key.
- Line drawing character set.
- Screen labeled system function keys (for selecting operating modes and performing other terminal control functions).
- Extended Character Set (ROMAN 8, supports national keyboard layouts).
- 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 numeric key pad.
- Extensive self-test capability.

Table 1-1 lists the terminal specifications.



1 = display screen, 2 = display labels, 3 = display Status Line, 4 = display brightness rotary switch, 5 = power ON/OFF pushbutton, 6 = keyboard:
a = character set group, b = cursor and display control group, c = numeric group, d = edit group, e = function key group, f = terminal control group.
7 = power cable connector, 8 = port 1 (RS232C/HP422, computer port), 9 = port 2 Datacomm Module (optional printer port), 10 = swivel pedestal, 11 = keyboard cable connector, 12 = battery support, 13 = identification label.

Figure 1-1. 2392A Display Terminal

Table 1-1. Specifications

GENERAL (Standard terminal)	
Screen Size:	170 mm (6.7 inches) by 250 mm (9.8 inches).
Screen Capacity:	24 lines of 80 columns (1920 characters). Two additional lines for function key labels, plus a status line.
Character Generation:	7 by 11 character in 9 by 14 dot cell.
Character Size:	2.4 mm (0.094 inch) by 3.5 mm (0.138 inch).
Character Set:	ROMAN 8 (comprising 128 ASCII character set and extended Roman set) and Line Drawing.
Cursor:	Blinking-underline.
Display Enhancements:	Inverse, underline, blinking, half-bright, security.
Display Memory:	Up to four pages.
Display Modes:	Light characters on dark background.
Refresh Rate:	60 Hz.
Keyboard:	Detached, with 1.2 m (4 feet) coiled cable. 8 screen labeled keys; auto-repeat; N-key rollover. Numeric keyboard and independent user and cursor keys.
Operating Modes:	Local/remote Character/block (line/page) Forms/non-forms.
Transmission Modes:	Full duplex, asynchronous point-to-point. Optional via full duplex modems.
Electrical Interface:	RS232C/HP 422.
Data Rates:	110, 134.5, 150, 300, 600, 1200, 1800, 2400, 4800, 9600, 19200 baud.
Parity:	Selectable: Even, odd, zero, one, none.

Continued . . .

Table 1-1. Specifications (Continued)

PHYSICAL CHARACTERISTICS

Weight:	Display Monitor: 10.8 kg (23.8 lbs) Keyboard: 2.2 kg (4.9 lbs)
Dimensions:	Display Monitor: 325 mm wide by 362 mm deep by 317 mm high (12.8 inches by 14.2 inches by 12.5 inches) Keyboard: 455 mm wide by 220 mm deep by 40 mm high (17.9 inches by 8.7 inches by 1.6 inches)

POWER REQUIREMENTS

Input Voltage:	115 V (+ 10%, – 25%) at 47-66 Hz 230 V (+ 10%, – 15%) at 47-66 Hz
Power Consumption:	50 watts (standard)

HP 2392A OPTIONS

015	230 V 50 Hz operation	
049	ANSI X 3.64 operation	
093	8-bit parallel datacomm port	User installable port 2 connector
092	25-pin RS232C datacomm port	
101	Swedish	National language keyboards
102	Norwegian	
103	French	
104	German	
105	English, UK	
106	Spanish, European	
107	Canadian, French	
108	Canadian, English	
109	Italian	
110	Dutch	
111	Finnish	
112	Danish	
113	Swiss, German	
114	Swiss, French	
115	Spanish, Latin American	
116	Flemish	
160	Memory extension (adds an additional four pages of display memory)	
301	US modem cable (same as accessory 40242M)	Computer cables
302	European modem cable (same as 40242M)	
303	RS232C cable (same as 40242C)	

Continued . . .

Table 1-1. Specifications (Continued)

304	HP direct connect type 232 cable (same as 40242X)	Computer cables
305	EMP protect cable (same as 40242Y)	
306	HP direct connect type 422 cable (same as 40242P)	

HP 2392A ACCESSORIES

40210P	8-bit parallel datacomm module
40210R	25-pin RS232C datacomm module
40242A	Combined RS232C and HP422 adapter. Male (25-pin)/female (50-pin), 1 m (3 ft)
40242C	RS232C cable. Male (25-pin)/female (25-pin), 5 m (16 ft)
40242M	US/European modem cable. Male (25-pin)/male (25-pin), 5 m (16 ft)
40242P	HP direct connect type 422 cable. Male (25-pin)/male (5-pin), 5 m (16 ft)
40242X	HP direct connect type 232 cable. Male (25-pin)/male (3-pin), 5 m (16 ft)
40242Y	EMP protect cable. Male (25-pin)/male (25-pin), 5 m (16 ft)
40242Z	RS232C modem bypass cable. Male (25-pin)/female (25-pin), 5 m (16 ft)
13232U	Modem bypass cable. Female (25-pin)/female (25-pin), 1.5 m (5 ft)

Note: The 40242 type cables provided RFI filtering, and must be used to comply with the FCC and VDE notices at the front of this manual.

Cables used with 40242A

13222C	RS232C cable. Male (50-pin)/female (25-pin), 2 m (6.6 ft)
13222M	European modem cable. Male (50-pin)/male (25-pin), 5 m (16 ft)
13222N	US modem cable. Male (50-pin)/male (25-pin), 5 m (16 ft)
13222P	HP direct connect type 422 cable. Male (50-pin)/male (5-pin), 5 m (16 ft)
13222W	RS232C cable. Male (25-pin)/female (50-pin), 5 m (16 ft)
13222X	HP direct connect type 232 cable. Male (50-pin)/male (3-pin), 5 m (16 ft)
13222Y	EMP protect cable. Male (25-pin)/male (50-pin), 5 m (16 ft)

See Section V for printer cables

46010A	American USASCII	Keyboards
46010AC	Canadian, French	
46010AD	German	
46010AE	Spanish, European	
46010AF	French	
46010AH	Dutch	
46010AL	Canadian, English	
46010AM	Spanish, Latin American	
46010AN	Norwegian	
46010AQ	Swiss, French	
46010AR	Swiss, German	
46010AS	Swedish	
46010AU	English, UK	
46010AW	Flemish	
46010AX	Finnish	
46010AY	Danish	
46010AZ	Italian	

KEYBOARD

The terminal keyboard 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, 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.

Terminal Control Group – This group of keys is used to interrupt terminal-to-computer communications, and initialize the terminal.

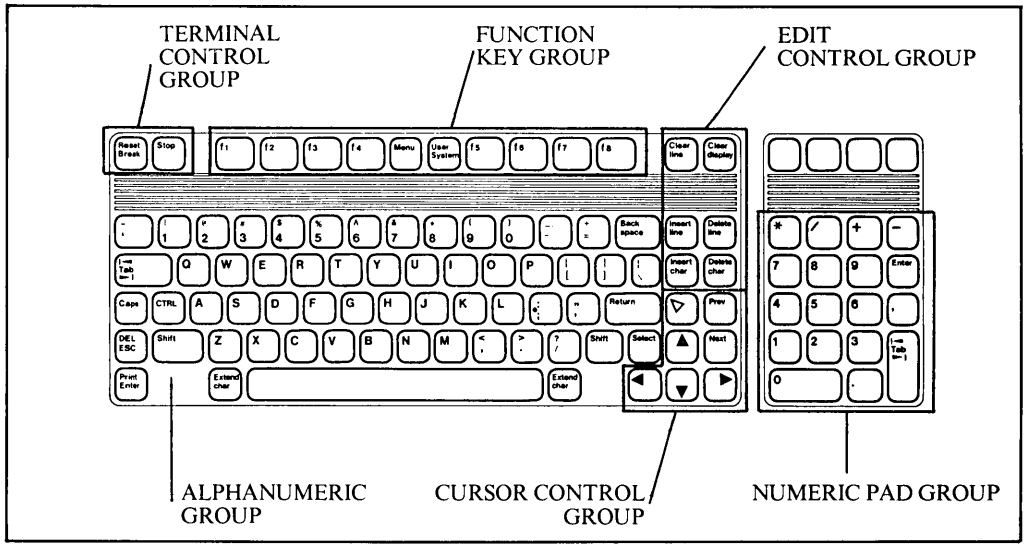


Figure 1-2. Standard USASCII Keyboard

The United States (USASCII) keyboard is the standard keyboard (see figure 1-2). As an option you can order any of the national keyboards listed in table 1-1. (The standard terminal is equipped with the ROMAN 8 character set that allows national characters to be accessed).

FUNCTION KEYS AND SCREEN LABELS

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 [User System] 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, accessing the terminal modes keys, and accessing the Device Control group of keys. The entire set of system function key labels for each terminal is illustrated in figure 1-5. Pressing [User System] always reinitializes [f1] through [f8] to the top row of functions (labels) shown in figure 1-5.

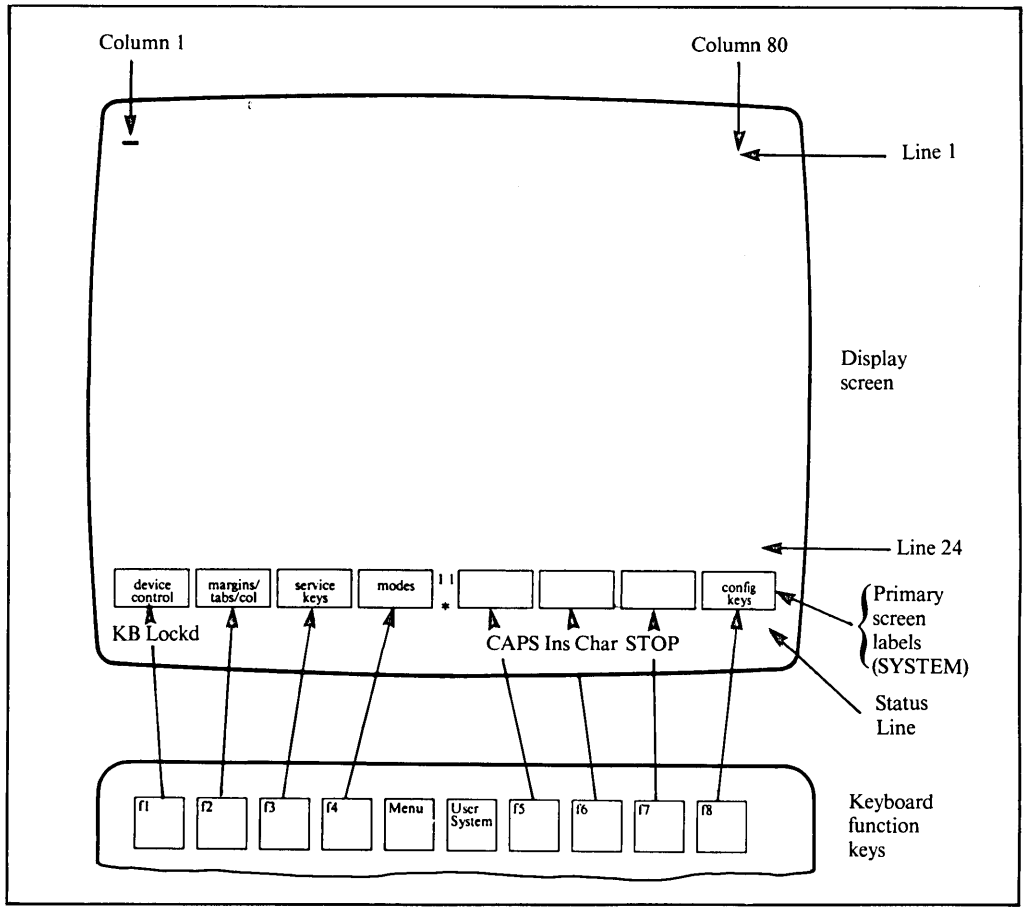


Figure 1-3. Function Keys and Screen Labels

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 key labels.
2. If a key label contains only uppercase letters, pressing the key will perform the function defined in the key label.

For example, the key corresponding to label **AUTO LF** 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 **[User System]**, then **[f4]** **[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.

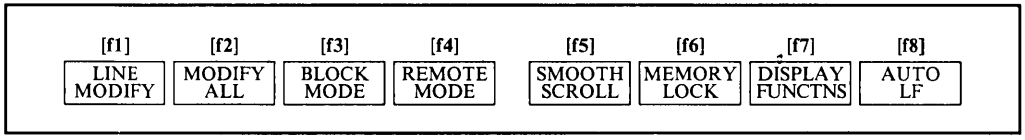
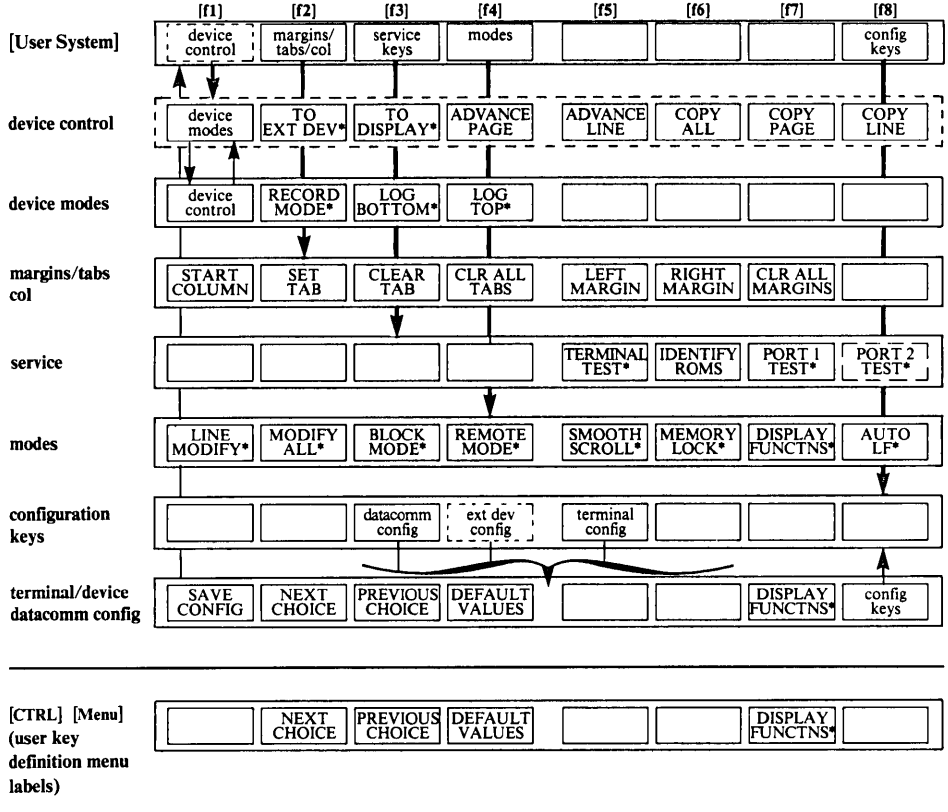


Figure 1-4. Mode Selection Labels



NOTES

1. An asterisk (*) in a label indicates an asterisk will be present in the label on the display when the function is active or enabled.
2. Labels surrounded by a dotted line do not appear if no printer port is present.

Figure 1-5. Function Key Hierarchy

When you press the [CTRL] and [Menu] 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. Then to enable the eight user keys, press the [Shift] and [User System] keys (or [CTRL] and [User System], or [Menu] twice). 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 [Menu] key removes the key labels from the screen. The user function keys, however, are still enabled. To re-enable the labels, press [User System] or [Menu] again.

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

STATUS LINE

This information is as follows:

KB Lockd – Keyboard locked out indicator. The keyboard is locked during a data transfer to the computer, when a handshake is pending, when the terminal is in record mode, and also by the program.

* – Modem indicator (displayed when modem is operational). See Asterisk field in datacomm configuration menu, Section VI.

CAPS – Keyboard set to capitals indicator (using [Caps] key).

Ins Char – Insert mode indicator (using [Insert char] key).

STOP – [Stop] key active indicator.

CONFIGURING THE TERMINAL

The terminal contains no physical straps or switches (other than the OFF/ON switch on the front 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.

TERMINAL PORTS

The standard terminal is equipped with one external port (Port 1), a combined RS232C/HP422 port used for the computer connection. The terminal may be optionally equipped with an interfacing module to provide a second port.

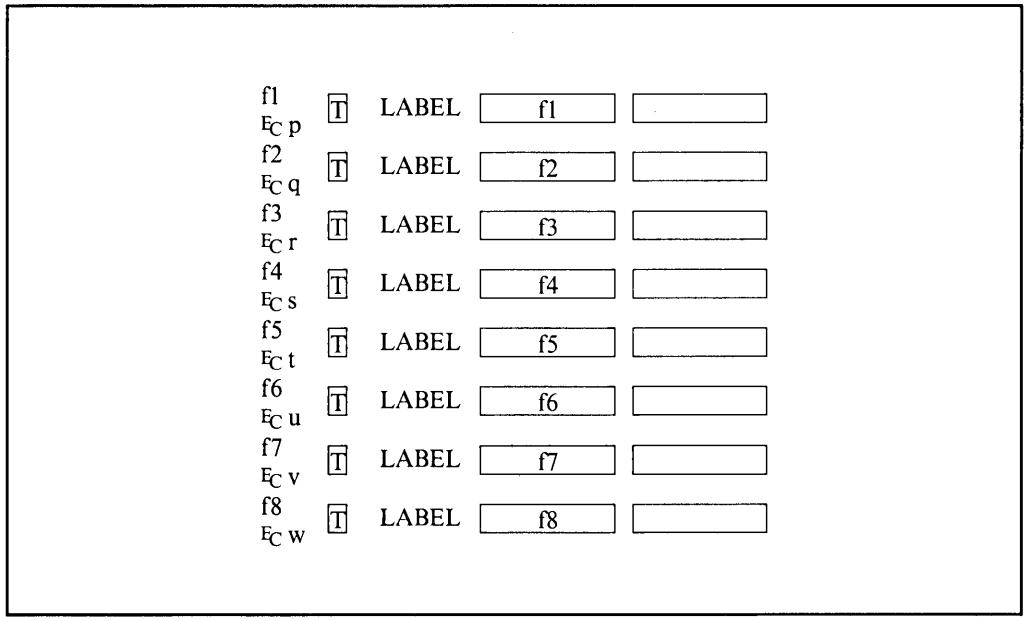


Figure 1-6. User Keys Definition Menu (default)

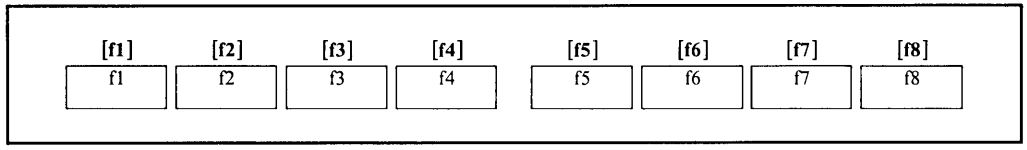


Figure 1-7. Default User Key Labels

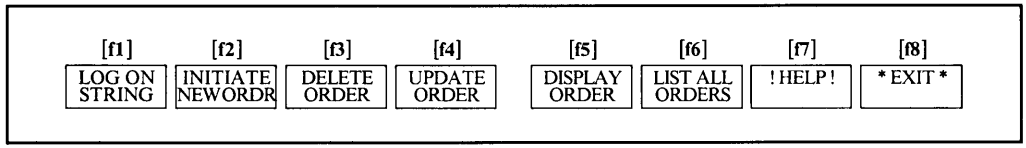


Figure 1-8. Sample User-defined User Key Labels

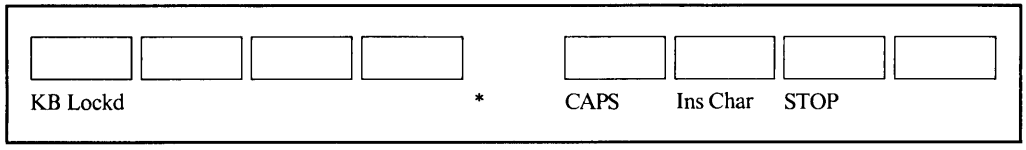


Figure 1-9. Screen Status Line

COMPUTER PORT DATA COMMUNICATIONS

The terminal can operate at speeds ranging from 110 to 19200 baud.

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

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

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

See Section VI for complete information.

EXTERNAL PRINTER PORT (Optional)

The external port at the rear of the terminal (when fitted) provides interfacing for serial or parallel printers. The port may be configured to operate from 110 to 9600 baud; choice of 0's, 1's, odd, even, or no parity; 0 to 255 nulls; choice of XON/XOFF, SRR, or CS/CB handshaking.

See Section V for complete information.

SELF-TEST

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

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

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

1. Port 1 or Port 2 test. This is a very flexible forms-selected test that verifies the integrity of the data communications ports. Loop-back via a test hood or a modem must be performed.
2. Terminal test. This self-test does a CRC (Cyclic Redundancy Check) verification of all ROM chips within the terminal, verifies the integrity of all RAM chips within the terminal, and then displays a test pattern on the screen. The test pattern includes all characters (and segments of the line drawing set) as well as all the character enhancements, plus the terminal status (see Section VIII).
3. Identify ROMs. This self-test generates a listing (on the display screen) of all installed ROMs.

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

Configuring the Terminal

SECTION II

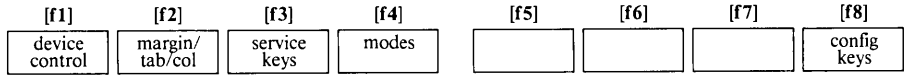
INTRODUCTION

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 function keys 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

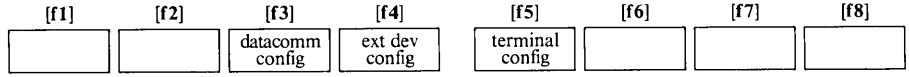
To gain access to the configuration menus through the keyboard, press the **[User System]** key. This causes the following label display across the bottom of the screen (where **[f1]**, **[f2]**, etc., refer to the function key corresponding to the label):



NOTE

The **"device control"** label is present only on terminals equipped with Port 2.

Pressing **config keys** (**[f8]**) changes the label display to



NOTE

The **"ext dev config"** label is present only on terminals equipped with Port 2.

Pressing **datacomm config** ([f3]), **ext dev config** ([f4]) or **terminal config** ([f5]) cause 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 config menu is described in Section VI, Data Communications. The ext dev config menu is described in Section V, Printer Control. The terminal configuration menu is described in this section. In addition, information is provided which will enable you to change the terminal configuration menu from the keyboard.

TERMINAL CONFIGURATION MENU

When you press the **terminal config** ([f5]) function key, the menu and function key display shown in figure 2-1 appear on the screen. Note that the menu 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, incoming data, including escape sequences, is stored in buffer (up to 256 bytes) and not processed and no data is transmitted, until the menu is exited. (Refer to Section VI for pacing considerations).

The menu contains a set of unprotected fields that you access using the [TAB► ◄] key. Except when the cursor is positioned in the fields labeled **"Terminal Id"**, **"Return Def"**, **"Fld Separator"**, or **"Blk Terminator"**, 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. The fields are listed in the order: terminal ID, language, keyboard function, datacomm functions and external device functions.

TERMINAL CONFIGURATION

Datacomm/ExtDev	PORT1/PORT2			Keyboard	USASCII		
TerminalId	2392A			Language	ENGLISH		
LocalEcho	OFF	CapsLock	OFF	Start Col	01	Bell	ON
XmitFnctn (A)	NO	SPOW (B)	NO	InhEolWrp (C)	NO	Line/Page (D)	LINE
InhHndShk (G)	NO	Inh DC2 (H)	NO	EscXfer (N)	NO		
FldSeparator	US	BlkTerminator	RS	ReturnDef	CR	TermMode	HP

[f1]	[f2]	[f3]	[f4]	[f5]	[f6]	[f7]	[f8]
SAVE CONFIG	NEXT CHOICE	PREVIOUS CHOICE	DEFAULT VALUES			DISPLAY FUNCTNS	config keys

Note: The "Datacomm/Ext dev" field is only displayed when the terminal is equipped with a second port.

The "Caps Lock" field only affects USASCII keyboard (ignored by terminal with national keyboard).

Figure 2-1. Terminal Configuration Menu

Table 2-1. Terminal Configuration Menu Fields

Terminal Id	This field allows the user to specify which Terminal Identification will be sent to the host computer upon a Terminal Id request (ESC * s ^) Values: Any string of up to five characters. Default: 2392A. Only the default value (2392A) is supported by Hewlett-Packard.
Keyboard	The terminal can be equipped with one of several keyboards (see table 1-1). This field allows the keyboard type to be selected, as the position of keys generating the same character differs from keyboard to keyboard, and the keycap labels are different (according to the country). The terminal includes as standard part of the ROMAN 8 Character Set, which includes the USASCII set plus an extended character set that supports the special characters associated with Western European languages (see Appendix B). Values: USASCII (United States), DANSK (Danish), NEDERLANDS (Dutch), SUOMI (Finnish), FRANÇAIS (French), ENGLISH CANADIAN, CANADIEN FRANÇAIS (French Can.), DEUTSCH (German), VLAAMS (Flemish), ITALIANA (Italian), NORSK (Norwegian), ESPAÑOL · EUR. (European Spanish), ESPAÑOL LAT. (Latin American Spanish), SVENSK (Swedish), SCHWEIZ-DEUTSCH (Swiss German), SUISSE ROMAN (Swiss French), UK (United Kingdom). Default: USASCII
Language	This field specifies in which language all function key labels and messages will be displayed. You may configure the terminal so that the value given to the language used for function keys and messages is different from the value given to the keyboard field. For example, a French programmer may use a USASCII keyboard and still have the function keys labels in French, in that case, all characters displayed, received or transmitted will be USASCII characters. Values: ENGLISH FRANÇAIS (French) DEUTSCH (German) ITALIANA (Italian) ESPAÑOL (Spanish) NEDERLANDS (Dutch) NORSK (Norwegian) SVENSK (Swedish) SUOMI (Finnish) DANSK (Danish) Default: ENGLISH

Continued . . .

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

Bell	This specifies whether the terminal's bell is enabled or disabled. When disabled, the bell will not sound when the cursor nears the end of line nor when the cursor advances from one field to the next in a formatted display. The bell will, however, still sound in response to an ASCII Bell control code (decimal 7; control G). Values: ON (Bell enabled, Ec & k 1 D) OFF (Bell disabled, Ec & k 0 D) Default: ON
Caps Lock	This field specifies whether the terminal generates the full 128-character ASCII set or only Teletype-compatible codes. ON (Ec&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 (Ec&k 0C) The terminal generates the full 128-character ASCII set of codes. Note that the Caps Lock field only affects USASCII keyboards, the field is ignored when a national keyboard is specified in the keyboard field. Default: OFF
SPOW (B)	This field specifies whether or not spaces entered through the keyboard will overwrite existing characters. NO (Ec&s 0B) Spaces entered through the keyboard will overwrite existing characters. YES (Ec&s 1B) Enable Space OverWrite (SPOW) latch. Once enabled, the SPOW latch 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 Ec&k sequence as follows: ON: Ec&k 1N OFF: Ec&k 0N When the SPOW latch is off, overwritting occurs as normal. When the SPOW latch is on, spaces entered through the keyboard move the cursor forward but do not overwrite existing characters. When a configuration or user key menu is displayed, the SPOW (B) field is ignored (equivalent to "NO"). Default: NO

Continued . . .

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

InhEolWrp (C)	This field specifies whether or not the end-of-line wrap is inhibited. NO (ESC&s0C) 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
Datacomm/Ext Dev	This field is present only when the terminal is equipped with two ports. It specifies which port (Port 1 or Port 2) is assigned to the host computer and which is assigned to the external printer. Values: Port 1/Port 2: Port 1 host computer, Port 2 printer. Port 1/Port 2: Port 1 printer, Port 2 host computer. Default: Port 1/Port 2 NOTE Attempting to assign a Centronics datacomm (printer) port to the host computer causes the error message "Function locked Press RETURN to clear" to be displayed.
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

Continued . . .

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

Start Col	If the line in which you are entering data is the bottom most 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 MODIFY ALL mode and you press **[Enter]** or **[Return]**, the data transmission from the terminal normally begins at the logical start-of-text pointer in the particular line. If the line has no logical start-of-text pointer, however, the data transmission begins at the designated start column. This 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 (EC&s 1A) The escape code sequences generated by control keys such as **[Insert char]** and **[Delete char]** are transmitted to the host computer. If local echo is ON, the function is also performed locally. NO (EC&s 0A) The escape code sequences for the major function keys are executed locally but NOT transmitted to the host computer. Note that with Xmit Functn (A) enabled, pressing the **"display functions"** function key will emit EC Y to a host computer (EC Z is never transmitted). Default: NO
InhHndShk (G) and Inh DC2 (H) (see Section VI)	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]**), or the **[Select]** key.

Continued . . .

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

- 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 D_1; terminal transmits block of data (D_1 handshake).
3. Computer sends D_1; terminal responds with D_2; computer responds with another D_1; terminal transmits block of data ($D_1/D_2/D_1$ handshake).

NOTE

When the $D_1/D_2/D_1$ handshake is enabled and the **line/page** field of terminal config is selected to be "line", a C_R or C_R L_F is transmitted after the D_2. If "line" is not selected, nothing is transmitted after D_2.

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

InhHndShk (G) = YES	Eliminates the use of the D_1 handshake (terminal will either use the $D_1/D_2/D_1$ handshake or no handshake at all).
Inh DC2 (H) = YES	Eliminates the use of the $D_1/D_2/D_1$ handshake (terminal will either use the D_1 handshake or no handshake at all).
Both = YES	No handshake.

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

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

Continued . . .

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

	If your terminal is connected to a Hewlett-Packard computer system, you will find that the default settings for these fields (both NO) are usually adequate for your purposes. If you are concerned about the specific type of handshake to be used for one or more of the particular types of block transfers, however, you should use the following summary to verify (or alter) the settings of the InhHndShk (G) and Inh DC2 (H) fields (also see Section VI): 1. Transmit only (T) user key or [Select] key in block line or character mode; or Cursor sense, terminal ID status, primary status, secondary status, display transfer initiated by E_Cd or device status request; or Device control completion code: InhHndShk (G) = NO InhHndShk (H) (ignored) → D_1 InhHndShk (G) = YES Inh DC2 (H) = NO → $D_1/D_2/D_1$ InhHndShk (G) = YES Inh DC2 (H) = YES → no handhskake 2. [Enter] key in block mode; or Transmit only (T) user key or [Select] key in block page mode: InhHndShk (G) (ignored) Inh DC2 (H) = NO → $D_1/D_2/D_1$ Inh DC2 (H) = YES → no handshake 3. [Enter] key in character mode; or [Enter] or [Return] key in modify mode: InhHndShk (G) = YES Inh DC2 (H) = NO • → $D_1/D_2/D_1$ Any other combination → no handshake

Continued . . .

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

	The G and H selections can be temporarily selected by the following escape sequences: E_C&s 0G disables G E_C&s 0H disables H E_C&s 1G enables G E_C&s 1H enables H Defaults: InhHndShk (G) = NO Inh DC2 (H) = 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 (E_C&s 0D) When operating in block mode, the terminal will transmit data a line at a time. Page (E_C&s 1D) When operating in block mode, the terminal will transmit the data in the display memory, either from the beginning of the memory or the current cursor position (see the Enter key, Section III). For detailed description of the differences between block line and block page mode, refer to "ENTER KEY" in Section III of this manual. Default: LINE
FldSeparator	When you press the **[Enter]** key while the terminal is in block page mode and display memory contains a formated 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 (or ROMAN 8 character in 8-bit operations) Default: U_S
Blk Terminator	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: R_S

Continued . . .

Table 2-1. Terminal Configuration Menu Fields

RETURN def	This field specifies the definition of the [Return] key. The default definition is an ASCII $^{C}_{R}$. The definition may consist of up to two characters. If the second character is a space, it is ignored. Default: $^{C}_{R}$
Esc Xfer (N)	This field controls the transfer of escape sequences to a printer, and has no effect if there is no second port. **YES** ($^{E}_{C}$&s 1N) = When transferring data from display memory to an external printer, each line automatically starts with $^{S}_{1}{}^{E}_{C}$&d@ (this is to return to the primary character set and to stop any character enhancements). In addition, escape sequences relating to the display (such as those specifying display enhancements, format mode fields, and alternate character sets) are sent to the external printer if encountered within the data. NO ($^{E}_{C}$&s 0N) = Escape sequences relating to the display are not sent to the external printer (and $^{S}_{1}{}^{E}_{C}$&d@ is not sent). NOTE: The Esc Xfer (N) field only affects data transfers between display memory and an external printer. It does NOT affect <Esc> &pW data transfers that go directly from the host computer to the external printer. Default: NO
TermMode	Selects one of three primary terminal operating modes. Except for terminals ordered with Option 049, the terminal should be operated in HP mode, which is the default mode. HP = **Terminal operates as a standard 2392A as described in this manual.** ANSI = Terminal recognizes and executes a subset of the terminal escape sequences specified in the American National Standards Institute documents X3.41-1974 and X3.64-1979, in addition to some of the standard and HP escape sequences. EM52 = Terminal emulates a VT52 compatible terminal by recognizing and executing escape sequences written for a DEC VT52 terminal. Default: HP

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, and the terminal returns to normal operations with the SYSTEM labels displayed.

While the terminal configuration menu is displayed on the screen, the [f4], [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 control characters in the **Return Def, Fld Separator,** or **Blk Terminator** fields. Control characters may be accessed using [CTRL] and the relevant alphabetic key (see Appendix B). 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:

[f1]	[f2]	[f3]	[f4]	[f5]	[f6]	[f7]	[f8]
		datacomm config	ext dev config	terminal config			

The "**ext dev config**" label is blanked if Datacomm Module not installed (Port 2)

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 and the "Function locked Press RETURN to clear" error message to be displayed. Note that when the configuration menus are locked, the **MODIFY ALL** ([f2]), **BLOCK MODE** ([f3]), **REMOTE MODE** ([f4]), **AUTO LF** ([f8]) mode selection keys are also locked.

To lock the menus, use the following escape sequence:

$_{EC}$&q 1L

To unlock the menus, use the following escape sequence:

$_{EC}$&q 0L

This is the only escape sequence to be retained in non-volatile memory.

Setting Terminal Configuration Parameters With Escape Codes

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

Parameter Name As Shown in Menu	Type of Escape Sequence Used
Local Echo **Caps Lock** **Bell** **Spow Latch**	ESC & k
XmitFnctn (A) **SPOW (B)** (enables SPOW latch) **InhEolWrp (C)** **Line/Page (D)** **InhHndShk (G)** **Inh DC2 (H)** **Esc Xfer (N)**	ESC & s

The ESC&k and ESC&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 value of the **Local Echo**, **Caps Lock** or **SPOW** parameters, use an escape sequence of the following form:

Local Echo = OFF: ESC & k 0L
Local Echo = ON: ESC & k 1L

Caps Lock = OFF: ESC & k 0C
Caps Lock = ON: ESC & k 1C

Bell = OFF: ESC & k 0D
Bell = ON: ESC & k 1D

Spow Latch = OFF: ESC & k 0N
Spow Latch = ON: ESC & k 1N

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

ESC&k 1l 1C
ESC&k 1c 1L

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

XmitFnctn (A) =	**NO:**	**E_C &s 0A**
XmitFnctn (A) =	**YES:**	**E_C &s 1A**
Spow (B) =	**NO:**	**E_C &s 0B**
Spow (B) =	**YES:**	**E_C &s 1B**
InhEolWrp (C) =	**NO:**	**E_C &s 0C**
InhEolWrp (C) =	**YES:**	**E_C &s 1C**
Line/Page (D) =	**LINE:**	**E_C &s 0D**
Line/Page (D) =	**PAGE:**	**E_C &s 1D**
InhHndShk (G) =	**NO:**	**E_C &s 0G**
InhHndShk (G) =	**YES:**	**E_C &s 1G**
Inh DC2 (H) =	**NO:**	**E_C &s 0H**
Inh DC2 (H) =	**YES:**	**E_C &s 1H**
Esc Xfer (N) =	**NO:**	**E_C &s 0N**
Esc Xfer (N) =	**YES:**	**E_C &s 1N**

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

E_C &s 1d 0g 1H
E_C &s 0g 1h 1D
E_C&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. Except for two keys ([**Reset Break**] and [**Stop**]), the overall keyboard can be logically divided into a character set group, a numeric pad, a cursor control group, an edit group, and a function key group. The function key group includes eight keys marked [**f1**] through [**f8**] and the keys [**Menu**] and [**User System**]. 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 from the function keys are displayed across the bottom of the screen (figure 3-1).

SELECTING THE MODE

Pressing keys [**User System**], then [**f4**] enables the mode selection keys and changes the [**f1**] through [**f8**] screen labels to the following:

[f1]	[f2]	[f3]	[f4]	[f5]	[f6]	[f7]	[f8]
LINE MODIFY	MODIFY ALL	BLOCK MODE	REMOTE MODE	SMOOTH SCROLL	MEMORY LOCK	DISPLAY FUNCTNS	AUTO LF

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.

Line Modify Mode ([f1])

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 ([f2])

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: E_C&k 1M
DISABLE: E_C&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 sequence, however, the designator is NOT altered.

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

NOTE

Modify mode can be used with the default terminal configuration parameters (see table 2-1). For instance: when using modify mode you will usually want the data block (not a D_1 handshake control code) to be sent when you press **[Return]** or **[Enter]**, and the default configuration parameters disable the $D_1/D_2/D_1$ handshake. Also when moving the cursor to the erroneous line you will normally not want the cursor movement escape sequences to be transmitted to the computer (as these will be interpreted as new data, and cause a new error message), and the default configuration parameters disable the transmission of escape sequences.

The modify all and modify line functions are ignored if the terminal is in format mode or block mode.

Character/Block Mode ([f3])

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. Control codes (such as C_R and L_F) are also 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, control codes (such as C_R and L_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.

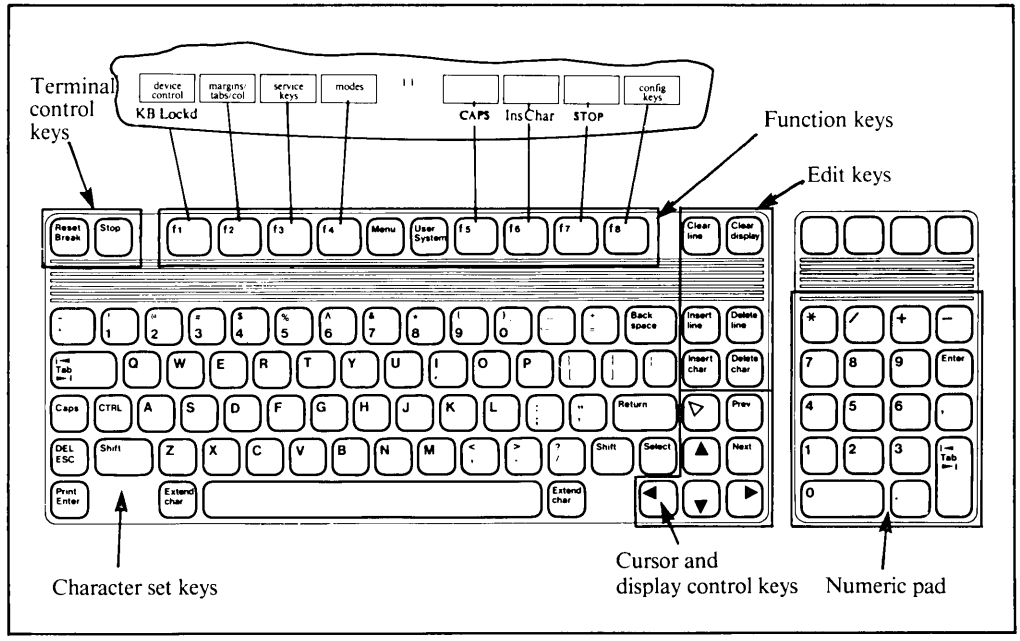


Figure 3-1. Screen-labeled Function Keys

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

ENABLE: EC&k 1B
DISABLE: EC&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.

Remote/Local Mode ([f4])

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 character 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: EC&k 0R
Remote: EC&k 1R

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

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

Smooth Scroll Mode ([f5])

When smooth scroll is enabled, rolling data up and down the screen (using [Shift] [▲] and [Shift] [▼]) is done smoothly, rather than "jumping" a line at a time.

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

ENABLE: EC&k 1[
DISABLE: EC&k 0[

Memory Lock Mode ([f6])

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 move the cursor to the first line on the screen 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 the error message "MEMORY FULL Press RETURN to clear". 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 display], 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. Press keys [Shift] [▲] (roll up) 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: EcI
DISABLE: Ecm

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 ([f7])

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

- In local mode, it displays control codes and escape sequences but does not execute them. For example, if you press the [◄] key the terminal displays EcD on the screen but does not perform the "cursor left" function.
- In remote mode, it transmits control codes and escape sequences but does not execute them locally. For example, if you press [Shift] and [▲] (roll up) the terminal transmits an EcS but does not perform the "roll up" function. If local echo is enabled (ON) then the EcS is also displayed on the screen.

There are two exceptions to the foregoing descriptions:

1. When you press the **DISPLAY FUNCTNS** key, the EcZ (which disables display functions mode) or EcY (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 enable) is transmitted, and (if echoed) it is executed and displayed and the terminal also performs a line feed.

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: EcY
DISABLE: EcZ

NOTE

There is interaction between display functions and the **Xmit=Fnctn (A)** field of the terminal configuration menu. If **XmitFnctn (A)** is on (YES), the **DISPLAY FUNCTNS** key transmit EcY but not EcZ.

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.

Auto Line Feed Mode ([f8])

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

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, see table 3-1).

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: Ec&k 1A
DISABLE: Ec&k 0A

When you enable or disable auto line feed mode using the **"AUTO LF"** key, you also alter the content of the **"AUTO LF"** field in 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 **"AUTO LF"** field in active memory.

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

Caps Mode ([Caps] key control)

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. On USASCII keyboard, 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: EC&k 1P
DISABLE: EC&k 0P

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

When enabled, **"CAPS"** is displayed in the status line.

Caps Lock Mode (Caps Lock menu control)

The caps lock mode only affects a USASCII keyboard (the **"Caps Lock"** field is ignored with a national keyboard).

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: EC&k 1C
DISABLE: EC&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.

Send Cursor Position Mode

Some programs need to know where the cursor is located before it is moved to the beginning of the data to be transmitted. Other application programs may need to do some other operation before sending data. Currently some application programs get the cursor location by sending a cursor sense escape sequence (EC a) after the terminal has sent a DC2. The terminal will then send the cursor address, the host sends a DC1, and the terminal then sends the data. But this does not work when handshaking is disabled.

Therefore, the Send Cursor Position (SCP) mode has been created. It is accessible by escape sequence only.

ENABLE: EC&x 1C
DISABLE: EC&x 0C

When disabled, the terminal will act as previously defined. When enabled, and the **[Enter]** key or **[Select]** key or a user defined key set to transmit (T) is pressed, the cursor position in the form of EC&axxcyyyR is inserted at the beginning of the block sent to the computer. The position is of the same form as that returned from an absolute cursor position sense status request (EC a) described in Section IV. The position value is that of the cursor BEFORE any repositioning is done, regardless of strap settings. The block transfer obeys the usual handshaking conventions.

The Send Cursor Position Mode does NOT apply to the **[Enter]** and **[Return]** keystrokes for line modify and modify all modes. These modes are defined only when in character mode and are intended to simulate a block mode operation.

A soft reset will leave the Send Cursor Position Mode unchanged. A hard reset will set the Send Cursor Position Mode to its default (disabled) setting.

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 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: EC W
DISABLE: EC X

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

Auto-keyboard Lock Mode

When a terminal is connected to a packet switching network (using X.25 protocol) via a controller/multiplexer, it is necessary to ensure that the packet sent is received and acted upon before another is sent (from the terminal). In order to achieve this, the keyboard must automatically lock such that it can only be unlocked by the receiving host. This is the Auto-keyboard Lock. This mode is accessible by escape sequence only.

When disabled (default), the terminal acts normally.

When enabled, any user key configured to be transmit only (T) or the [Enter] key or [Select] key causes the keyboard to be locked after the data transfer has taken place and the message "KB Lockd" to appear in the screen Status Line. The host computer can then unlock the keyboard (with E_Cb) when it is ready to receive more data; when the keyboard is unlocked, the keyboard buffer is flushed (cleared).

A soft Reset will not affect the auto-keyboard lock mode, but will still unlock the keyboard. A hard Reset or Power-on will put this mode in its default state (disabled).

Note that the Send-display sequence (E_Cd) does not simulate the [Enter] keystroke, and is not affected by the auto-keyboard lock mode (i.e. it does not lock the keyboard). The function key triggering sequence (E_C&f<key>E, where <key> is the function key number) is defined to act as though the key was depressed, so it is affected by this mode (i.e. it locks the keyboard).

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

ENABLE: E_C&k 1K
DISABLE: E_C&k 0K

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 means that:

1. You can assign to each key a string of ROMAN 8 alphanumeric characters and/or control codes (such as C_R or L_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 (when the key is subsequently used).

The definition of each user key may contain up to 80 characters (alphanumeric characters, control characters, and explicit escape sequence characters). The definition is saved after power off in the non-volatile memory, but is reset to E_Cp-E_Cw following a hard reset from the host computer using E_CE.

Defining Keys Locally

To define one or more keys from the keyboard, press the [CTRL] and [Menu] keys simultaneously (or use EcJ). 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▶ ◀] key.

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 as though it were entered through the keyboard (N).

  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 physically pressed. When entering characters into this field you may use display functions mode to display control characters. Note that when the menu is displayed the alternate line drawing set and display enhancements are inhibited.

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] key can be used for including CR codes (with display functions mode enabled) in key definitions. If auto line feed mode is also enabled, the [Return] key will generate a CR LF, 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 have finished defining all the desired keys, press the [User System] key (the user keys menu disappears from the screen). When you press [Shift] [User System] or [CTRL] [User System], or [Menu] twice, or enter EcjB, 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:

E_C&f <attribute> a <key> k <label length> d <string length> L <label> <string>

where:

<attribute>	= 0: normal 1: local only 2: transmit only	(0 is the default)
<key>	= 1–8: f1 - f8, respectively	(1 is the default)
<label length>	= 0–16 (0-32 diacritical combinations)	(0 is the default)
<string length>	= 0–80 (0-160 diacritical combinations)	(1 is the default) (–1 causes field to be erased)

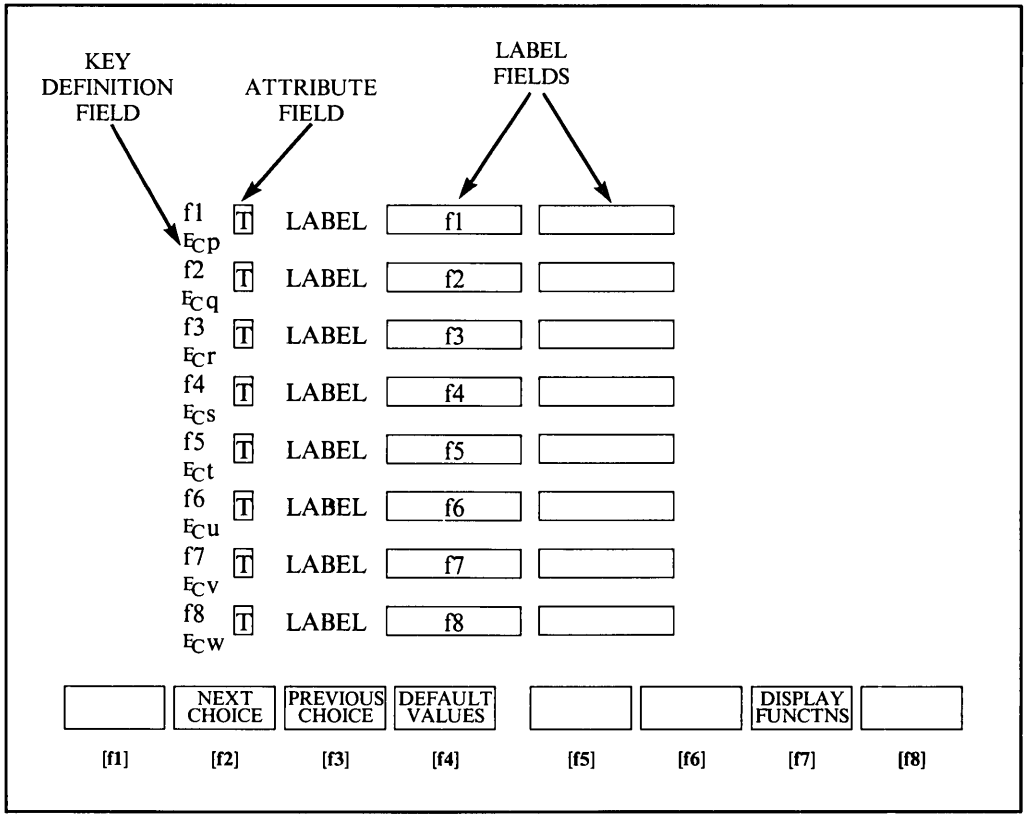


Figure 3-2. User Keys Definition Menu and Default User Key Labels

<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, control codes such as C_R and L_F, and explicit escape sequence characters) in the label string must not exceed 16, and in the definition string must not exceed 80.

NOTE

One function key may be used to trigger another using the "E_C&f<key>E" sequence. However, only one key may be triggered. That is, only one "E_C&f<key>E" sequence may be included in a key's <string>.

Example: Assign **LOG-ON** (6 characters) as the label and **HELLO USER.ACCOUNT** Cr (19 characters) as the definition for the **[f5]** user key, and enable the user keys. The key is to have the default attribute **"N"**.

E_C&f5k6d19LLOG-ON HELLO USER.ACCOUNT C_R E_C&jB

After issuing this 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 C_R

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 approriate terminator to the string. The E_C&jB sequence turns on the user labels.

Note that in remote mode when the user key definition menu is displayed, pressing **[Enter]** or sending escape sequence E_Cd (transmit a block) can be used to transfer a copy of the menu to the computer. The returned escape sequence is identical to the one used by the host to configure the user keys (E_C&f...). In addition, in block page mode, eight escape sequences are returned (one for each of the eight keys). In character mode and block line mode, only the escape sequence for the user key in whose field the cursor is located is returned. For example, if the default menu is displayed with the cursor in one of the fields of key **[f1]**, the following escape sequence is returned to the computer in character mode or block line mode:

E_C&f2a1k16d2L f1 E_Cp

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: E_Cj
REMOVE MENU: E_Ck

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 pressing the **[Menu]** key).
- You can enable the mode selection keys (this is the equivalent of pressing the **[User System]** key, then **[f4]** key).
- You can enable the user keys (this is the equivalent of pressing **[Shift]** and **[User System]**, or **[CTRL]** and **[User System]**, or **[Menu]** twice).
- You can "lock" the current set of labels on the screen (i.e. disable the **[User System]** key and **[Menu]** key).
- You can reenable the **[User System]** key and **[Menu]** key.

The escape sequences are as follows:

E_C**&j@**	Enable the user keys, and remove all key labels and status line from the screen.
E_C**&jA**	Enable and display the **"modes"** labels.
E_C**&jB**	Enable and display the user function key labels.
E_C**&jR**	Enable keys **[User System]** and **[Menu]** and label **"modes"**.
E_C**&jS**	Disable keys **[User System]** and **[Menu]** and label **"modes"**.
E_C**&j**<**xx**> L <**message**>	Remove the key labels from the screen and display the character string <**message**> (which consists of <**xx**> characters; where <**xx**> may be up to 160 characters, 320 diacritical combinations). The terminal beeps at the end of the displayed <**message**>. The labels are restored by "E_C&jC" or pressing **[Return]**. Note that all control characters (except C_R and L_F) included in the <**message**> are displayed. Also if the number of characters in the <**message**> is less than the specified <**message length**>, characters entered on the keyboard (if echoed) will be displayed on the labels line until the specified number is reached.
E_C**&jC**	Remove your <**message**> from the screen and restore the current key labels.

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: EcB
DISABLE: Ecc

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

When the keyboard is disabled, the "KB Lockd" indication appears in the status line.

[Print Enter] KEY

Enter

When the terminal is in remote mode, pressing the **[Print Enter]** key on its own sets pending a block transfer of data from display memory to host computer. This locks the keyboard (causing the message "KB Lockd" to appear in the display's status line) until the resultant data transfer is complete or until the host unlocks the keyboard if auto-keyboard lock mode is on.

When the Send Cursor Position Mode is enabled, and the **[Enter]** key is pressed, the cursor position is inserted at the beginning of the block sent to the host computer (see Send Cursor Position Mode).

The type of handshaking used and precisely what data is 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. (Sections VI details the handshake operations).

Print

Pressing the **[Print Enter]** key with **[Shift]** causes the cursor to be "homed up" and the display memory to be copied to the external printer (if installed). If no "to" device has been specified the message "No "TO" device Press RETURN to clear" is displayed. If only the display was set as the "to" device the message "Source = Destination Press RETURN to clear" is displayed. If a printer module is not installed, no action is taken.

SEND DISPLAY (E_Cd)

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:

E_Cd

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 (see Section VI for details):

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

The E_Cd sequence also temporarily disables the keyboard (with "KB Lockd" in the status line) so that the **[Enter]** key cannot be used until the current data transfer is completed. If the E_Cd sequence is received while an **[Enter]** key data transfer is in progress, the escape sequence is ignored.

Note that an E_Cd 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 E_Cd, 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).

Table 3-1. **[Enter]** Key Operation

CHARACTER MODE

The cursor is repositioned to left margin.

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.

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 C_R (or a C_R L_F 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 C_R (or a C_R L_F 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 the configured block terminator followed by a C_R (or a C_R L_F if auto line feed mode is enabled).

The type of handshaking used is determined as follows:

InhHndShk (G) = YES
Inh DC2 (H) = NO
} $D_1/D_2/D_2$

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.

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 C_R (or a C_R L_F 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 C_R (or a C_R L_F if auto line feed mode is enabled). The cursor remains positioned immediately following the terminator.

Continued . . .

Table 3-1. [Enter] Key Operation (Continued)

If there is no data to be transmitted, the terminal sends a block terminator followed by a C_R (or a C_R L_F if auto line feed mode is enabled). The C_R 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 } $D_1/D_2/D_1$

Any other combination → no handshake

BLOCK LINE MODE

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

Inh DC2 (H) = YES

The cursor is repositioned to column 1 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.

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 C_R (or a C_R L_F 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 C_R (or a C_R L_F 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 C_R or C_R L_F is transmitted. The cursor is not moved.

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

Continued . . .

Table 3-1. **[Enter]** Key Operation (Continued)

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".

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.

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 C_R (or a C_R L_F 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 C_R (or a C_R L_F 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 C_R (or a C_R L_F if auto line feed mode is enabled). The C_R that is transmitted has no effect on the terminal locally, and the cursor remains unmoved.

The type of handshaking used is determinated as follows:

InhHndShk (G) (ignored)
Inh DC2 (H) = NO → $D_1/D_2/D_1$

Inh DC2 (H) = YES → no handshake

Block Page Mode

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

Inh DC2 (H) = YES

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

Inh DC2 (H) = NO

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

Continued . . .

Table 3-1. [**Enter**] Key Operation (Continued)

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 a $_{CR}$ $_{LF}$ followed by the configured block terminator.

The type of handshaking used is determined as follows:

InhHndShk (G) (ignored)

Inh DC2 (H) = NO → $D_1/D_2/D_1$

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.

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 → $D_1/D_2/D_1$

Inh DC2 (H) = YES → no handshake

Continued . . .

Table 3-1. [Enter] Key Operation (Continued)

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.

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 C_R (or a C_R L_F if auto line feed mode is enabled). If **LocalEcho = OFF**, the cursor is repositioned to the column at which the transmission began, otherwise the cursor is repositioned to column 1. A line feed is performed if auto line feed mode is enabled.

If the operation is terminated by encountering a block terminator, the terminal sends a block terminator followed by a C_R (or a C_R L_F if auto line feed mode is enabled).

The cursor remains positioned immediately following the terminator.

The type of handshaking used is determined as follows:

InhHndShk (G) = YES
Inh DC2 (H) = NO } $D_1/D_2/D_1$
Any other combination → no handshake

The [Enter] key can be triggered from the host by the following sequence: E_C&f-1E

[Reset Break] KEY

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 buffer. If the datacomm configuration menu "RecvPace" field is set to "XON/XOFF" and an XOFF had been sent (before the soft reset), an XON is sent.
7. Turns off record mode, if on.

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

From the keyboard, you perform a soft reset by simultaneously pressing the [**Shift**] and [**Reset Break**] keys.

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

$E_C g$

Hard Reset

A hard reset has the same effect as turning the terminal's power off and then back on; except that the power-on self-test is not performed. A hard reset from the computer using $E_C E$ will reset the user definable softkeys to their default values ($E_C p$, $E_C q$ to $E_C w$).

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 the 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. If the datacomm configuration menu "RecvPace" field is set to "XON/XOFF", an XON is sent.
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, data logging, smooth scroll, memory lock mode, and modify line.
   Disables display functions mode, caps mode, data logging, and modify line.
   Resets the left margin to column 1.
   Resets the right margin to column 80.
   Turns off the "insert character" function edit.
   Resets the User Keys to default values (only using hard reset from computer; keys retain pre-set value after hard reset from the keyboard using **[Shift]** **[CTRL]** **[Reset Break]**).
   Resets record mode.

From the keyboard, you perform a hard reset by simultaneously pressing the **[Shift]**, **[CTRL]** and **[Reset Break]** keys.

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

EcE

Break

In remote mode, pressing the **[Reset Break]** key on its own transmits a 200 ms space on the asynchronous "transmit data" communications line. This serves as a "break" signal to interrupt computer operations.

[Extend char] KEY

Pressing this key allows access to some of the ROMAN 8 characters (National and displaced USASCII) which are not directly accessed on the keyboard. This key is only active if 8 bit transmission is used (any attempt to use this key in 7 bit transmission will cause the beeper to sound), and is a non-latching key.

[Select] KEY

This key is treated the same as a "Transmit" softkey and sends EcE&P when pressed. It can be used to easily select a field in a form or a menu. Typically, the user will move the cursor to the requisite field, and then press **[Select]**. See Section VI for handshaking considerations.

[Stop] KEY

Alternatively stops and resumes the transfer of data from the datacomm input buffer to the screen. This key is executed only if the X-ON/X-OFF receive hanshake pacing is active.

NOTE

Pressing the **[Stop]** key will not necessarily result in sending an **XOFF** to the host, the **XOFF** is only sent when the input buffer pointer reaches the upper limit. So, pressing this key is not equivalent to pressing **[CTRL]** **[S]** / **[CTRL]** **[Q]**.

BELL

The terminal includes a speaker (located in the display monitor) for sounding an audible tone in response to the Bell control code and for alerting the terminal operator when certain error conditions occur (the tone and duration are fixed and cannot be changed).

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 a 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:

E_C@

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

Note that while an E_C@ is in effect, the cursor disappears from the screen, the passing of data from the keyboard to the screen is delayed, and the passing of data from the datacomm buffer 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> E_C@ E_C@ <BELL>

MODEM DISCONNECT

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

Alphanumeric Display Control

SECTION

IV

INTRODUCTION

This section discusses the alphanumeric display control of the terminal.

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, the code for the character is recorded at the associated position in display memory, the character is displayed at the cursor position, 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, line blinking and security fields display enhancements.
- Change from one character set to another (e.g. Line Drawing).

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

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

CURSOR CONTROL

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

Home Up ([▷])

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

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

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

ESC H or ESC h

When memory lock is enabled, the [▷] key rolls the text down as far as possible below the locked area of the screen, instead of below the top of the screen, and leaves the cursor positioned at the 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 non-locked text down (if required).

Home Down ([Shift] [▷])

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:

$E_C F$

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:

$E_C A$

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 position in the top row of the screen.

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

$E_C B$

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:

$E_C C$

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:

$E_C D$

Roll Text Up ([Shift] [▲])

Each time you press keys [Shift] [▲] (shift cursor up), the non-locked 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 both keys down, the text continues to roll upward until you release the keys or until the final line of data in display memory appears in the top row of the screen. In the later case, pressing or continuing to hold down the keys has no further effect. The "roll up" function is illustrated in figure 4-1 A.

In the configuration and user softkey definition menus, roll up 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:

$E_C S$

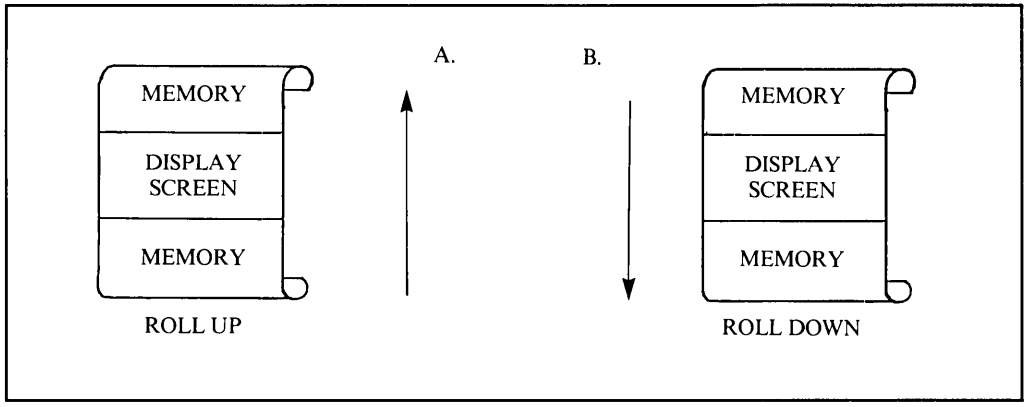


Figure 4-1. The "Roll" Data Functions

Roll Text Down ([Shift] [▼])

Each time you press keys [Shift] [▼] (shift cursor down), 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 keys has no further effect. The "roll down" function is illustrated in figure 4-1 B.

In the configuration and user key definition menus, roll down 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:

EcT

Next Page/Previous Page ([Next] / [Prev])

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] 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 so that the final line in display memory appears in the top line of the screen.

In the configuration and user-key definition menus, this key is disabled. In memory lock mode, the locked text remains at the top of the screen, and only the unlocked lines are moved.

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

EcU

Pressing the [Prev] 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 so the first line in display memory appears in the top line of the screen.

In the configuration and user-key definition menus, this key is disabled. In memory lock mode, the locked text remains at the top of the screen, and only the unlocked lines are moved.

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

$E_C V$

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

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 typically up to 96 lines (0-95). 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

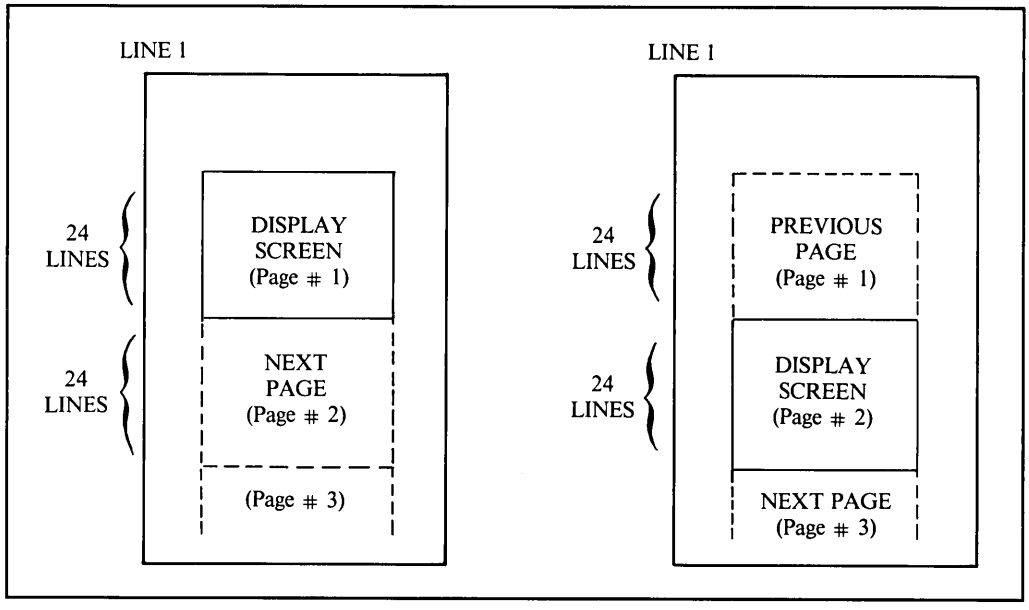


Figure 4-2. Previous Page and Next Page Concepts

Row addressing. Figure 4-3 illustrates the way the three types of addressing affect line (or row) 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 absolute and screen relative column addressing. Figure 4-4 illustrates the difference between absolute and relative addressing. The cursor is shown in column 5.

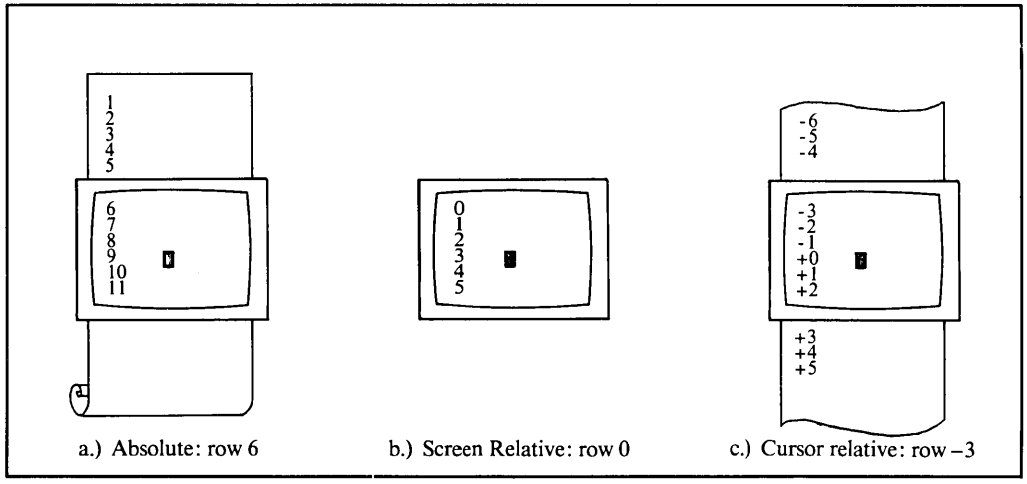


Figure 4-3.Row Addressing

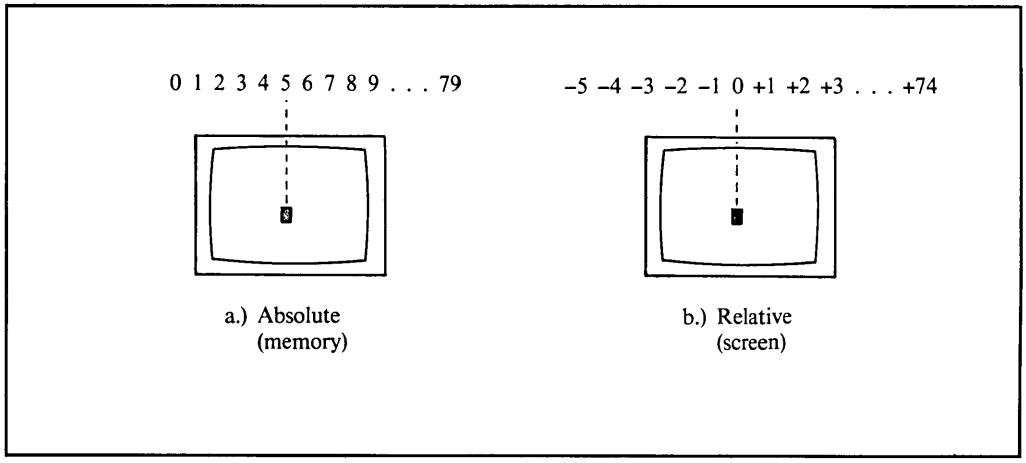


Figure 4-4. Column Addressing

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 (if it is, a roll is performed). For example, in figure 4-3c a cursor relative row address of −10 would cause the cursor to be positioned at the top of the curent screen (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 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 EC a

Example: The cursor is at column 20, row 40.

computer: EC a

terminal: EC &a 020c 040R

Relative Sensing EC

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

computer: EC `

terminal: EC &a 020c 005Y

NOTE

Columns are numbered 1 to 80 on the screen but 0 to 79 in escape sequences. Similarly, rows are numbered 1 to 24 on the screen but 0 to 23 in escape sequences.

Cursor Positioning

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

Screen Relative Addressing

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

E_C&a <column number> c <row number> Y
or E_C&a <row number> y <column number> C
or E_C&a <column number> C
or E_C&a <row number> Y

where:

<column number> is a decimal number specifying the screen column (0-79) to which you wish to move the cursor. Zero specifies the leftmost column.

<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.

Note that if a screen column > 79 is specified, the cursor stops on column 79 and if a screen row > 23 is specified the cursor stops on row 23.

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:

E_C&a 6y 19C

Memory Absolute Addressing

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

E_C&a <column number> c <row number> R
or E_C&a <row number> r <column number> C
E_C&a <column number> C
or E_C&a <row number> R

where:

<column number> is a decimal number specifying the relative column (0 to 79) 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 row coordinate (within display memory) of the character at which you want the cursor positioned. Zero specifies the first (top) row in display memory.

Note that if a screen column > 79 is specified the cursor stops in column 79 and if a row in excess of the screen memory is specified the cursor stops in the last line of the memory.

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 necessary) so that it is positioned at the character residing in the 60th column of the 27th row in display memory:

E_C&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 the row coordinate is ignored when memory lock mode is on). To move the cursor to another character position using cursor relating addressing, use any of the following escape sequences:

E_C&a +/– <column number> c +/– <row number> R
or E_C&a +/– <row number> r +/– <column number> C
or E_C&a +/– <column number> C
or E_C&a +/– <row number> R

where:

<column number> is a decimal number specifying the relative column (o to 79) 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 downward 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:

E_C&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.

E_C&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.

E_C&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.

E_C&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 ([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 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:

$E_C L$

Delete Line ([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:

$E_C M$

Insert Character ([Insert char])

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 position through the right margin move one column to the right. Characters that are forced over the right margin are lost; and 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.

The 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 automaticallly moves to the first character position of the next subsequent unprotected field when the first character is inserted.

In the user-keys definition menu, insert character acts the same as in format mode: insert character is disabled in a configuration menu.

From the keyboard, you enable and disable the insert character editing function using the **[Insert char]** key. When enabled, "Ins Char" is 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: $E_C Q$
DISABLE: $E_C R$

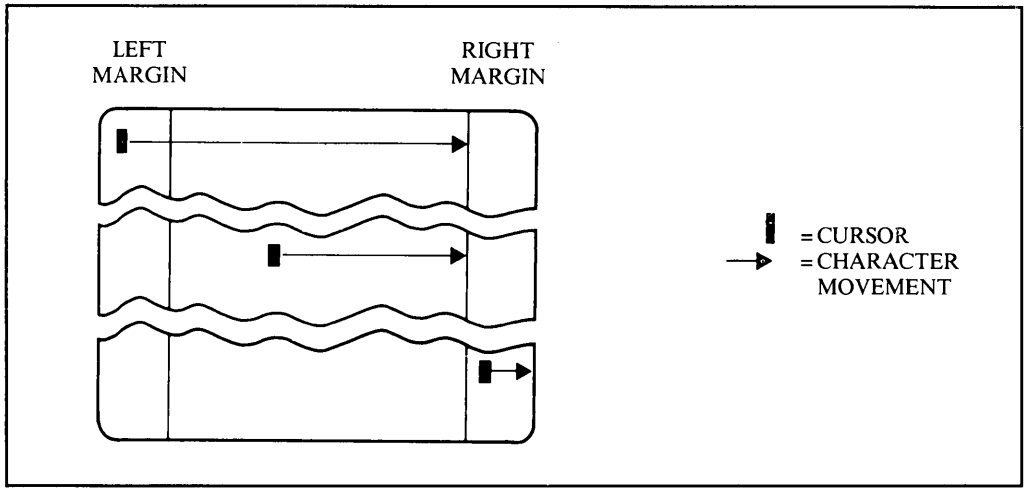


Figure 4-5. Character Insert with Margins

Delete Character ([Delete char])

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.

The 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.

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.

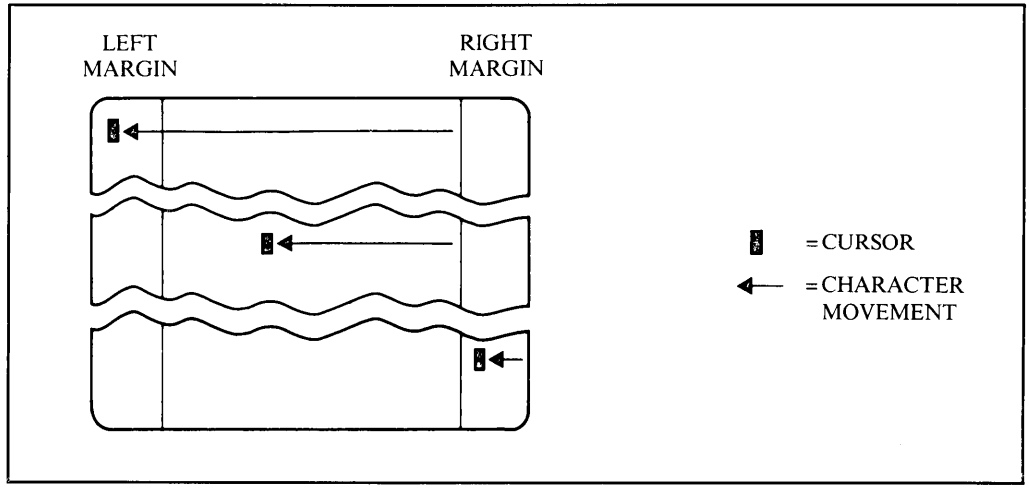


Figure 4-6. Character Delete with Margins

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 keys definition menu, delete character acts the same as in format mode: delete character is disabled in a configuration menu.

From the keyboard, each time you press the **[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:

$E_C P$

Clear Display ([Clear display])

When format mode is off, pressing the **[Clear display]** 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 display]** 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-keys definition and configuration menus.

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

$E_C J$

Clear Line ([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-keys 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:

$E_C K$

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 C_R control code is received, the cursor always moves to the left margin in the current line regardless of whether or not auto line feed mode is enabled.

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

[User System] then **[f2]** margins/ tabs/col

This changes the function key labels to the following:

[f1]	[f2]	[f3]	[f4]	[f5]	[f6]	[f7]	[f8]
START COLUMN	SET TAB	CLEAR TAB	CLR ALL TABS	LEFT MARGIN	RIGHT MARGIN	CLR ALL MARGINS	

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: EC 4
SET RIGHT MARGIN: EC 5
CLEAR ALL MARGINS: EC 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/tabs/col** set of function keys. To get to that set, use the following keystroke sequence:

[User System] then [f2] margins/ tabs/col

This changes the function key labels to the following:

[f1]	[f2]	[f3]	[f4]	[f5]	[f6]	[f7]	[f8]
START COLUMN	SET TAB	CLEAR TAB	CLR ALL TABS	LEFT MARGIN	RIGHT MARGIN	CLR ALL MARGINS	

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. Also, tab stops are ignored in format mode.

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

SET TAB: E_C1
CLEAR TAB: E_C2
CLEAR ALL TABS: E_C3

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.

FORWARD TAB ([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 H_T control code (decimal 9; Control "I") or the following escape sequence:

E_CI

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 ([Tab |◄])

From the keyboard you can move the cursor backward to the previous tab stop using the [Shift] and [Tab |◄] keys.

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:

E_Ci

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

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

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

SCREEN BLANKING

From a program executing in a host computer you can turn off and on the alphanumeric display video, excluding the function key labels and Status Line. This feature may be used to turn off the display video while a form is being drawn into alphanumeric memory. After the form is completed, the video may be turned on. The escape sequences follow:

Turn off display video: E_C&w13F
Turn on display video: E_C&w12F

This escape sequence does not effect menus.

DISPLAY ENHANCEMENTS

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

- Security Video – characters are not displayed (this enhancement is used in conjunction with fields in which passwords or similar security-sensitive data must be entered through the keyboard).
- Inverse Video – dark characters are displayed against a light 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.

NOTE

Using display enhancements reduces the amount of memory available for displaying characters on the screen. Consequently, when the display memory is full activating a display enhancement will cause existing data in memory to be "lost".

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:

E_C&d <enhancement code>

where enhancement code is one of the uppercase letters A through O and S specifying the desired enhancement(s) or an @ to specify end of enhancement as shown in table 4-1.

When a video display enhancement is enabled, it affects all the characters from the current cursor position to the end of the line or to an existing change in enhancement (or character set) in the line, whichever occurs first.

Table 4-1. Display 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" (E_C&d@) or the escape sequence for another video enhancement, will end the previous enhancement.

The terminal also offers a Security field. To enable the security field, the character "s" or "S" is used as an enhancement code.

E_C&dS: starts security video, and stops the previous video attribute.

E_C&ds <A-O> enables both security field and the A to O attribute.

E_C&ds@ is converted to E_C&dS, and E_C&d@S is not executed.

In format mode, E_C&dS is not allowed (the bell will ring if it is typed on the keyboard). However, if E_C&dS is received before entering format mode it remains in operation (i.e. Security fields cannot be created or removed in format mode).

In format mode, E_C&ds <A-O> is converted to E_C&d <A-O>.

Example 1: Define columns 10 through 14 of line 5 to be inverse video and blinking.

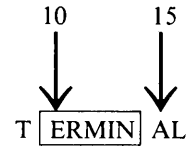
Step 1. Position the cursor at column 10 in line 5.

Step 2. Enter E_C&dC.

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

Step 4. Enter E_C&d@ (this ends the enhancements). The field should be light colored.

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



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

Example 2: Enter Underline, Half-bright Inverse, Inverse, and End Enhancement escape sequences into the [f1] through [f4] user keys (figure 4-7).

Step 1. Press [CTRL] [Menu] to display the menu.

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

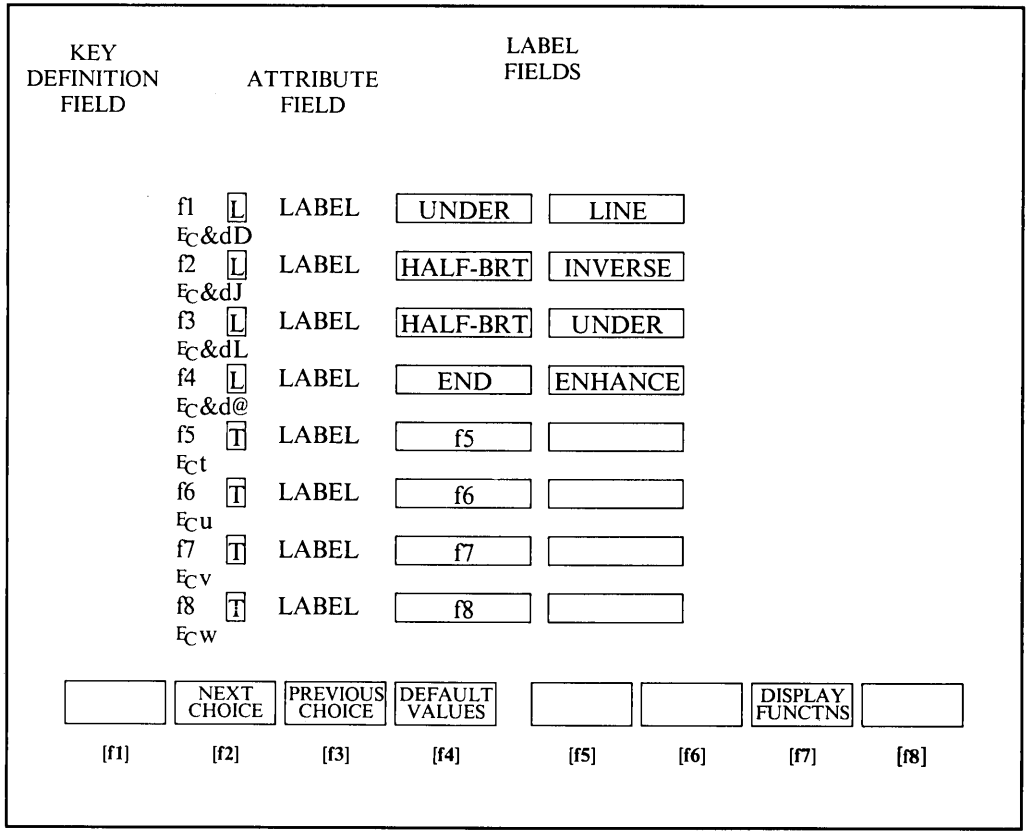


Figure 4-7. Menu After Entering Four Enhancement Escape Sequences

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

Step 4. Press [**Tab ►I**] to position the cursor to the definition field; then press the "**DISPLAY FUNCTIONS**" key. Now, enter the escape sequence for the enhancement; then, turn off display functions. (Display functions is turned on to enter the "$_{EC}$", escape, character).

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

Step 6. When you have finished entering the data into the menu, press [**Shift**] [**User System**] (or [**CTRL**] [**User System**] or [**Menu**] twice) to return to the normal display with the user key labels.

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

DESIGNING AND USING FORMS

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

When format mode is enabled, the cursor automatically moves to the start of the first unprotected field in the form. Henceforth, the terminal operator can only enter data into those portions of the display screen which lie within unprotected fields, the remainder of the screen is protected. When the operator enters a character into the last position of a field, the cursor advances to the start of the next unprotected field. In addition, the [**Tab ►I**] and [**Shift**] [**Tab I◄**] keys can be used to move the cursor to the start of the next or preceding unprotected field. 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: $_{EC}$W
DISABLE: $_{EC}$X

These sequences may be entered directly through the keyboard, executed from 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 optional line drawing set.
2. Define the various unprotected fields within the form.
3. Programmatically read the completed form and store it in the host computer for future use.

DRAWING FORMS

The 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 ROMAN 8 character sets.

When you are designing a form through the keyboard, you use S_O (control-N) and S_I (control-O) control codes to switch back and forth between the Line Drawing and ROMAN 8 character sets.

The correspondence between the alphanumeric characters and the line drawing elements on the USASCII keyboard is shown in figure 4-8 (see Appendix B for national keyboards). Note that as the position of keys generating the same alphanumeric character varies from keyboard to keyboard, so does the position of line elements. Also, national language characters that do not have a corresponding line element generate a space.

You switch from the Base set ROMAN 8 to the Line Drawing set by issuing an S_O code (control-N) and you switch from the Line Drawing set back to the Base set by issuing an S_I code (control-O). Note that the S_O code affects only those characters from the current cursor position through the next S_O or S_I code, or through the next video enhancement, 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 S_O code at the start of the second line.

Figure 4-9 illustrates a sample form and identifies the keystrokes used for the various different types of line segments.

One approach to generating a form structure through the keyboard is to load two of the user keys with the S_O and S_I codes (control-N and control-O, respectively, with displays functions mode enabled), define both as Local keys, define their Labels as **"Line Draw"** and **"Base Set"**, respectively, and enable them by pressing **[Shift] [User System]**, or **[CTRL] [User System]**, or **[Menu]** twice. Then draw the form structure and alphanumeric annotations using the Base set. 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. You may also, if you wish, load some of the more repetitive line definitions into user keys to speed up the drawing of the main body of the form.

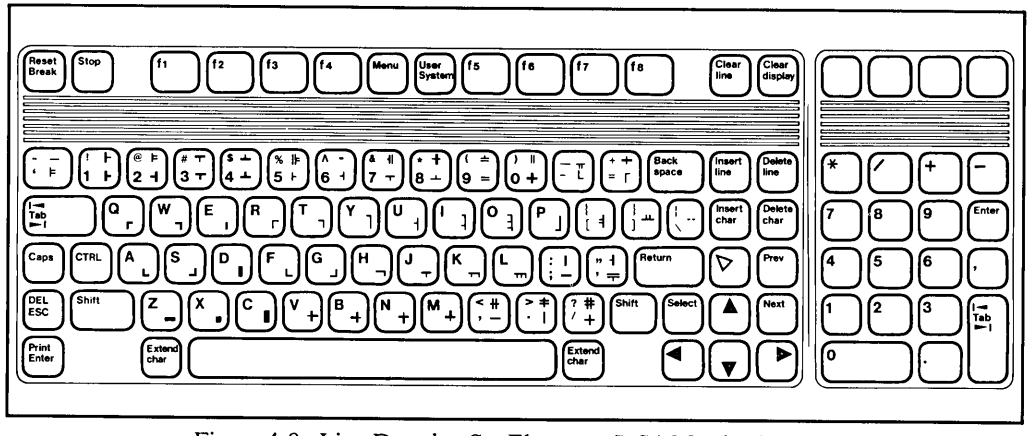
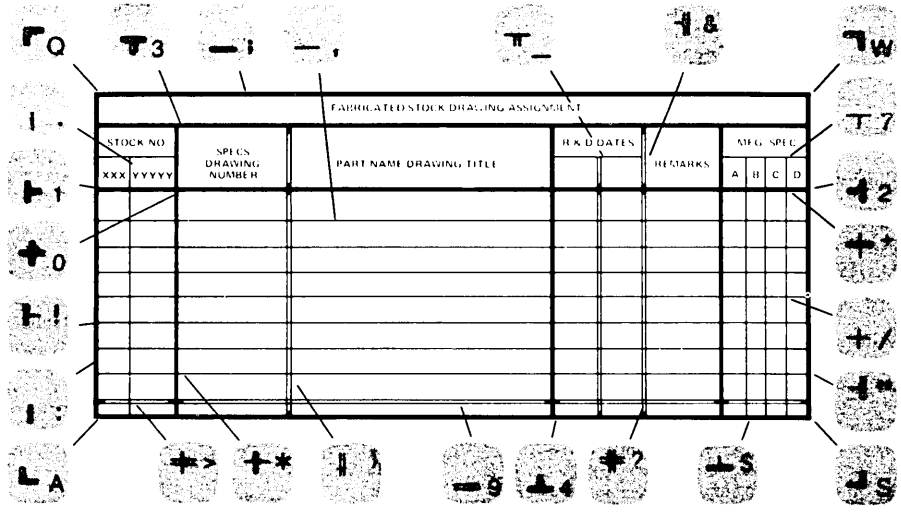


Figure 4-8. Line Drawing Set Elements (USASCII keyboard)

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

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

Note that the Line Drawing set is always the alternative character set (and does not need to be defined). However, the program must go through the same process of shifting out of the base character set (S_O = control-N = decimal 14) to draw the linear portions of the form, and shift back into the base character set (S_I = control-O = decimal 15) to structure the alphanumeric portions of the form.



Note: On Swiss French and Swiss German keyboards, in 7-bit operations "è" must be used instead of ">" to generate the "#" element.

Figure 4-9. Sample Data Entry Form

FORMS MODE (FORMAT MODE)

In forms mode, the terminal prevents you from overwritting or transmitting data in protected fields. Forms mode is normally entered under control of the computer. Forms mode is turned on by sending EcW (the cursor is homed to the beginning of the first unprotected field). Normal operation is turned with EcX (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. Except for the last unprotected field, after reaching the end of an unprotected field, the cursor moves to the beginning of the next unprotected field. For the last unprotected field, the cursor remains at the first character position following the last unprotected field (this is a protected position). Then, if a character is entered from the keyboard, the cursor moves to the first position in the first unprotected field on the screen, and the typed character is entered in this position.

The tab functions can be used to move from one unprotected field to the beginning of the next unprotected field (tab stops are ignored in forms mode). EcI causes the cursor to be positioned at the beginning of the previous unprotected field. Fields are defined as "unprotected" by using Ec[at the start of the field. Ec] or the end of the line is used to end the field.

In figure 4-10, 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.

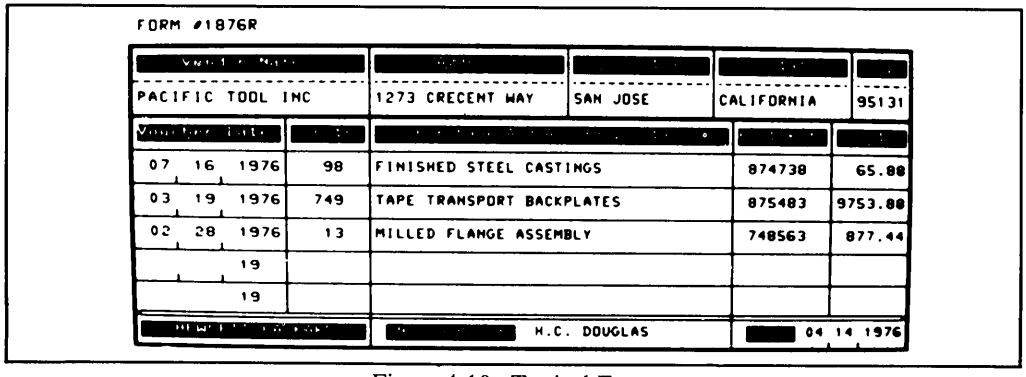


Figure 4-10. Typical Form

NOTE

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

External Printer Control

SECTION V

INTRODUCTION

The standard terminal has one port (Port 1), but may be equipped with a Datacomm Module to provide a second port if required (see the terminal accessories table 1-1).

On terminals equipped with two ports, either of the ports (Port 1 or Port 2) can be used for connecting an external printer. The choice of the printer port is made in the Terminal Configuration Menu (see Section II). With the external printer you may do any of the following:

- Print the line containing the cursor.
- Print all lines from the one containing the cursor through the bottom line on the screen.
- Print all lines from the one containing the cursor through the end of display memory.
- Copy a configuration menu from the screen.
- Enable data logging (to occur either from the top or bottom of display memory, as designated by you when you enable it).
- Enable record mode.
- Perform a line feed (advance the paper one line).
- Perform a form feed (advance the paper one page).

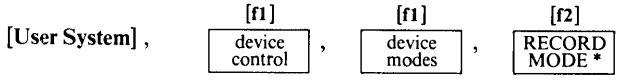
All of the above printer control functions can be initiated either locally by operator keystrokes or remotely by escape sequences sent from the host computer.

Record Mode

Record mode copies data from the datacomm to the selected "to" device(s).

- If in local mode, record mode has no action.
- If in remote mode, the data stream on the datacomm line is sent directly to the selected "to" device (display and/or printer).

Record mode may be initiated from either the keyboard or from an E_C&p escape sequence. To initiate record mode from the keyboard, press



An asterisk will appear in the softkey label to indicate that record mode is enabled. While in record mode, the keyboard is disabled except for the **[Shift]**, **[CTRL]**, **[Reset Break]**, and **"RECORD MODE"** keys. Pressing **[Shift]**, **[Reset Break]**, or **[Shift]**, **[CTRL]**, **[Reset Break]**, or **"RECORD MODE"** function key will terminate record mode.

To initiate record mode from an escape sequence, send:

Turn on record mode: E_C&p <NUM> p 20C
(<NUM> p is optional)

The optional <NUM> parameter defines the character which may be used to turn off record mode. <NUM> is the decimal equivalent of an ASCII character that will turn off record mode if it is the first character in a record mode. The default is "0". If "<NUM>" is omitted, or if "0p" is specified, no character will terminate record mode. Termination can only occur by pressing the **"RECORD MODE"** softkey, pressing **[Shift]** **[Reset Break]** (a soft reset), or pressing **[CTRL]** **[Shift]** **[Reset Break]** (a hard reset).

The termination character is valid only for the current activation of record mode (i.e., when record mode is terminated, the termination character returns to "0", the default character).

The terminal returns an "S", "F", or "U" to the host computer if the escape sequence is received from datacomm. An "S" means that the terminal executed the escape sequence successfully; an "F" means that the terminal failed to execute the escape sequence; a "U" means that you terminated execution.

When the status character is sent depends upon whether or not a $D_1/D_2/D_1$ handshake is enabled. (Refer to the Terminal Configuration Menu discussion in Section II for handshake types). If both the D_1 and $D_1/D_2/D_1$ handshakes are disabled (no handshake), the character is sent immediately after the escape sequence is received from the host computer. If the handshake is enabled, the character is sent after record mode is turned off and the appropriate number of D_1's are received from the host computer.

A 256-character buffer is used to hold each record prior to sending it to the specified "to" device(s). If the record exceeds 256 characters, the terminal's handshake holds off any further transmission from the host until the buffer's contents is sent to the "to" device(s). Records shorter than 256 characters are terminated by a L_F (linefeed) character. Again, the terminal's handshake holds off any further transmission from the host until the record in the buffer is sent to the "to" device(s).

If record mode is turned off, the contents of a partially filled buffer will be sent to the "to" device(s).

If the record mode termination character is the first character into the buffer, record mode is terminated; the termination character is not sent to the "to" device(s).

DATA LOGGING

The terminal includes a mechanism called "data logging" whereby data can be automatically routed to the external printer (provided it has been set as the "to" device). There are two types of data logging: top and bottom.

Top Logging

When the display memory is full 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 sent to the external device. Thus, while the line is "lost" from display memory, it is maintained in hard copy form. Top logging is disabled if overflow protect is active (see memory lock mode, Section III).

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 sent to the external device. 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 external device can process data (the device throughput rate).

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 label.

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

ENABLE BOTTOM LOGGING: E_C&p 11C
ENABLE TOP LOGGING: E_C&p 12C
DISABLE LOGGING: E_C&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 DEVICE ALPHANUMERIC DATA TRANSFERS

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

[User System], then [f1] device control

This changes the function key labels to the following:

[f1]	[f2]	[f3]	[f4]	[f5]	[f6]	[f7]	[f8]
device modes	TO EXT DEV*	TO DISPLAY*	ADVANCE PAGE	ADVANCE LINE	COPY ALL	COPY PAGE	COPY LINE

To select an external printer as the "to" device, press the **"TO EXT DEV"** function key ([f2]). An asterisk appears in the associated function key label.

Programmatically, you can define the "to" device by using a device control escape sequence (E_C&p):

Select external device: E_C&p 4D

When data is being copied to an external device, keyboard entries are not executed until the transfer is complete (and only the last eight alphanumeric entries are buffered in the terminal's memory — all other keystrokes are lost).

Advance Page ([f4])

When the external printer is selected as a destination device, pressing the **ADVANCE PAGE** key ([f4]) in the **device control** set of system function keys sends an F_F control code to the printer, thus causing the paper to be advanced to the top of the next page.

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

E_C&p 4u 0C

Advance Line ([f5])

When the external printer is selected as a destination device, pressing the **ADVANCE LINE** key ([f5]) in the **device control** set of system function keys sends an C_R L_F control code sequence to the printer, thus causing the paper to be advanced by one line.

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

E_C&p 1p 4u 1C

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.

Copy All ([f6])

When the external 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 1, NOT the left margin). If the cursor is at a line that is beyond the last displayable line, no data is sent to the external printer.

From the keyboard, you copy "all" using the **COPY ALL** ([f6]) key in the **device control** set of system function keys. The transfer can be stopped at the end of the current line by pressing **[Return]**.

From a program executing in a host computer, you copy "all" using one of the following escape sequences:

E_C&pM or E_C&p0M

Copy Page ([f7])

When the external 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 1, NOT the left margin). If the cursor is at a line that is beyond the last displayable line, no data is sent to the external printer.

From the keyboard, you copy a page of data using the COPY PAGE key ([f7]) in the device control set of system function keys. The transfer is stopped at the end of the current line when you press [Return].

From a program operating in a host computer, you copy a page of data using one of the following escape sequences:

E_C&pF or E_C&p0F

Copy Line ([f8])

When the external 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 1, NOT the left margin). If the cursor is on an empty line after the last line of data in display memory, **COPY LINE** should not cause anything to print. If the cursor is in an empty line between two blocks of data, **COPY LINE** causes a C_R L_F to be sent to the printer.

From the keyboard, you copy one line of data using the **COPY LINE** key (**[f8]**) in the **device control** set of system functions keys.

From a program executing in a host computer, you copy one line of data using one of the following escape sequences:

E_C&pB or E_C&p0B

Copy Display Memory ([Print Enter])

When the external printer is selected as a destination ("to") device, you can copy all of display memory to it by using the [Shift] [Print Enter] keys or using an E_C0 sequence. In response to this sequence, the terminal homes the cursor and then copies the entire content of display memory to the printer. Note that when display lock mode is active, cursor home up is below the last locked line.

During the data transfer, block terminators and non-displaying terminators within display memory are ignored.

The E_C0 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 [Print Enter] key (i.e. enter) or [Shift] [Print Enter] keys (print) 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 external printer by pressing the [Print Enter] key by itself or by pressing the [Print Enter] and [Shift] keys together. If the user key definition menu is currently displayed, in local mode you can copy the menu to the external printer using the [Print Enter] key or by pressing the [Print Enter] and [Shift] keys together. In remote mode, you can copy the menu by pressing the [Print Enter] and [Shift] keys together. Using the [Print Enter] key transmits the function key definition escape sequence E_C&f...to the computer (see Section III).

Device Control Completion Codes

After issuing a copy line, copy page, copy all (from line containing cursor to end of memory), advance line, or advance page E_C&p sequence, the remote program determines whether or not the operation was successfully performed by reading a one-character status byte from the terminal. The transer is initiated by a handshake with the terminal. For example, from a BASIC program, an INPUT statement would be used. The terminal responds by sending an "S", "F", or "U". An "S" indicates sucessful 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 C_R (or a $C_R L_F$ if auto line feed mode is enabled) following the completion code. In block page mode, it sends a block terminator character (as defined in the terminal configuration menu described in Section II of this manual).

As an alternative to completion codes, you can send a device status request to the terminal following the print operation (refer to "Device Status", Section VII).

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 VII.

COMPUTER TO DEVICE DATA TRANSFERS

When the external printer is selected as a destination ("to") device, you can initiate a data tansfer from a program executing in a host computer to the printer by using the following device control escape sequence:

E_C&p <character-count> W <record>

where:

<**character-count**> is an integer within the range 1-256 specifying the number of binary bytes in <**record**>. The record is terminated when the specified number of binary bytes have been transmitted. If this parameter is not present, normal (ROMAN 8) transfers are initiated and the record is terminated when the 256th data byte after the "**w**" is transmitted or by the first L_F code, whichever occurs first. If the record is terminated by an L_F, the L_F is also passed to the printer.

<**record**> is the data record to be transmitted.

Example: Send the next 15 binary bytes from the computer to all "to" devices.

E_C&p 15W

This escape sequence is recognized only when received over a datacomm 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 (E_C&p) sequence. In any case, the only destination devices that are recognized by this type of data transfer operation are the display (**3d**) and external printer (**4d**).

If no destination devices are specified within the above escape sequence, the current "to" device assignments are used. If nothing is currently selected as a "to" device, the data record is accepted over the host computer port and then is discarded by the terminal (also an "**F**" is returned as the device control competition code).

Binary transfers (where parity checking is disabled) are of the form E_C&p <**character-count**> W <**record**>; all eight bits received are passed to the destination devices. Other transfers (USASCII, Extended Roman), with parity checking performed as set in datacomm configuration menu, are of the form E_C&pW <**record**>, where L_F or the 256th data character terminates the record.

1. If the escape sequences does NOT include a <**character-count**>, the following applies:

   - If **EnqAck** = **YES** in the active data communications configuration menu, the datacomm firmware strips all **ENQ** characters from the incoming data and responds to each by transmitting an **ACK**.
   - If **EnqAck** = **NO** in the active data communications configuration menu, all incoming characters (including **ENQ** and **ACK** but excluding **NULLS** and **DELS**) are treated as data bytes. **NULLS** and **DELS** are stripped from the incoming data.

2. If the escape sequence includes a <**character-count**>, then the following apply:

   - If **EnqAck** = **YES** in the active data communications configuration menu, the first **ENQ** character received in the following data stream is treated as part of the **ENQ/ACK** handshake (the **ENQ** is stripped and an **ACK** returned). Only after returning **ACK** does the terminal enter binary mode. All subsequent characters (including **ENQ, ACK, NUL** and **DEL**) are treated as 8-bit data bytes, regardless of the datacomm configuration menu. (All characters preceding the **ACK** are processed according to the datacomm configuration menu).
   - If **EnqAck** = **NO** in the active data communications configuration menu, binary mode is active immediately after the "W" character of the escape sequence. All **ENQ, ACK, NULL** and **DEL** characters are treated as data.

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 reading a one-character status byte from the terminal. The transfer is initiated by a handshake with the terminal. For example, from a BASIC program, an INPUT statement would be used. The terminal responds by sending an "S", "F", or "U". An "S" indicates successful completion, an "F" indicates that the operation failed, and a "U" indicates that the terminal operator interrupted the data transfer by pressing [Return]. Note that these completion codes cannot be suppressed by configuration parameters or any other means. They are always transmitted and your programs should include input commands explicitly for accepting them. The keyboard is disabled ("locked") until the status is sent.

Note that in either character or block line mode, the terminal sends a CR (or a CR LF if auto line feed mode is enabled) following the completion code; in block page mode, it sends a block terminator character (as defined in the terminal configuration menu).

As an alternative to completion codes, you can send a device status request to the terminal following the print operation (refer to "Device Status", Section VII).

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 VII of this manual.

CONFIGURING THE EXTERNAL PRINTER

Port 1 is always a serial RS232C/HP422 port. On terminals equipped with two ports, Port 2 can be serial or parallel (Centronics). Configuring an external printer consists of assigning it to the port and specifying parameters in the external device configuration menu.

Printer Cabling

The available printer cables are listed in table 5-1.

The male connector on the end of the cable attaches to the external device port on the rear of the terminal, and the other end attaches to the printer as shown in figure 5-1.

You can use cables other than those listed in table 5-1, as long as they are equipped with the correct connector and their pinouts are compatible with the HP cables.

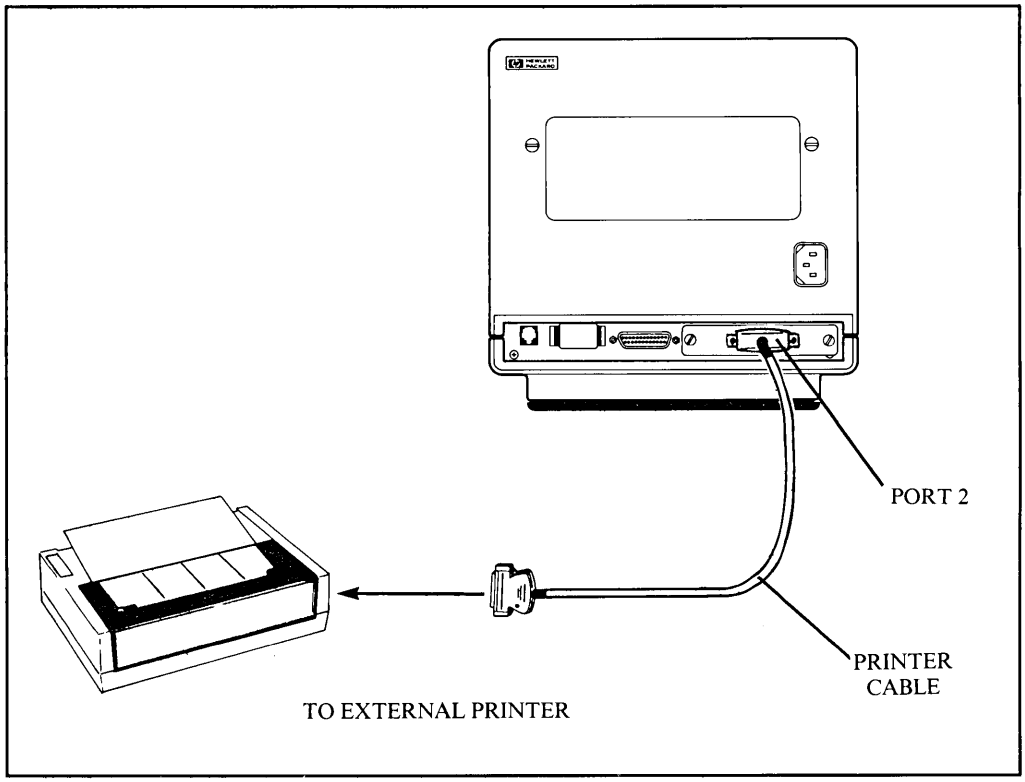


Figure 5-1. External Device Port Cabling

Table 5-1. External Device Port Data Communications Cables

Cable No.	HP Part No.	Description
		Serial Printer Cable
40242G	–	RS232 PRINTER CABLE MALE 25 PIN TO MALE 25 PIN Male RS232C 25-pin connector for interfacing the terminal to RS232C compatible printers such as the HP 263X, HP 267X, HP 293X. Length: 5 meters (16 feet)
		Parallel Printer Cable
40242D	–	PRINTER CABLE AMPHENOL MALE 36 PIN TO MALE 36 PIN Length: 2 meters (6.6 feet)

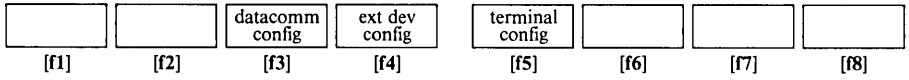
Filling In The Configuration Menu

Now that you have made the physical connections between the terminal and the external printer, you are ready to configure the terminal's external device port.

To configure the port, first use the following keystroke sequence:

[User System], config keys **[f8]**

This changes the function key labels to the following:



The **ext dev config** function key, when pressed, causes the external device configuration menu to appear on the screen and redefines the function keys to a set of functions that will assist you in manipulating the various parameters within the menu (see figure 5-2). The configuration menu displays the currently stored menu parameters from non-volatile memory.

NOTE

Whenever a configuration menu is on the screen, incoming data, including escape sequences, is stored in buffer (up to 256 bytes) and not processed and no data is transmitted, until the menu is exited. (Refer to Section VI for pacing considerations).

Whenever a configuration menu is displayed on the screen, the teminal is implicitly in format mode. The menu contains a set of unprotected fields that you access using the [Tab◄►] key. For most of the fields (the ones containing the underlined video enhancement) you select the desired parameters using the **"NEXT CHOICE"** ([f2]) and **"PREVIOUS CHOICE"** ([f3]) function keys.

The meanings of the various fields are described in table 5-2.

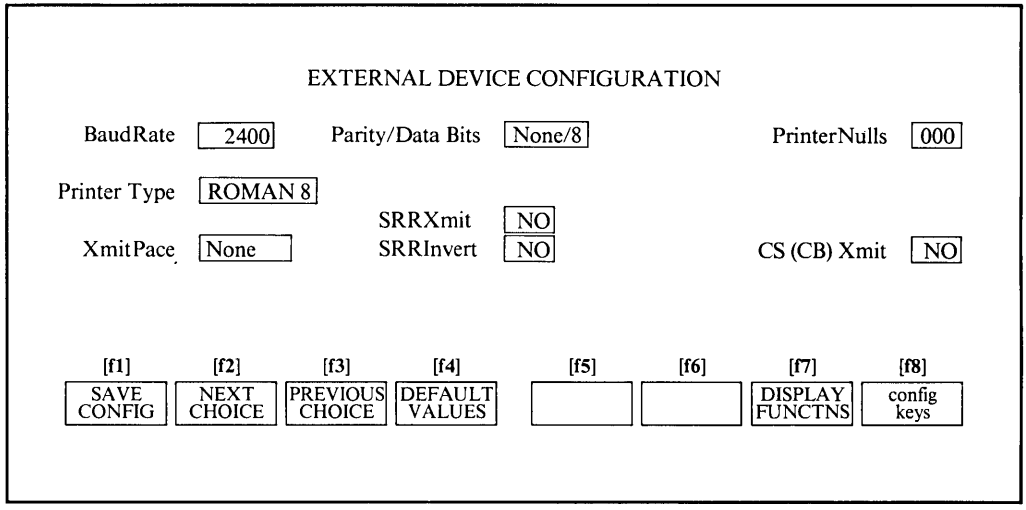


Figure 5-2. External Device Configuration Menu

NOTE

When using record mode or pass through mode (EC&pW) to transfer data directly from the datacomm to the printer, both the external device and the datacomm should use identical data bits, also the same method of accessing extended Roman characters as specified in table B-5 should be used. If not (or not possible), unexpected results may occur.

When the external printer is a parallel printer (or current loop), the following configuration fields should be set to NO: SRRXmit and CS(CB)Xmit.

The HP 2392A Service Manual (part number 02390-90003) lists the configurations recommended for different types of printer.

Table 5-2. External Device Configuration Menu Fields

BaudRate	This field specifies at what speed you want the data transmission to take place (in bits per second). Values: **110** **600** **2400** **134.5** **1200** **4800** **150** **1800** **9600** **300** Default: **2400**
Parity/Data Bits	This field specifies what type of parity generation and checking you wish used with each data character. (Note that parity must be set to None for binary transfers) and the number of bits per character transmitted or received. Values: **None/8** (no parity bit) **0's/7** (parity bit always zero) **ODD/7** (odd parity) **1's/7** (parity bit always one) **EVEN/7** (even parity) Default: **None/8**
PrinterNulls	This field specifies the number of null codes (0-255) to be transmitted to an external printer after each ASCII control code. Value: **0-255** Default: **0**
PrinterType	This field specifies which type of language control should be used to send national characters to the external printer. There are two types of printers: those which support the Roman Extension set as a secondary set, accessed either by the SO control code or by the 8th bit (extended Roman) and those which allow the ROMAN 8 set to be accessed with full 8-bit control (the ROMAN 8 is the base set): ROMAN 8. Value: **EXT ROMAN, ROMAN 8** Default: **ROMAN 8** See Appendix B for more details.

Continued . . .

Table 5-2. External Device Configuration Menu Fields (Continued)

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 (<**DC3**>), it stops transmitting data. When the terminal subsequently receives an XON code (<**DC1**>), it resumes transmitting data. If this field is set to "**NONE**", the terminal does NOT recognize the <**DC1**> and <**DC3**> codes as XON and XOFF. For other forms of transmit pacing, refer to the descriptions of the SRRXmit and CS(CB)Xmit fields below. Values: **NONE** **XonXoff** Default: **NONE**
SRRXmit	This field specifies whether or not an "on" state (+ 12 V) on the RS232C Secondary Receiver Ready (SRR) or Secondary Carrier detect (SCD) control line is a required condition for transmitting data. This mechanism is primarily used in conjunction with printers which must be able to control the transmission of data from other devices. The SRR/SCF control line is connected to RS232C pin number 12. This field should be set to NO when using a parallel printer. Values: **YES** **NO** Default: **NO**
SRRInvert	This field applies only when the SRRXmit field is set to "**YES**". When both the **SRRXmit** and **SRRInvert** are set to "**YES**", the true state of the RS232C Secondary Receiver Ready (SRR) or Secondary Carrier Detect (SCF) control line is inverted from + 12 V to – 12 V. Values: **YES** **NO** Default: **NO**
CS(CB)Xmit	This field specifies whether or not an "on" state (+ 12 V) on the RS232C Clear to Send (CS/CB) control line is a required condition for transmitting data. This field should be set to NO when using a parallel printer. Values: **YES** **NO** Default: **NO**

Data Communications

SECTION VI

INTRODUCTION

The standard terminal has one port (Port 1), a combined RS232C/HP422 port. The terminal can be optionally equipped with one of several interfacing modules (see table 1-1) to provide a second port. Either Port 1 or Port 2 may be used to connect the terminal to the host computer (Port 2 cannot be used when equipped with a parallel port); the port used for the computer connection being selected in the terminal configuration menu (see Section II).

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.

Communications Link:	The means by which a terminal is connected to a host computer. This always includes some type of communications line (a coaxial cable, the public telephone network, or a leased telephone line), and it may also include a pair of modems (one at each end of the line).
Point-to-Point:	A data communications configuration in which a single terminal is connected to a host computer over a communications link. The terminal is designed for use with a point-to-point communications link.
Asynchronous:	A mode of transmission in which each data character is framed by a "start bit" and one or more "stop bits". The interval between successive data characters is random. The terminal is designed for use with an asynchronous communications link.
Full Duplex:	A communications link in which data can be transmitted in both directions simultaneously. The terminal is designed for full duplex data communications.
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 on the keyboard are merely stored in display memory. When a block transfer is subsequently initiated (by the host computer or by pressing the [Enter] key), a group of data characters is sent from the terminal to the computer as a block.

A point-to-point configuration is the standard form of data communications within the industry (it is sometimes referred to as a "Teletype-compatible" communications link). Point-to-point is supported by most computers. At any given time, it accommodates only one terminal per communications 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 RS232C (to which the terminal conforms) limits cable lengths to a maximum of 15 meters (50 feet).

HP422 is limited to a maximum of 1200 meters (4000 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. Please note that a hardwired connection for your terminal is always full duplex (the terminal does not support half-duplex connections).

Table 6-1. Data Communications Cables and Test Connectors

Cable No.	HP Part No.	Description
40242A	-	COMBINED RS232C and HP422 ADAPTER CABLE MALE 25 PIN TO FEMALE 50 PIN CABLE Provides for 50-pin cable connections Length: 1 meter (3 feet)
40242C	-	RS232C CABLE MALE 25 PIN TO FEMALE 25 PIN CABLE Length: 5 meters (16 feet)
40242M	-	US/EUROPEAN MODEM CABLE MALE 25 PIN TO MALE 25 PIN CABLE Male RS232C connector for interfacing the terminal to European telephone system via Bell 103 or 202C type European Modems. Length: 5 meters (16 feet)
40242P	-	HP DIRECT CONNECT TYPE 422 CABLE MALE 25 PIN TO MALE 5 PIN CABLE Length: 5 meters (16 feet)
40242X	-	HP DIRECT CONNECT TYPE 232 CABLE MALE 25 PIN TO MALE 3 PIN CABLE Used to interface the terminal to an HP 3000 Series 44 or 64. Length: 5 meters (16 feet)
40242Y	-	EMP PROTECT MALE 25 PIN TO MALE 25 PIN CABLE Male RS232C 25-pin connector for interfacing the terminal to an HP 1000, or 3000 Multiplexer. Provides protection from lightning-induced transients. For use in hardwired configurations only. Length: 5 meters (16 feet)
40242Z	-	RS232C MODEM BYPASS CABLE MALE 25 PIN TO FEMALE 25 PIN RS232C direct connect cable. Length: 5 meters (16 feet)
13222C	13222-60003	TERMINAL TO RS232 CABLE (used with 40242A cable) MALE 50 PIN TO FEMALE 25 PIN CABLE Female RS232C 25-pin conenctor. Length: 2 meters (6.6 feet)

Continued . . .

Table 6-1. Data Communications Cables and Test Connectors (Continued)

Cable No.	HP Part No.	Description
13222M	13222-60002	EUROPEAN MODEM CABLE (used with 40242A cable) MALE 50 PIN TO MALE 25 PIN CABLE Male RS232C 25-pin connector for interfacing the terminal to the European telephone system via Bell 103 or 202C type European modems. Length: 5 meters (16 feet)
13222N	13222-60001	U.S. MODEM CABLE (used with 40242A cable) MALE 50 PIN TO MALE 25 PIN CABLE Male RS232C 25-pin connector for interfacing the terminal to an HP 1000, 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 feet)
13222P	-	HP DIRECT CONNECT TYPE 422 CABLE (used with 40242A cable) MALE 50 PIN TO MALE 5 PIN CABLE Length: 5 meters (16 feet)
13222W	13222-60007	RS232C CABLE (used with 40242A cable) MALE 50 PIN TO FEMALE 25 PIN CABLE Female RS232C 25-pin connector for interfacing the terminal to an HP 3000 Computer System. Length: 5 meters (16 feet)
13222X	-	HP DIRECT CONNECT CABLE TYPE 232 CABLE (used with 40242A cable) MALE 50 PIN TO MALE 3 PIN CABLE Length: 5 meters (16 feet)
13222Y	13222-60005	EMP PROTECT (used with 40242A cable) MALE 50 PIN TO MALE 25 PIN CABLE Male RS232C 25-pin connector for interfacing the terminal to an HP 1000, or 3000 Multiplexer. Provides protection from lightning-induced transients. For use in hardwired configurations only. Length: 5 meters (16 feet)
13222U	5061-2403	MODEM BYPASS CABLE FEMALE 25 PIN TO FEMALE 25 PIN CABLE Crosses the signals so that two terminals can communicate with one another. Length: 1.5 meters (5 feet)

Continued...

Table 6-1. Data Communications Cables and Test Connectors (Continued)

Cable No.	HP Part No.	Description
-	02620-60056	This is a male 50-pin test hood for use with 40242A cable.
-	02620-60062	This is a male RS232C test hood for use on port # 1.
-	02645-60004	This is a female RS232C test hood for use on an HP 13222M/N/Y or 40242M/Y datacomm cable.
-	5061-3248	This is a male HP422 test hood for use on port # 1.
-	5061-6703	This is a female (3 pin) RS232C test hood for use on an HP 13222X or 40242X datacomm cable.
-	5061-6704	This is a female (5 pin) HP422 test hood for use on an HP 13222P or 40242P datacomm cable.

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. Contact your local HP office for a list of supported modems.

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.

Whenever the modem line (Data Set Ready) is active, and when the Asterisk field in the Datacomm Configuration Menu is set to "DM" (see table 6-3), an asterisk appears below the space between the fourth and fifth screen labels in the Status Line 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 reflects the state of the Data Set Ready (DSR) input line to the terminal. When a modem is connected, the DSR line is high (active) and the modem indicator (asterisk) is on. When the DSR line is low, this signals a modem disconnect and the asterisk disappears from the screen.

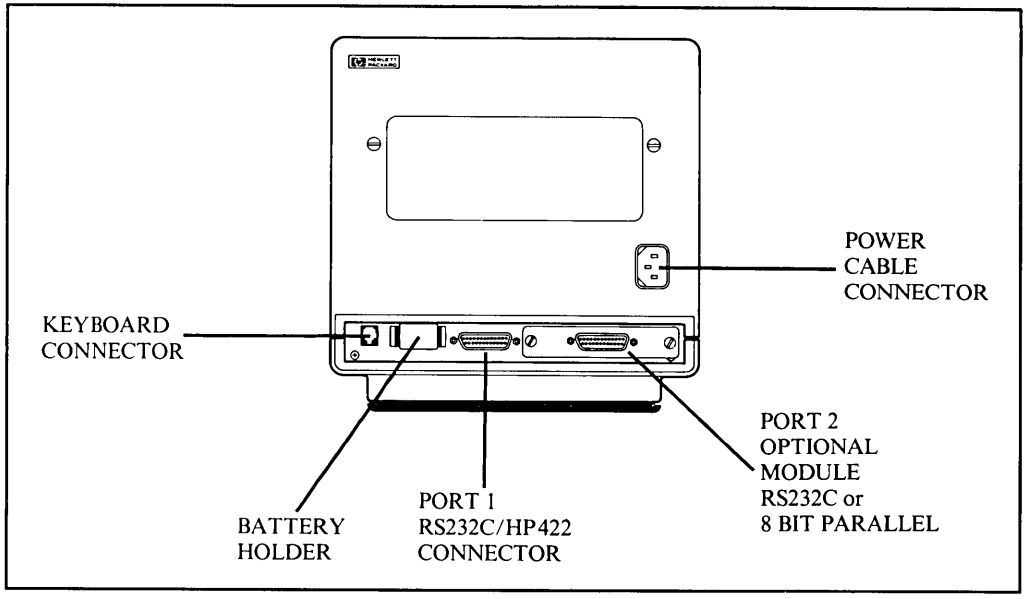


Figure 6-1. Terminal Rear View

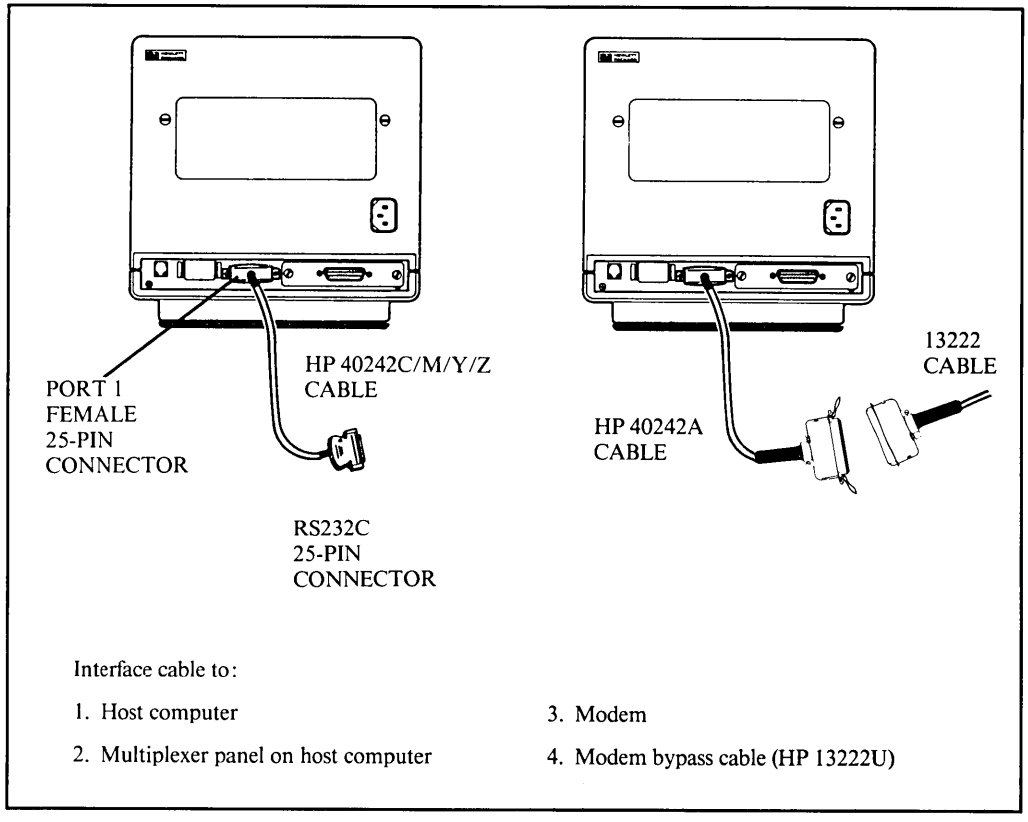


Figure 6-2. Terminal Cabling (HP 40242 and 13222 Cables)

INSTALLING A POINT-TO-POINT CONFIGURATION

The terminal's datacomm port may be connected to a computer via a 25-pin, female RS232C/HP422 compatible connector provided on the back of the terminal (see figure 6-1).

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:

[User System], [f8] config keys, [f3] datacomm config

When you press the **datacomm config** ([f3]) function key, the datacomm configuration menu currently stored in non-volatile memory appears on the screen and the function key labels change to the following:

[f1]	[f2]	[f3]	[f4]	[f5]	[f6]	[f7]	[f8]
SAVE CONFIG	NEXT CHOICE	PREVIOUS CHOICE	DEFAULT VALUES			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 6-3).

The menu contains a set of unprotected fields that you access using the [Tab◄►] key. You select the desired parameters in these fields using the **NEXT CHOICE** ([f2]) and **PREVIOUS CHOICE** ([f3]) function keys.

The meanings of the various fields in datacomm menu are described in table 6-2.

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.

NOTE

Whenever a configuration menu is on the screen, incoming data, including escape sequences, is stored in buffer (up to 256 bytes) and not processed and no data is transmitted, until the menu is exited.

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:

ext dev config label is blank when Port 2 is not equipped with a datacomm module

[f1]	[f2]	[f3]	[f4]	[f5]	[f6]	[f7]	[f8]
		datacomm config	ext dev config	terminal config			

DATACOMM CONFIGURATION

BaudRate [2400] Parity/DataBits [None/8] EnqAck [YES]

Asterisk [OFF] Chk Parity [NO] SR (CH) [LO]

RecvPace [None] CS (CB) Xmit [NO]

XmitPace [None]

Figure 6-3. Datacomm Configuration Menu

Table 6-2. Datacomm Configuration Menu Fields

Baudrate	This fields specifies at what speed you want the data transmission to take place. Values: **110** **600** **4800** **134.5** **1200** **9600** **150** **1800** **19200** **300** **2400** 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. Default: **2400**

Continued . . .

Table 6-2. Datacomm Configuration Menu Fields (Continued)

Parity/Data Bits	This field specifies what type of parity generation and checking you wish used with each data character and the number of bits per received or transmitted character. Note that when parity is set to NONE the number of Data Bits is always 8. Values: **None/8** (no parity bit) **0's/7** (parity bit always zero) **ODD/7** (odd parity) **1's/7** (parity bit always one) **EVEN/7** (even parity) Default: **None/8**
EnqAck	This field enables or disables the use of the Hewlett-Packard ENQ-ACK handshake. This type of handshaking is described under "Pacing Mechanisms" in the "Point-to-Point Programming Information" portion of this section. Values: **YES** (enable) (default) **NO** (disable)
Asterisk	This field specifies whether the transmit (modem) indicator in status line should be enabled or disabled and, if enabled, which RS232C control line it should reflect. The value **"OFF"** disables the TRANSMIT indicator altogether. The value **"CS"** specifies that the TRANSMIT indicator should reflect the state of the RS232C Clear to Send (CS) control line (asterisk = HI; no asterisk = LO). The value **"DM"** specifies that the TRANSMIT indicator should reflect the state of the RS232C Data Mode (DM) or Data Set Ready (CC) control line (asterisk = HI; no asterisk = LO). Values: **OFF** (default) **CS** **DM**
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. Note that the specified parity bit is still transmitted. Values: **YES** (enable) **NO** (disable) (default)
SR(CH)	This field specifies the desired state of the RS232C SR line when the terminal's power is first turned on or when the terminal is reset. The SR line, RS232C pin number 23, is defined as the Data Signal Rate Selector (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)

Continued . . .

Table 6-2. Datacomm Configuration Menu Fields (Continued)

Recvpace	Receive pacing is a mechanism by which the terminal automatically controls (halts and resumes) the transmission of data from the remote device. There is one way of performing receive pacing: by using the XON and XOFF control codes. If this field is set to **"XonXoff"**, the terminal will automatically perform receive pacing using XON (D_1) and XOFF (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 when XON/XOFF receive pacing is on. The [STOP] key can also be used to stop the flow of data from the input buffer to the screen and then resume it. Values: **NONE** (default) **XON/XOFF**
CS(CB)Xmit	This field specifies whether or not an "on" state (+ 12 V) on the RS232C 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 (D_3), it stops transmitting data. When the terminal subsequently receives an XON code (D_1), it resumes transmitting data. This type of handshaking can be used in conjunction with D1/D2 handshaking. If this field is set to **"NONE"**, the terminal does NOT recognize the 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** (default) **XON/XOFF**

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 space or "zero" line state (+ 12 V) that lasts for 1.0 bit time; it is affixed to the begining of a serial character bit stream (which may also include a parity bit). A stop bit is a mark or a "one" line state (– 12 V) that lasts for 1.0 bit time; it is appended to the end of each serial character by 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 is automatically set to "NONE" in the datacomm configuration menu.

The terminal offers the following five 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.

See Appendix B for ROMAN 8 transmissions.

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 quarter full (60 characters), 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 occured. 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 destination device.

NOTE

Because NULL and DEL characters are automatically stripped from datacomm, the only DEL characters appearing on the screen from datacomm are the result of datacomm errors.

Local/Remote Mode

The data communications portion of the terminal operates independently whether the terminal is in local or remote mode. If the terminal is switched from remote to local while data is being received from the remote device, the datacomm portion of the terminal continues receiving data (it does NOT halt the transmission). In such a case, the data received while the terminal is in local is discarded by the terminal's maincode firmware.

Full-duplex Operation

In a full-duplex environment, the terminal is capable of transmitting and receiving data simultaneously. The ability to transmit may be inhibited temporarily, but it is never exclusive of the ability to receive. Two physical sets of data lines are required; control lines are needed only when hardware handshaking or a modem is used. Transitions on the control lines have no effect on the actual transmit/receive state of the terminal.

When the terminal is connected to the host computer via a modem, the following primary control lines are required:

Request to Send (RS/CA)

Clear to Send (CS/CB)

Data Terminal Ready (CD/TR)

If the terminal is hardwired directly to an HP 3000 computer system (no modem), only Transmit Data (SD/BA), Receive Data (RD/BB), and Signal Ground (AB) are required.

Pacing Mechanisms

In a full-duplex environment, the terminal can participate in either of the following forms of transmit pacing:

1. Hardware handshake. The host computer can temporarily restrain the terminal from transmitting by lowering the Clear to Send (CB) line. Note that this type of transmit pacing can only be used in a hardwired configuration where the Clear to Send (CB) line exists in the cabling.
2. **XON-XOFF** handshake. The host computer or external printer uses the control codes **XON** (D_1) and **XOFF** (D_3) to start and stop the terminal from transmitting. Note that a single **XON** code cancels any number of preceding **XOFF** codes.

In a full-duplex environment, the terminal can also participate in the XON/XOFF handshake form of receive pacing, in which the terminal uses the 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 code 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)

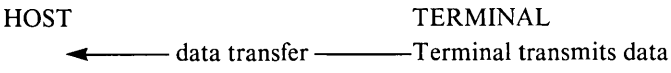
Transfers And Handshaking

The terminal provides three types of data transfer operations:

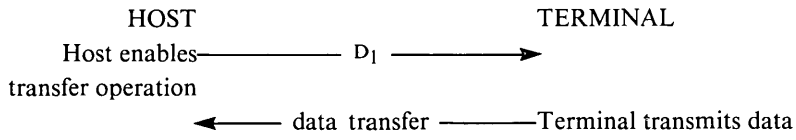
1. Short block transfer. Used in data transfer operations involving:
   a) Terminal primary status (response to Ec^).
   b) Terminal secondary status (response to Ec~).
   c) Device status (response to Ec&p4^).
   d) Cursor sense, relative or absolute (response to Ec` or Eca).
   e) Transmit only user key (T) or [Select] key when the terminal is in block line mode or character mode.
   f) Ecd (initiate data transfer).
   g) Device completion status (S, F or U).
   h) Terminal ID status (terminal response to Ec*s^).
2. Long block transfer. Used in data transfer operations involving:
   a) Transmit only user key (T) or [Select] key when the terminal is in block page mode.
   b) [Enter] key in block mode (line or page).
3. Long character transfer. Used in data transfer operations involving:
   a) [Enter] key in character mode.
   b) [Return] or [Enter] key in modify mode.

In addition, the terminal provides three types of handshaking:

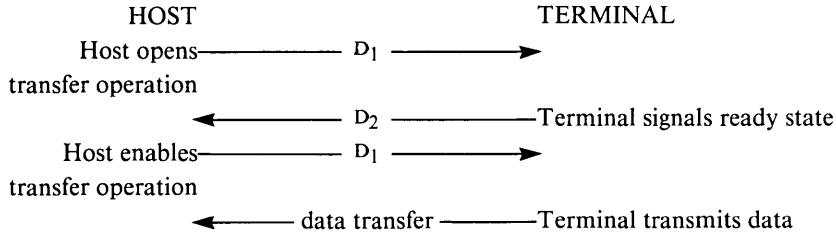
Type 1 – No handshake



Type 2 – D_1 handshake



Type 3 - $D_1/D_2/D_1$ handshake



The handshake used depends on the type of data transfer operation, and the setting of the "InhHndShk (G)" and "InhDC2 (H)" fields in the terminal configuration menu (see Section II). Table 6-3 lists the available handshakes.

Table 6-3. DC1/DC2 Handshake Used In Data Transfer

InhHndShk (G)	InhDC2 (H)	Short Block Transfer	Long Block Transfer	Long Character Transfer
		HANDSHAKE		
NO (0)	NO (0)	Type 2 (DC1)	Type 3 (DC1/DC2/DC1)	Type 1 (no handshake)
NO (0)	YES (1)	Type 2 (DC1)	Type 1 (no handshake)	Type 1 (no handshake)
YES (1)	NO (0)	Type 3 (DC1/DC2/DC1)	Type 3 (DC1/DC2/DC1)	Type 3 (DC1/DC2/DC1)
YES (1)	YES (1)	Type 1 (no handshake)	Type 1 (no handshake)	Type 1 (no handshake)

Data Transfer Priority

If more than one transfer request is pending at the same time, the execution priority is:

1. Primary status.
2. Secondary status.
3. Device status.
4. Cursor sense.
5. Transmit user key (T) or **[Select]** key.
6. **[Enter]** or ECd.
7. Device completion status (S, F or U).
8. ID status.

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):

XmitFnctn (A)	**SPOW (B)**	**InhEolWrp (C)**	**Line/Page (D)**
InhHndShk (G)	**InhDC2 (H)**	**EscXfer (N)**	

3. Secondary Terminal Status. This request returns seven bytes that report the status of the memory lock, buffer memory, and I/O firmware.
4. Device Status. This request returns three bytes that report the status of the external 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: C_R or C_R L_F
Block Line Mode: C_R or C_R L_F
Block Page Mode: < **Blkterminator** >

In either character mode or block line mode, the C_R L_F is used if auto line feed mode is enabled. In block page mode, the block terminator is < **Blk Terminator** > (R_S).

The type of handshaking used is determined by the setting of the **InhHndShk** and **Inh DC2** fields of the configuration menu as follows:

1. **InhHndShk (G) = YES**
   Inh DC2 (H) = YES
   } **No handshake**
2. **InhHndShk (G) = NO**
   Inh DC2 (H) = YES or NO
   } **D_1 handshake**
3. **InhHndShk (G) = YES**
   Inh DC2 (H) = NO
   } **$D_1/D_2/D_1$ handshake**

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 characters shown in table 7-1.

For a terminal ID request, the terminal returns the 5-character defined in the terminal configuration menu.

Terminal ID Status

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

E_C*s^

The terminal responds by sending back the following five-character string defined in the **Terminal Id** field in the terminal configuration menu (see Section II). The default value is:

2392A

Table 7-1. Status Characters

CHARACTER	BINARY EQUIVALENT
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:

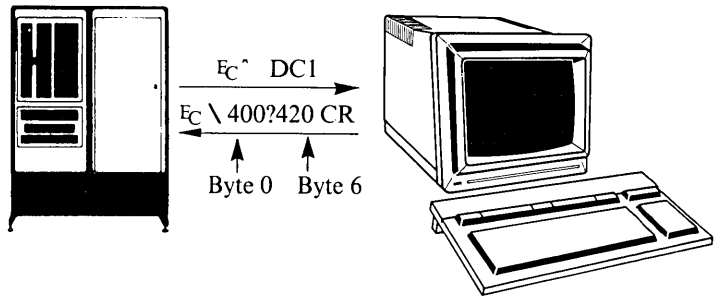
E_C^

The terminal responds with an E_C\, 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 D_1 handshake is being used and that the appropriate terminator is a C_R

SECONDARY TERMINAL STATUS. You request the second set of terminal status bytes (bytes 7-13) by issuing the following escape sequence:

E_C~

The terminal responds with an E_C ! , 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 D_1 handshake is being used and that the appropriate terminator is a C_R



BYTE	ASCII	BINARY	STATUS
0	4	0011 0100	4K bytes of display memory (2 pages of 24 lines)
1	0	0011 0000	Function key transmission disabled Space overwrite latch disabled Inhibit end of line wraparound disabled Strapped for line
2	0	0011 0000	Disable handshake Inhibit DC2 disabled
3	?	0011 1111	Upper case only Block mode Auto line feed Terminal sends secondary status
4	4	0011 0100	ENTER key has been hit
5	2	0011 0010	No data comm errors Last Self-test ok
6	0	0011 0000	No device status pending No device operation status pending

Configuration Straps A-H

Continued . . .

Figure 7-1. Primary Terminal Status Example

PRIMARY STATUS BYTES

BYTE 0 DISPLAY MEMORY SIZE

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

- 1K bytes
- 2K bytes
- 4K bytes
- 8K bytes

This byte specifies the amount of display memory available in the terminal.

BYTE 1 CONFIGURATION STRAPS A-D

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

Strap D
Page/Line
1 = Page
0 = Line

Strap C
(Inhibit End-of-Line Wraparound)
1 = yes (Enabled)
0 = no (Disabled)

Strap A
(Function Key Transmission)
1 = yes (Transmitted)
0 = no (Not transmitted)

Strap B
(Space Overwrite Latch)
1 = yes (Enabled)
0 = no (Disabled)

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

Continued . . .

Figure 7-1. Primary Terminal Status Example (Continued)

BYTE 2 CONFIGURATION STRAPS E-H

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

Strap H (Inhibit DC2)
1 = yes (Enabled)
0 = no (Disabled)

Strap G (DC2 Handshake)
1 = yes (Enabled)
0 = no (Disabled)

Strap E

Strap F

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

BYTE 3 LATCHING KEYS

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

Terminal sends secondary status

AUTO LF Key
1 = auto LF
0 = no auto LF

CAPS LOCK field in terminal configuration menu
1 upper case only
0 upper and lower case

BLOCK MODE Key
1 block mode
0 character mode

BYTE 4 TRANSFER PENDING FLAGS

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

Secondary Status Pending
1 yes
0 no

ENTER Key Pending
1 yes
0 no

Cursor Sense pending
1 yes
0 no

Function Key Pending
1 yes
0 no

Continued...

Figure 7-1. Primary Terminal Status Example (Continued)

BYTE 5 ERROR FLAGS

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

Data Comm
1 parity or buffer overflow error framing overrun error
0 no error

Self Test
1 no error
0 error

BYTE 6 DEVICE TRANSFER PENDING FLAGS

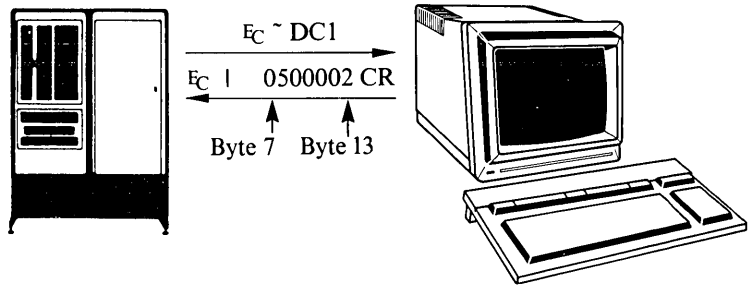
8	7	6	5	4	3	2	1
0	0	1	1	0	0	1/0	1/0

Device Status Pending ($F_C\&p_0^{4\hat{}}$)
1 yes
0 no

Device Operation Status Pending*
1 yes
0 no

* Tracks "S", "F", or "U" completion codes associated with $F_C\&p$ device control sequences.

Figure 7-1. Primary Terminal Status Example (Continued)



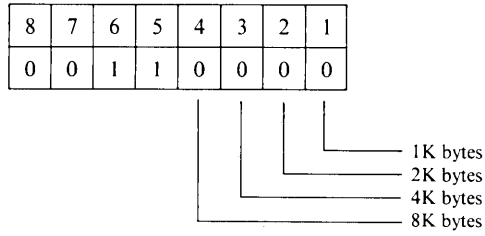
BYTE	ASCII	BINARY	STATUS
7	0	0011 0000	Always 0
8	5	0011 0100	I/O firmware installed Terminal identifies itself
9	0	0011 0000	Always 0
10	0	0011 0000	Do not send escape codes to printer
11	0	0011 0000	Always 0
12	0	0011 0000	Do not check parity
13	2	0011 0010	Memory locked

Continued . . .

Figure 7-2. Secondary Terminal Status Example

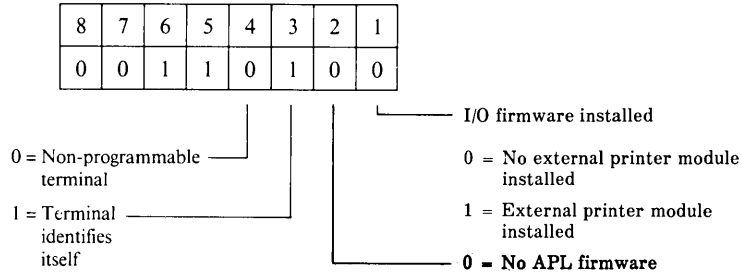
SECONDARY STATUS BYTES

BYTE 7 BUFFER MEMORY
(always zero)



Memory installed in addition to display memory that is available for use as data buffers. Note that the HP 2392A terminal always return a 0 value.

BYTE 8 TERMINAL FIRMWARE CONFIGURATION



Continued . . .

Figure 7-2. Secondary Terminal Status Example (Continued)

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 L
Strap J
Strap K

Straps J-M do not apply to the terminal.

BYTE 10 KEYBOARD INTERFACE KEYS (N-R)

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

Strap R
Strap Q
Strap P
Strap N (Printer Escape Code Transfer)
1 = open (Send ESC code)
0 = closed (do no sent code)

Straps P-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 U
Strap S
Strap T

Straps S-V do not apply to the terminal.

Continued . . .

Figure 7-2. Secondary Terminal Status Example (Continued)

BYTE 12 CONFIGURATION STRAPS W-Z

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

Strap W

Strap Z (Parity Check)
1 = enable
0 = disable

Strap X (Data Speed Select)
1 = data speed signal high
0 = data speed signal low

Strap Y

Straps W and Y 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

Figure 7-2. Secondary Terminal Status Example (Continued)

DEVICE STATUS

The status of the external printer can be obtained by issuing a device status request. This request would typically be made following a print operation or after examining byte 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:

$$E_C\&p\ <\text{device code}>\ \hat{}$$

where <device code> is either 4 or 6. The external printer is 4 on the HP 2392A. If <device code> = 6 (internal printer), the terminal will respond by indicating there is no internal printer.

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

The terminal responds with the sequence E_C p <device code>, followed by three status bytes followed by a terminator (where <device code> = 4).

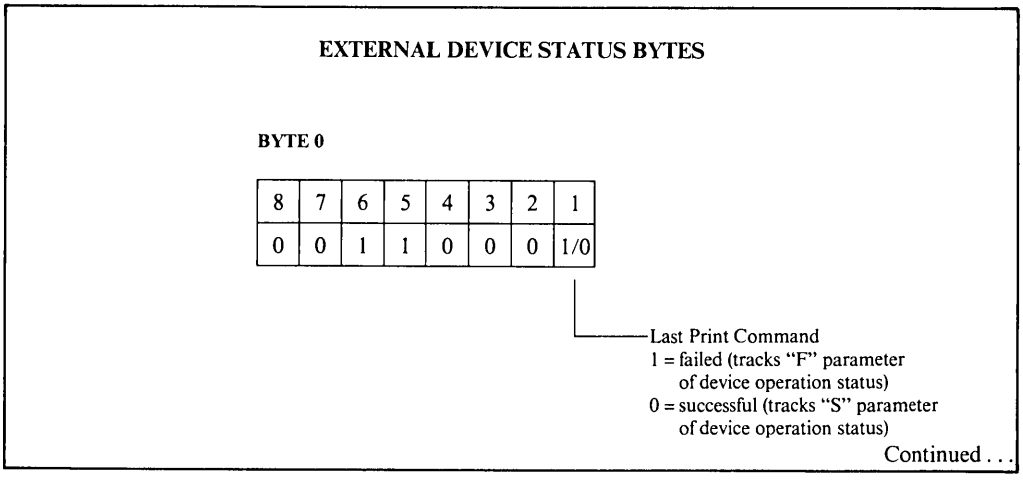


Figure 7-3. External Device Status Bytes

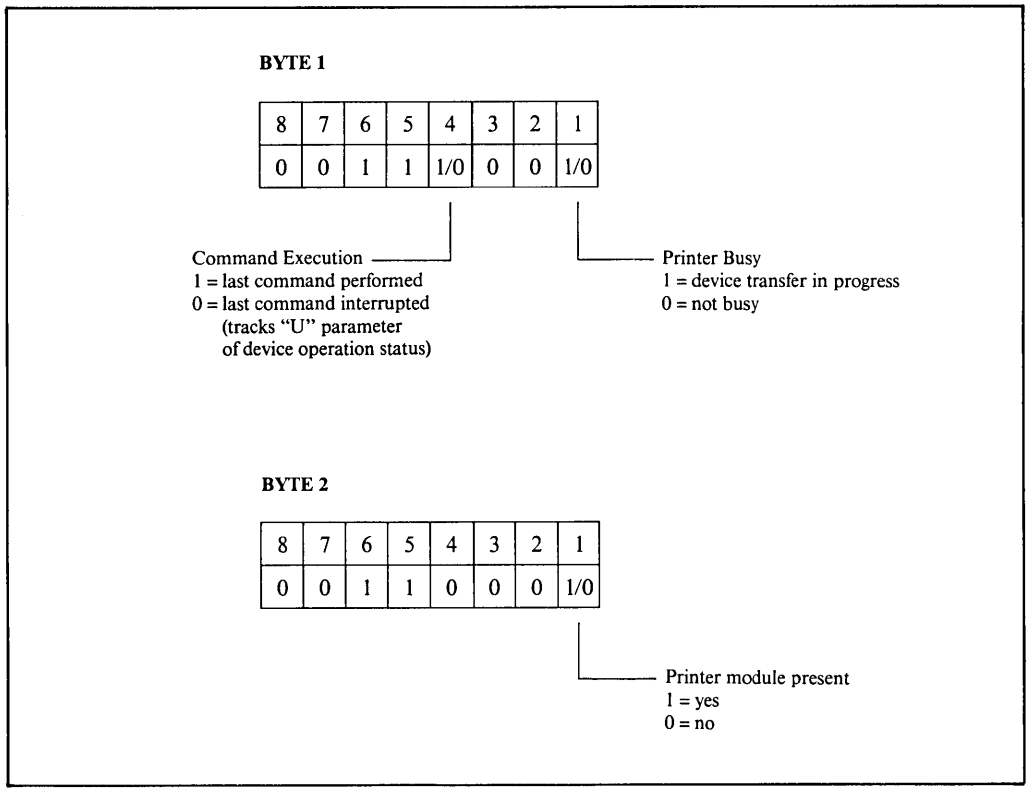


Figure 7-3. External Device Status Bytes (Continued)

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). Pressing **[Return]** will 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).

The **"default config used"** power-on message may indicate that the batteries need to be changed. Instructions on how to change the batteries are provided in Section IX, "Terminal Maintenance Procedures".

After clearing the message (by pressing **[Return]**), you may then reconfigure the terminal as you desire.

No 'TO' device
Press RETURN to clear

You attempted to initiate a device control data transfer (copy line, copy page, copy all) but no destination device is currently defined. Press **[Return]**, use the **"device control"** set of function keys to define an external printer as the "to" device, and then retry the copy operation.

MEMORY FULL
Press RETURN to clear

Memory Lock is enabled, and you have no more space available to enter more data in display memory.

Function locked
Press RETURN to clear

You attempted to use a function that has been locked-out. For example, after receiving the lock menu escape sequence (EC&q 1L) you tried to access a configuration menu or **"modes"** label. Also attempting to assign port 2 to the computer when this port is equipped with an 8-bit parallel module causes the error message to be displayed.

Source = Destination
Press RETURN to clear

You have defined the same device as the **"from"** device and the **"to"** device when copying data. The display is automatically the **"from"** device, and if it is the only device set as the **"to"** device the error message is displayed when you attempt to copy data to a device.

TERMINAL SELF-TESTS

The terminal includes five types of self-tests:

- Power-On Test
- Terminal Test
- Identify ROMs
- Port 1 Test
- Port 2 Test (if module present)

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 pressing **[f5]** in the **service keys** level of function keys).

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 10 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 several times and no softkey labels will appear on the screen.

If one of these error conditions occur, call the nearest HP Sales and Service Office and arrange to have the terminal repaired.

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.
4. Displays the test pattern shown in figure 8-1.

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

[User System] , [f3] service keys , [f5] TERMINAL TEST

Pressing **[Shift]** simultaneously with **[Terminal test]** will start a continuous test and an asterisk will appear in the label: **[Terminal Test*]**. The only way to stop the continuous test is to press **[Terminal Test*]** again.

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 the number 1 or 2. This message contains information identifying the bad ROM chip(s). 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 4. This message also contains information identifying the bad RAM chip(s).Write down the message and call your nearest HP Service representative.

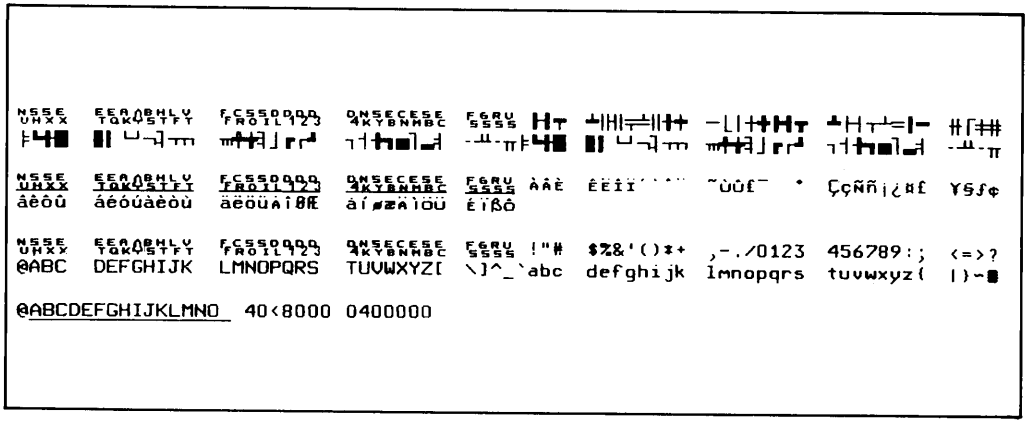


Figure 8-1. Screen Test Pattern (typical)

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:

[User System] , [f3] service keys , [f6] IDENTIFY ROMS

A list similar to the one shown in figure 8-2 is displayed on the screen.

Firmware ROMs

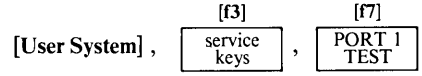
1818 – xxxx

1818 – xxxx

Figure 8-2. ROM Identification Listing

Port 1 Test

The Port 1 test checks the 25-pin RS232C/HP422 port at the rear of the terminal. To enable the Port 1 self test, press the following keys in the sequence shown:



and the test will be executed one time. Now, if you press **[Shift]** simultaneously with **[PORT 1 TEST]**, then a continuous test will start and an asterisk will appear in the label **[PORT 1 TEST*]**. The only way to stop this continuous test is to press **[PORT 1 TEST]** again (even a power down will not stop it).

A test hood (HP part no. 02620-60062 for RS232, 5061-3248 for HP422) must be connected to the port to perform the test, otherwise. "Datacomm Error 1" will appear on the screen.

The loopback test consist of a data loopback operation. While the test is running, "Testing" is displayed on the terminal screen. If no errors are found, the terminal beeps and displays the labels. 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 = Error in Control lines (RS232C only).

3 = Character did not loop back.

4 = Received character NOT same as one transmitted.

5 = Framing error in character.

6 = A character was overrun.

Port 2 Test

Port 2 can be equipped with various modules (RS232C, 8-bit parallel).

When equipped with an RS232C module, the test hood should be 02620-60062.

When equipped with an 8-bit parallel module, no test can be performed. No action is taken when **[PORT 2 TEST]** is pressed.

To initiate PORT 2 press the following keys in the sequence shown:

[User System] , [f3] User System , PORT 2 TEST

To start a continuous test press **[Shift]** **[PORT 2 TEST*]** (an asterisk will appear in the label).

The only way to stop a continuous Port 2 test is to press **[PORT 2 TEST*]** again.

Terminal Maintenance Procedures

SECTION IX

INTRODUCTION

This section provides information on preventive maintenance for your terminal, such as cleaning the screen and keyboard.

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 two batteries located just above the rear panel of your terminal. Figure 9-1 shows the rear panel and the location of the batteries.

The two batteries require no special care or maintenance. They should, however, be replaced every 12 months. You may purchase replacement batteries through conventional retail stores. When doing so, request UNION CARBIDE Type E90 or DURACELL MN9100. You may also order replacement batteries through your local HP Sales and Service Office using the following nomenclature and part number:

HP 2392A Battery
HP Part No. 1420-0255

If your terminal has an external printer you may wish to record the various configuration menus on paper before removing the old batteries.

Replace the batteries as follows:

1. Switch ON the terminal.
2. Squeeze the tabs on either side of the battery support (shown in figure 9-1) with enough pressure to disengage the flanges that hold the battery support in place.
3. Gently pull the support outward until it is completely free from the terminal housing.
4. Remove the old batteries from the support.
5. Install the new batteries in the support making sure that the positive end of the battery matches the positive end of the support (+ "to" + "and" – "to" –).
6. Reinsert the battery support into the terminal. The notch in the base of the support ensures it can only be inserted when correctly positioned. Check that the support engages with the terminal.

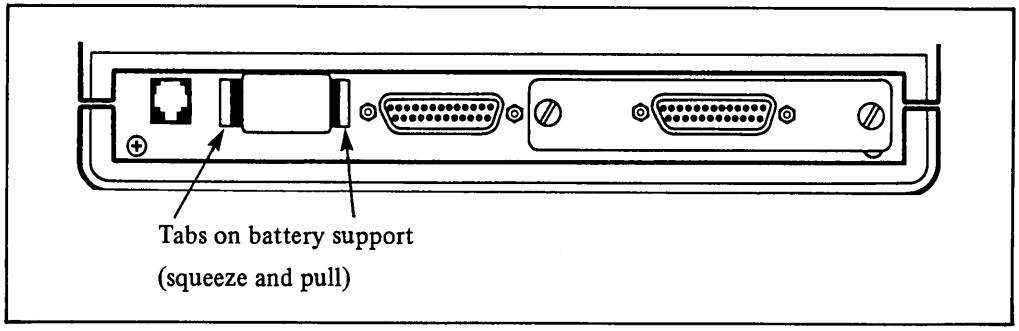


Figure 9-1. Removing the Batteries

Escape Codes

APPENDIX

A

KEY(S)			CODE	FUNCTION
TERMINAL CONTROL FUNCTION				
[Print Enter]	(as used in Local mode)		ESC 0	Copy memory to destinations(s)
[User System] ,	margins / tabs / col ,	SET TAB	ESC 1	Set tab
[User System] ,	margins / tabs / col ,	CLEAR TAB	ESC 2	Clear tab
[User System] ,	margins / tabs / col ,	CLR ALL TABS	ESC 3	Clear all tabs
[User System] ,	margins / tabs / col ,	LEFT MARGIN	ESC 4	Set left margin
[User System] ,	margins / tabs / col ,	RIGHT MARGIN	ESC 5	Set right margin
[User System] ,	margins / tabs / col ,	CLR ALL MARGINS	ESC 9	Clear all margins
			ESC @	Delay one second
[▲]			ESC A	Cursor up
[▼]			ESC B	Cursor down
[►]			ESC C	Cursor right
[◄]			ESC D	Cursor left
[Shift] [CTRL] [Reset]			ESC E	Hard reset
[Shift] [▽]			ESC F	Cursor home down
			ESC G	Move cursor to left margin
[▽]			ESC H	Cursor home up
[Tab►]			ESC I	Horizontal tab (tab right)
[Clear display]			ESC J	Clear display from cursor to end of memory

Continued . . .

KEY(S)		CODE	FUNCTION
TERMINAL CONTROL FUNCTION (Continued)			
[Clear line]		ESC K	Clear line from cursor to end of line
[Insert line]		ESC L	Insert line
[Delete line]		ESC M	Delete line
[Delete char]		ESC P	Delete character
[Insert char]		ESC Q	Start insert character mode
[Insert char]		ESC R	End insert character
[Shift] [▲]		ESC S	Roll up
[Shift] [▼]		ESC T	Roll down
[Next]		ESC U	Next page
[Prev]		ESC V	Previous page
		ESC W	Format mode on
		ESC X	Format mode off
[User System], Modes ,	DISPLAY FUNCTION*	ESC Y	Display Functions mode on
[User System], Modes ,	DISPLAY FUNCTNS	ESC Z	Display Functions mode off
		ESC [	Start unprotected field
		ESC]	End unprotected field
		ESC ^	Primary terminal status request
		ESC *s^	Terminal I.D. status request
		ESC `	Sense cursor position (relative)
		ESC a	Sense cursor position (absolute)
		ESC b	Unlock keyboard
		ESC c	Lock keyboard
		ESC d	Transmit a block of text to computer

Continued . . .

KEY(S)	CODE	FUNCTION
TERMINAL CONTROL FUNCTION (Continued)		
	Esc f	Modem disconnect
[Shift] [Reset Break]	Esc g	Soft reset
[▽]	Esc h	Cursor home up
[Shift] [Tab◄►]	Esc i	Backtab
[CTRL] [Menu]	Esc j	**Display User Key Menu and begin User Key Definition Mode**
[User System] or [Shift] [User System]	Esc k	**Restore normal display and end User Key Definition Mode**
Modes , MEMORY LOCK*	Esc l	Begin Memory Lock mode
Modes , MEMORY LOCK	Esc m	End Memory Lock mode
[f1]	Esc p	Default definition for user definable function key f1
[f2]	Esc q	Default definition for user definable function key f2
[f3]	Esc r	Default definition for user definable function key f3
[f4]	Esc s	Default definition for user definable function key f4
[f5]	Esc t	Default definition for user definable function key f5
[f6]	Esc u	Default definition for user definable function key f6
[f7]	Esc v	Default definition for user definable function key f7
[f8]	Esc w	Default definition for user definable function key f8
[User System] , service keys , TERMINAL TEST	Esc z	Initiate terminal self test
	Esc ~	Secondary terminal status request
	Esc&x1C	Enable the send cursor position mode
	Esc&x0C	Disable the send cursor position mode

CURSOR CONTROL OPERATIONS

NOTE

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

Ec &a <col> c <row> Y	Moves the cursor to column "col" and screen row "row" on the screen (screen relative addressing).
Ec &a <col> c <row> R	Moves the cursor to column "col" and row "row" in memory (absolute addressing).
Ec &a ± <col> c ± <row> Y	Moves the cursor by columns "col" and rows "row" (on the 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.
Ec &a ± <col> c ± <row> R	Moves the cursor by columns "col" and rows "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

Ec &q 0L Unlock all configuration menus.
Ec &q 1L Lock all configuration menus; plus functions **MODIFY ALL, BLOCK MODE, REMOTE MODE** and **AUTO LF.**

} Sequence retained in non-volatile memory.

ESCAPE SEQUENCE	DESCRIPTION	VALUE	<x>
Ec &k <x> A	AUTO LF	OFF	x = 0
		ON	x = 1
Ec &k <x> B	BLOCK MODE	OFF	x = 0
		ON	x = 1
Ec &k <x> C	Caps Lock	OFF	x = 0
		ON	x = 1
Ec &k <x> D	Bell	OFF	x = 0
		ON	x = 1
Ec &k <x> I	None/8 bits	NO (7 bits + existing parity)	x = 0
		YES (none/8 bits)	x = 1
Ec &k <x> K	Auto Keyboard Lock	OFF	x = 0
		ON	x = 1
Ec &k <x> L	Local Echo	OFF	x = 0
		ON	x = 1
Ec &k <x> M	MODIFY ALL	OFF	x = 0
		ON	x = 1

Continued . . .

CONFIGURATION OPERATIONS (Continued)

ESCAPE SEQUENCE	DESCRIPTION	VALUE	<x>
Esc &k <x> N	SPOW Latch	OFF ON	x = 0 x = 1
Esc &k <x> P	Caps Mode ([Caps] key)	OFF ON	x = 0 x = 1
Esc &k <x> R	REMOTE	OFF ON	x = 0 x = 1
Esc &k <x> [	SMOOTH SCROLL	OFF ON	x = 0 x ≠ 0
Esc &s <x> A	XmitFnctn (A)	NO YES	x = 0 x = 1
Esc &s <x> B	SPOW (B) (enable SPOW latch)	NO YES	x = 0 x = 1
Esc &s <x> C	InhEolWrp (C)	NO YES	x = 0 x = 1
Esc &s <x> D	Line/Page (D)	LINE PAGE	x = 0 x = 1
Esc &s <x> G	InhHndShk (G)	NO YES	x = 0 x = 1
Esc &s <x> H	Inh DC2 (H)	NO YES	x = 0 x = 1
Esc &s <x> N	Esc Xfer (N)	NO YES	x = 0 x = 1
Esc &s <x> X	Data Speed Select SR(CH)	LO HI	x = 0 x = 1
Esc &s <x> Z	Check Parity	NO YES	x = 0 x = 1

DATA TRANSFER OPERATIONS

The following escape sequences control data transfer to and from the external printers and display memory. Only effective if printer port present. The HP 2392A does NOT support an internal printer.

ESC &p[<x>d]<x>D — Selects device "<x>" as the destination device (displays "*" in selected "to" device screen label and turns off previous "to" device).
x=3 display, x=4 external device and [<x>d] = optional.

For the following sequences, if "<x> d" (destination) parameter is not specified, the selected "to" devices in the screen labels are assumed to be the destinations.

ESC &p <a>d<b>d<Y> — Copies "<Y>" amount of data to destination devices "<a>" and "<b>". As many destinations as desired can be specified.

<a> and <b>	DEVICE
3	Display.
4	External 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
4	External printer.

ESC &p <x>p<y>u<z>C — Performs the action specified by "<z>" on device "<y>".

<z>	ACTION
0	Generate 1 form feed.
1	Space number of lines specified by "<x>".
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	(ignored).
15	(ignored).
16	(ignored).
17	(ignored).
18	(ignored).
19	(ignored).
20	Turn on Record Mode.

Continued . . .

DATA TRANSFER OPERATIONS (Continued)

<y>	DEVICE
4	External printer.
3	Display.

<x> p

Optional parameter which can define the character to turn off Record mode.
Alternatively, it may be used in conjunction with 1C (parameter <z>) to specify the number of lines to advance e.g. to generate a ten line space on the external device use E_C&p 10p 4u 1C.

E_C **&p <a>d <b>d <x>W <data string>**	Transfers "x" bytes of the data string from the computer to the selected destination device in binary form.
E_C **&p <a>d <b>d 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

E_C [	Starts a field.
E_C]	Ends the field.

FUNCTION KEY AND ERROR MESSAGE OPERATIONS

To enable and disable the function keys ([f1] thru [f8]), use the following escape sequence:

E_C **&j <x>**

<x>	MEANING
A	Display the modes set of function key labels.
B	Enable the user function keys. (The user key labels are displayed).
C	Clear message and restore the current key labels.
@	Remove the function key labels from the screen. The user function keys, however are still active.

To enable or disable the Function Control keys:

S	Disable the [User System] and [Menu] keys, and label "modes".
R	Enable the [User System] and [Menu] keys, and label "modes".

Continued . . .

FUNCTION KEY AND ERROR MESSAGE OPERATIONS (Continued)

To define the function keys:

E_C &f <attribute> a <key> k <label length> d <string length> L <label> <string>

TERM	SYMBOL	MEANING	DEFAULT
<attribute> (a)	0	Normal (N)	0
	1	Local only (L)	
	2	Transmit only (T)	
<key> (k)	1	[f1] function key	1
	2	[f2] function key	
	3	[f3] function key	
	4	[f4] function key	
	5	[f5] function key	
	6	[f6] function key	
	7	[f7] function key	
	8	[f8] function key	
<label length> (d)	0 thru 16	Number of characters in the label. (32 diacritical combinations)	0
<string length> (L)	0 thru 80	Number of characters in the string. (160 diacritical combinations)	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.	

E_C &f <key> E Initiate execution of key: – 1 = [Enter], 1 = [f1] to 8 = [f8]

To replace the function key definition with your own message:

E_C &j <string length> L <message>

<string length> – A number indicating the number of characters in the string (up to 160 characters, 320 diacritical combinations).
<message> – The content of the message

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
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															

Esc &d S Start Security field.

Esc &d s <char> Enable Security field and other selected enhancement(s).

Esc &w 12F Turns on the display screen (top 24 lines).

Esc &w 13F Turns off the display screen (top 24 lines).

CONTROL CHARACTERS

Control characters (hexadecimal 00 to 1F inclusive) are excluded from the control sequence syntax, but may be embedded within a block of text. The following table lists the control characters processed by the terminal, all other control characters are ignored.

Control Character	Hex Code	Terminal Action
ENQ	05	Transmits an ACK character if the "EnqAck" field in the datacomm configuration menu is set to YES.
BEL	07	Sounds the bell.
B_S	08	Moves the cursor one position to the left, stopping at the left margin of current line.
H_T	09	Moves the cursor to the next tab stop, stopping at the right margin if no further tab stops are in the line.
L_F	0A	Executes a line feed.
C_R	0D	Moves cursor to left margin of current line.
S_O	0E	Invokes alternate character set (line drawing).
S_I	0F	Invokes base character set (configured language).
X-ON (D_1)	11	Causes terminal to resume transmission (see Section VI).
X-OFF (D_3)	13	Causes terminal to stop transmitting all control characters except XON and XOFF.
E_C	1B	Introduces a control sequence.

Character Sets and Keyboards

APPENDIX

B

CHARACTER SETS

The terminal offers as standard two character sets: ROMAN 8 and Line Drawing. The ROMAN 8 set is a combination of the USASCII set and the Roman Extension set, and is shown in table B-1 (the terminal does not display the characters located in columns 14 and 15). The Roman Extension set contains "national characters" that have a decimal value from 161 to 223. The Line Drawing Set comprises the 64 characters shown in figure B-2.

The ROMAN 8 is always the Base Set (or Primary Set) which is invoked by default at power-on, by receipt of the Shift-in (SI) control code, or when the cursor goes into a new line (see figure B-1).

The Line Drawing Set is always the Alternate (Secondary) Set which is invoked by receipt of a Shift-out (SO) control code (see figure B-1).

The Shift-in character causes all subsequent characters to be displayed from the ROMAN 8 set. The terminal remains shifted in until receipt of a Shift-out control code.

The Shift-out control code causes subsequent characters to be displayed from the alternate set (Line Drawing). The terminal remains "shifted out" until receipt of a Shift-in control code or until the cursor moves to a new line. See figure B-1.

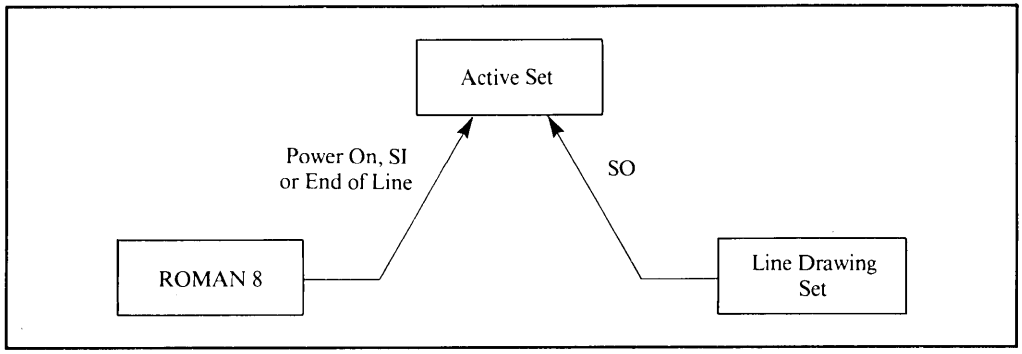


Figure B-1. Accessing ROMAN 8 and Line Drawing Set

Table B-1. ROMAN 8 Characters Supported By 2392A

b8	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
b7	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
b6	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
b5	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

b4	b3	b2	b1		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	0	0	0	NUL	DLE	SP	0	@	P	`	p				¯	â	Å	Á	Þ
0	0	0	1	1	SOH	DC1	!	1	A	Q	a	q			À		ê	î	Ã	þ
0	0	1	0	2	STX	DC2	"	2	B	R	b	r			Â		ô	Ø	ã	
0	0	1	1	3	ETX	DC3	#	3	C	S	c	s			È	º	û	Æ	Đ	
0	1	0	0	4	EOT	DC4	$	4	D	T	d	t			Ê	Ç	á	å	đ	
0	1	0	1	5	ENQ	NAK	%	5	E	U	e	u			Ë	ç	é	í	Í	
0	1	1	0	6	ACK	SYN	&	6	F	V	f	v			Î	Ñ	ó	ø	Ì	–
0	1	1	1	7	BEL	ETB	'	7	G	W	g	w			Ï	ñ	ú	æ	Ó	¼
1	0	0	0	8	BS	CAN	(	8	H	X	h	x			´	¡	à	Ä	Ò	½
1	0	0	1	9	HT	EM	)	9	I	Y	i	y			`	¿	è	ì	Õ	ª
1	0	1	0	10	LF	SUB	*	:	J	Z	j	z			^	○	ò	Ö	õ	º
1	0	1	1	11	VT	ESC	+	;	K	[	k	{			¨	£	ù	Ü	Š	«
1	1	0	0	12	FF	FS	,	<	L	\	l	|			~	Y	ä	É	š	■
1	1	0	1	13	CR	GS	-	=	M	]	m	}			Ù	§	ë	ï	Ú	»
1	1	1	0	14	SO	RS	.	>	N	^	n	~			Û	ƒ	ö	ß	Ÿ	±
1	1	1	1	15	SI	US	/	?	O	_	o	DEL			£	¢	ü	Ô	ÿ	

USASCII Set

Roman Extension Set

Characters stored in memory but not displayed on screen.

Table B-2. Non-displayable Control Character Set

Character	Meaning	Decimal	Octal	Hexadecimal	Generated At Keyboard Using
NUL	Null	0	000	00	[CTRL] [@]
SOH	Start Of Heading	1	001	01	[CTRL] [A]
STX	Start Of Text	2	002	02	[CTRL] [B]
ETX	End Of Text	3	003	03	[CTRL] [C]
EOT	End Of Transmission	4	004	04	[CTRL] [D]
ENQ	Enquiry	5	005	05	[CTRL] [E]
ACK	Acknowledge	6	006	06	[CTRL] [F]
BEL	Bell	7	007	07	[CTRL] [G]
BS	Backspace	8	010	08	[CTRL] [H]
HT	Horizontal Tabulation	9	011	09	[CTRL] [I]
LF	Line Feed	10	012	0A	[CTRL] [J]
VT	Vertical Tabulation	11	013	0B	[CTRL] [K]
FF	Form Feed	12	014	0C	[CTRL] [L]
CR	Carriage Return	13	015	0D	[CTRL] [M]
SO	Shift Out	14	016	0E	[CTRL] [N]
SI	Shift In	15	017	0F	[CTRL] [O]
DLE	Data Link Escape	16	020	10	[CTRL] [P]
DC1	Device Control 1	17	021	11	[CTRL] [Q]
DC2	Device Control 2	18	022	12	[CTRL] [R]
DC3	Device Control 3	19	023	13	[CTRL] [S]
DC4	Device Control 4	20	024	14	[CTRL] [T]
NAK	Negative Acknowledgement	21	025	15	[CTRL] [U]
SYN	Synchronous Idle	22	026	16	[CTRL] [V]
ETB	End Of Transmission Block	23	027	17	[CTRL] [W]
CAN	Cancel Line	24	030	18	[CTRL] [X]
EM	End Of Medium	25	031	19	[CTRL] [Y]
SUB	Substitute	26	032	1A	[CTRL] [Z]
EC	Escape	27	033	1B	[CTRL] [[]
FS	File Separator	28	034	1C	[CTRL] [\]
GS	Group Separator	29	035	1D	[CTRL] []]
RS	Record Separator	30	036	1E	[CTRL] [^]
US	Unit Separator	31	037	1F	[CTRL] [_]

Note that RS is always displayed, whether or not display functions mode is enabled.

Table B-3. National Characters Replaced By USASCII Characters In 7-Bit Mode

KEYBOARDS	CHARACTERS													
USASCII	#	’	<	>	@	[	\	]	∧	‘	{	|	}	~
Danish	§	ˊ	<	>	@	Æ	Ø	Å	∧	ˋ	æ	ø	å	¨
Dutch	#	’	<	>	@	ç	\	§	∧	ˋ	*f*	|	´	¨
Finnish	#	’	<	>	É	Ä	Ö	Å	Ü	é	ä	ö	å	ü
Flemish and French	£	’	<	>	à	°	ç	§	ˆ *	‘	é	ù	è	¨ *
English Canadian and French Canadian	#	’	<	>	@	[	ç	]	∧	ˋ	é	Ç	É	¨
French Swiss and German Swiss	£	ˊ	é	è	à	°	ç	§	∧	ˋ	ä	ö	ü	¨
German	£	’	<	>	§	Ä	Ö	Ü	∧	‘	ä	ö	ü	ß
Italian	£	’	<	>	§	°	ç	é	∧	ù	à	ò	è	ì
Norwegian	#	ˊ	<	>	@	Æ	Ø	Å	∧	ˋ	æ	ø	å	¨
European Spanish	#	ˊ *	<	>	@	¡	Ñ	¿	°	ˋ *	’	ñ	ç	¨ *
Latin Spanish	#	ˊ *	<	>	@	¡	Ñ	¿	∧	ˋ *	’	ñ	ç	¨ *
Swedish	#	’	<	>	É	Ä	Ö	Å	Ü	é	ä	ö	å	ü
U.K.	£	’	<	>	@	[	\	]	∧	‘	{	|	}	~
Line Drawing Elements	┬	╤	╫	╪	╞	╡	··	╨	·	╞	╡	··	╨	·

Note: In 7-bit operations, characters marked by a “*” are diacritical marks. See Table B-4 for diacritical marks in 8-bit operations.

Table B-4. Diacritical marks In 8-bit Operation

KEYBOARDS	DIACRITIC MARKS
USASCII	None
Danish	´ ` ¨
Dutch	´ ^ ` ¨
Finnish	None
Flemish and French	^ ¨
English Canadian and French Canadian	^ ` ¨
French Swiss and German Swiss	´ ^ ` ¨
German	None
Italian	^
Norwegian	´ ` ¨
European Spanish	´ ` ¨
Latin Spanish	´ ` ¨
Swedish	None
U. K.	None
[Extend char] Key	´ ^ ` ¨ ~

7 versus 8 Bit Transmission To The Computer

On national keyboards up to 14 of the characters in the USASCII character are be replaced by national characters (in order that all the characters normally associated with a particular national keyboard are available). The replaced USASCII characters used depend on the configured keyboard (as these are USASCII characters not normally employed by that keyboard). How the terminal handles the national characters for transmission and reception depends on whether it is set for 7 or 8-bit operations in the "Parity/DataBits" field in the datacomm configuration menu.

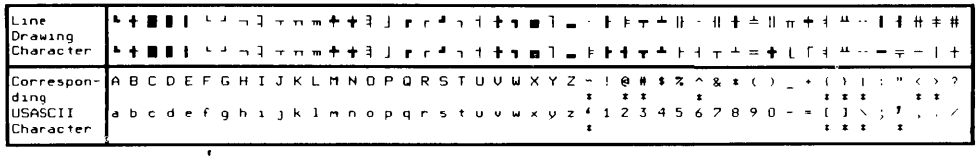
7 bit operations. When the terminal is configured for 7-bit operations (that is one parity bit and seven data bits), characters from the Roman Extension set cannot be directly sent to/received from a computer (as these characters are coded on eight bits, whereas USASCII characters are coded on seven bits). Consequently, the terminal automatically substitutes the corresponding USASCII character code for the Roman Extension characters in transmissions, and vice-versa for reception. (This is why the **[Extend char]** key cannot be used to access additional characters). Note that the correct Roman Extension character is displayed on the terminal.

The Roman Extension characters replacing USASCII characters depends on which national keyboard is specified in the terminal configuration menu (see table B-3).

The Line-drawing character set is affected by this as the elements correspond to USASCII characters, and some of these are not available (as they are replaced by national characters). Consequently, in 7-bit operations using national keyboards, the corresponding Roman Extension characters should be used to access the line-drawing elements (as shown in table B-3).

8 bit operations. When the terminal is configured for eight bit transmissions (that is no parity bit and eight data bits), all the ROMAN 8 characters can be sent to/received from the computer. The characters are coded on eight bits. Bit 8 in USASCII characters is always set to "zero", and to "one" in Roman Extension Characters. In this case the **[Extend char]** key can be used to access characters that do not appear on keycaps.

When using the Line-drawing character set with national keyboards, the USASCII character that corresponds to a line element should be accessed using the **[Extend char]** key (see figure B-2) in order to generate the correct line-drawing element.



With national keyboards, the USASCII characters marked "*" (below) may not be directly available (that is, do not appear on keycaps). When the terminal is configured for 8-bit operations, these USASCII characters must be accessed using the [Extend char] key. When the terminal is configured for 7-bit operations, these USASCII characters are replaced by national characters that must be used instead (see Table B-3).

Figure B-2. Line Drawing Set Elements

Data Transfers To The External Printer

In some HP printers, the Roman Extension set is accessed as an alternate (secondary) set. This secondary set can be accessed either by the shift-out control code (SO) or by the 8th bit treated as a SO control code (when the terminal is configured for 8-bit transmission). During a transfer of data to the printer, the terminal first designates the Roman Extension as the secondary set by sending the Esc)A sequence, then by sending a Shift-out (SO) control code to access the secondary set or by setting the 8th bit of the character to be sent to "1". Such printers belong to type "Extended Roman" (Printer Type = EXT ROMAN in the External Device Configuration Menu).

New HP Printers access characters from the Roman Extension directly with 8th bit = 1 (ROMAN 8 is the Base Set). Such Printers belong to type "ROMAN 8" (Printer Type = ROMAN 8 in the External Device Configuration Menu).

Table B-5. Extended Roman Data Transfers To Printer

	Printer Type = EXTROMAN	**Printer Type = ROMAN 8**
7-bit transmission	Extended Roman is secondary, accessed by <So> control code.	Not Recommended for HP Printers. (Extended Roman) characters are replaced by USASCII codes (see tables B-2 and B-3).
8-bit transmission	Extended Roman is secondary, accessed by 8th bit = 1.	Extended Roman is primary, accessed with full 8-bit control.

KEYBOARDS

The terminal supports 17 different keyboards: Canadian English, Canadian French, Danish, Dutch, Finnish, French, Italian, German, Norwegian, Spanish European, Spanish Latin American, Swedish, Swiss French, Swiss German, UK, USASCII and Flemish.

The keyboard choice is specified in the "Keyboard" field in the terminal configuration menu. Figures B-4 through B-20 show the various national keyboards.

Note that the terminal is able to accomodate any keyboard layout, independently of the physical keyboard installed: the way a key is interpreted depends on the keyboard choice made in the "Keyboard" field. For example, if the "Keyboard" field is equal to "USASCII" though the keyboard layout in French, the terminal will interpret any character typed onto the keyboard as if the key in the corresponding position had been pressed on a USASCII keyboard.

Likewise, when a national keyboard has been selected, any key pressed is interpreted as the corresponding key on the appropriate national keyboard.

On a National Terminal configured for 8-bit transmission, displayed USASCII characters not available on the national keyboard and Roman Extension characters are still accessible by the use of the **[Ext Char]** key. Pressing this key gives access to a new keyboard layout where all Roman Extension characters shown in table B-1 and the 12 displaced USASCII characters are available. See figure B-3.

When a national terminal is configured for 7-bit transmission, only the characters specific to the country specified in the "Keyboard" field (see terminal configuration menu, Section II) can be accessed; in particular, displaced USASCII characters are no longer accessible.

Diacritical Characters

Some national keyboards (e.g. French and Spanish) employ accented characters that do not appear on a keycap but which are formed by combining two characters, an accent (mute) followed by an alpha character (e.g. circumflex plus vowel, etc.).

The terminal handles these characters as follows:

1) When the mute (first part of the diacritical character) is typed. For seven bit data transmission, the corresponding character code (see table B-1) is transmitted and (if echoed) the character is displayed with the cursor under it. For 8-bit transmission the character is displayed with the cursor under it but nothing is transmitted.
2) When the second part of the character is typed:
   a) If this is a valid character (i.e. one giving a valid diacritical combination), the corresponding combination is displayed and the cursor moves to the next character position. In seven-bit transmission, the corresponding code of the second character is transmitted. In 8-bit transmission, the corresponding code for the combination character is transmitted.

b) If the space bar is pressed after the mute key, the mute character remains on the screen and the cursor moves to the next character position. In 7-bit transmission, the code for the space is transmitted. In 8-bit transmission, the code for the mute character is transmitted.

c) If the **[Return]** key is pressed after the mute key, in 8-bit transmission the code for the mute character is sent to the host, followed by the normal action of the **[Return]** key. In 7-bit transmission, the normal action of the **[Return]** key occurs (as the code has already been sent).

d) If any other key is pressed after the mute key, the mute character is erased and the normal action associated with that key is taken. For example, if it is an invalid character, it replaces the previously displayed character. In both 7-bit and 8-bit transmissions the code of this second character is transmitted.

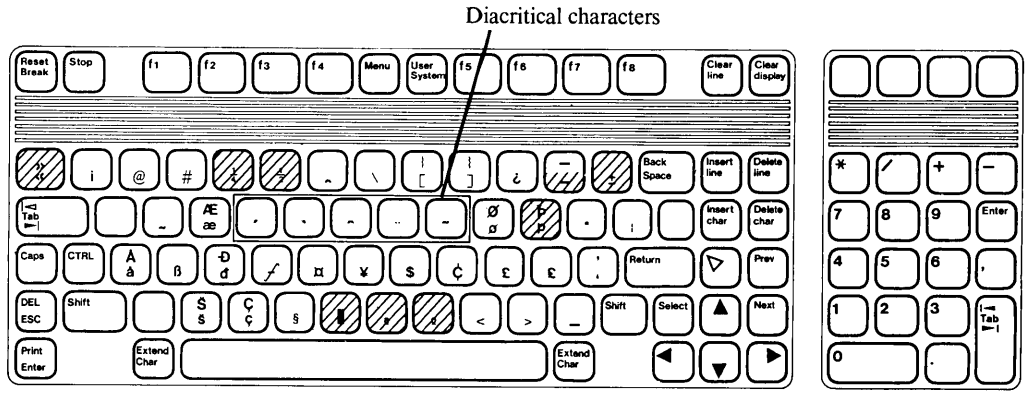
e) If characters from the computer are received prior to the second keystroke, the terminal will process the received data normally. Subsequent keystrokes will be processed as if no data had been received (the mute sequence will continue). If an Escape Sequence is entered through the keyboard prior to the second keystroke, the mute character is ignored.

LANGUAGES

The terminal supports 10 different languages: Danish, Dutch, English, Italian, French, Finnish, German, Norwegian, Spanish, Swedish. The "Language" field in the terminal configuration menu (Section II) specifies which language will be used to display the function keys labels and User Messages.

NOTE

All languages are supported on the standard product (NO option required), this means that a French programmer may use a USASCII keyboard and still receive all the messages in French (the two fields "Keyboard" and "Language" in the terminal configuration menu (see Section II) are totally independent).



For shaded keys, the character is stored in memory but not displayed (replaced by a space).

Figure B-3. Mapping of ROMAN 8 characters (accessed by the **[Extend char]** key)

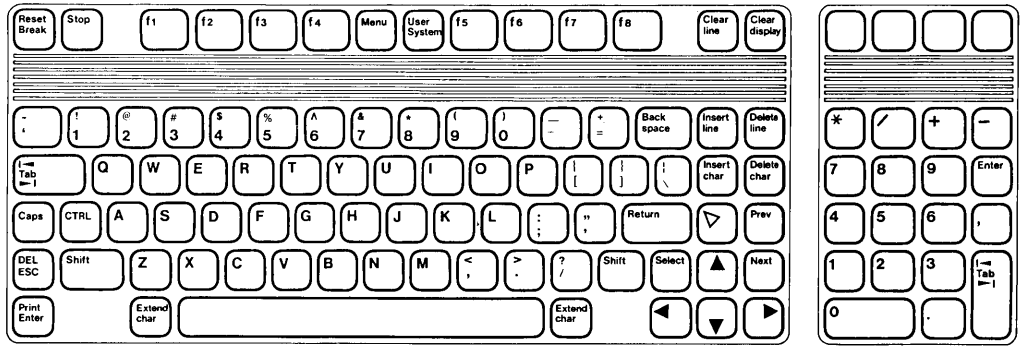


Figure B-4. American USASCII Keyboard (Standard)

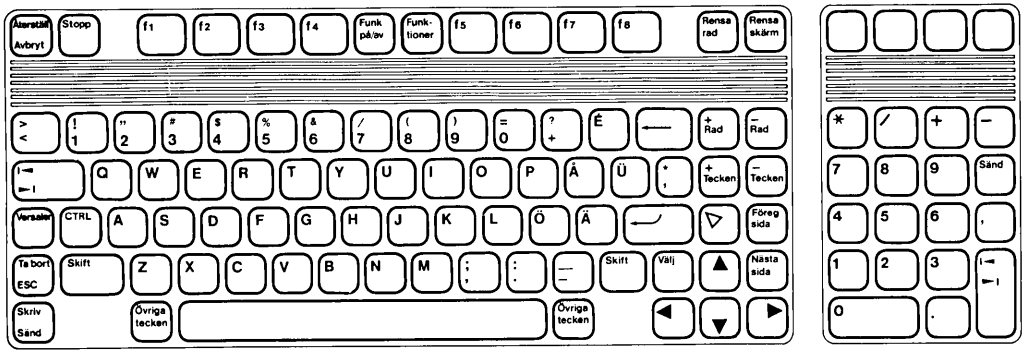


Figure B-5. Swedish Keyboard (option 101)

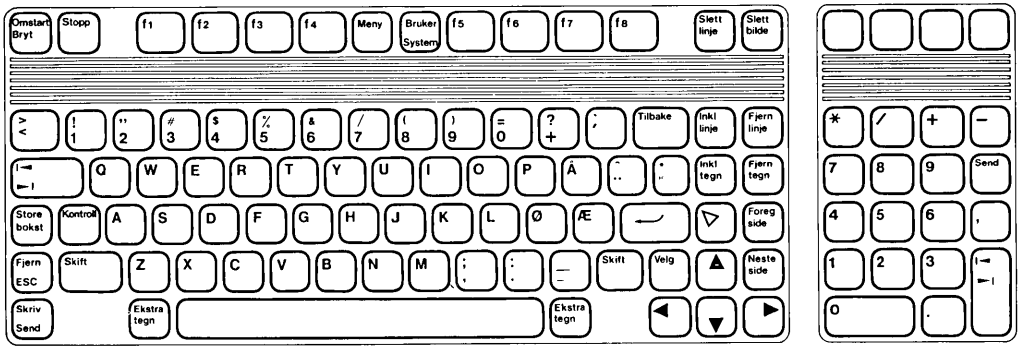


Figure B-6. Norwegian Keyboard (option 102)

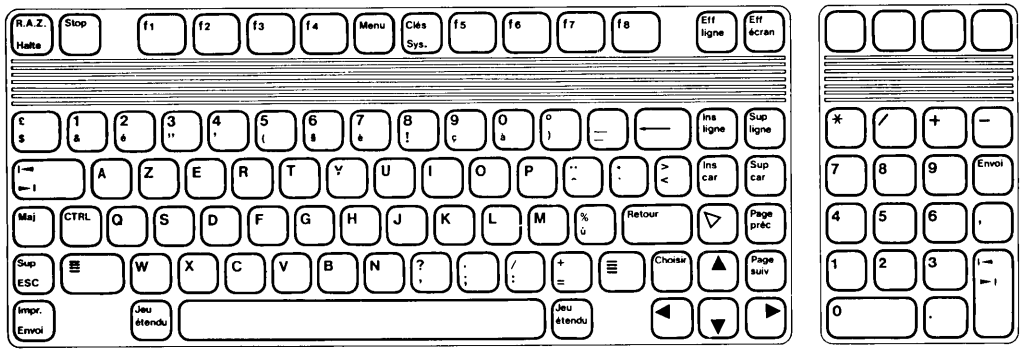


Figure B-7. French Keyboard (option 103)

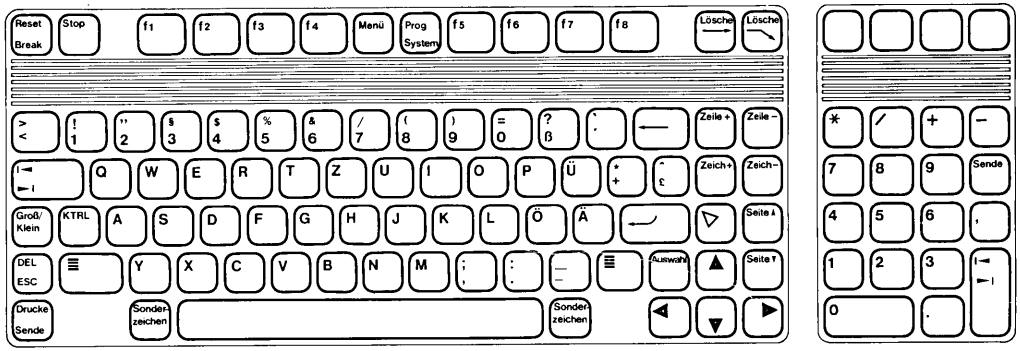


Figure B-8. German Keyboard (option 104)

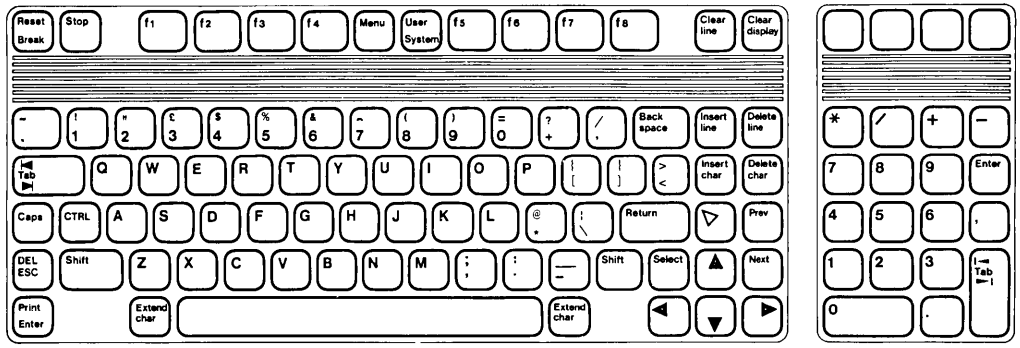


Figure B-9. English (UK) Keyboard (option 105)

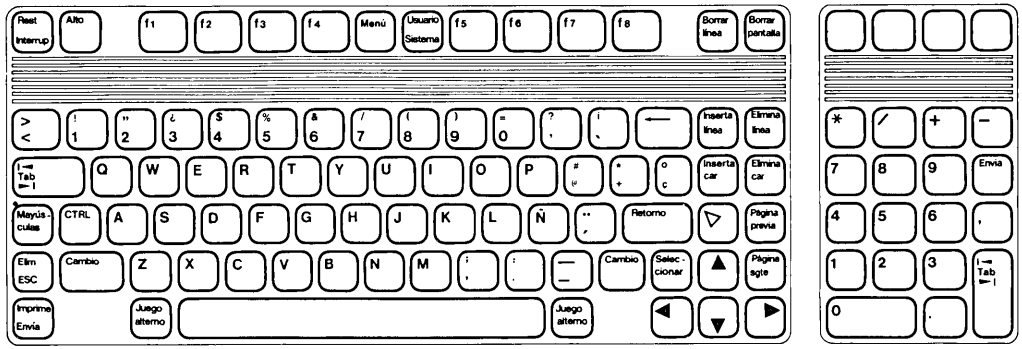


Figure B-10. Spanish (European) Keyboard (option 106)

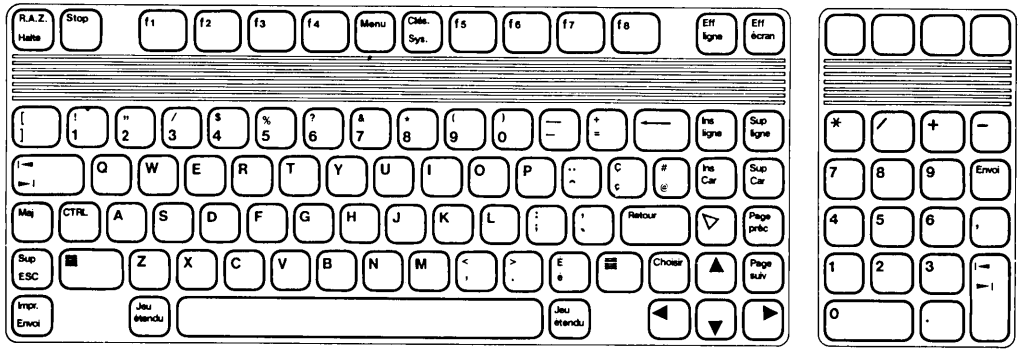


Figure B-11. Canadian (French) Keyboard (option 107)

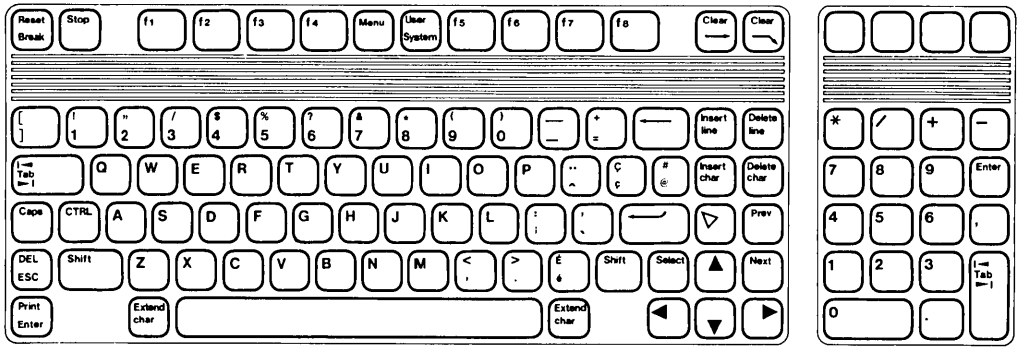


Figure B-12. Canadian (English) Keyboard (option 108)

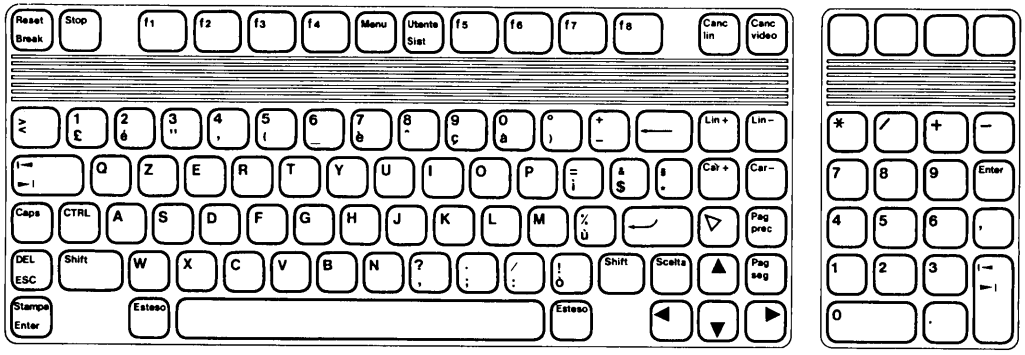


Figure B-13. Italian Keyboard (option 109)

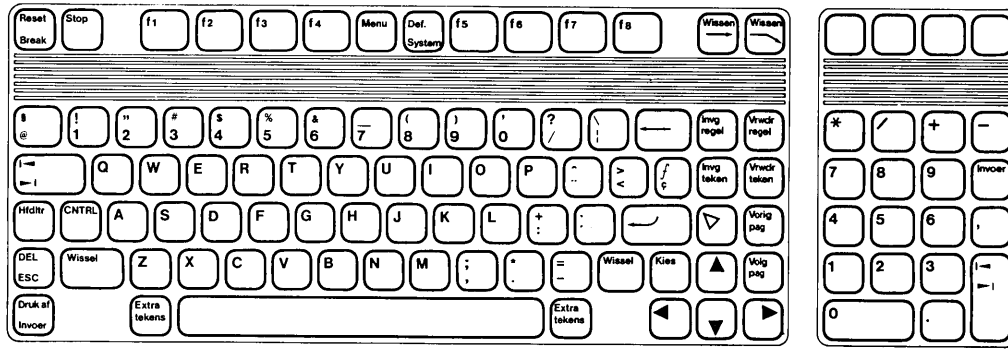


Figure B-14. Dutch Keyboard (option 110)

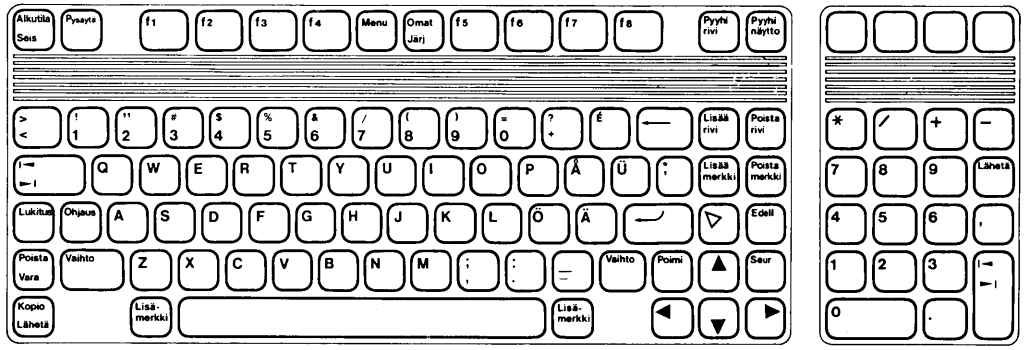


Figure B-15. Finnish Keyboard (option 111)

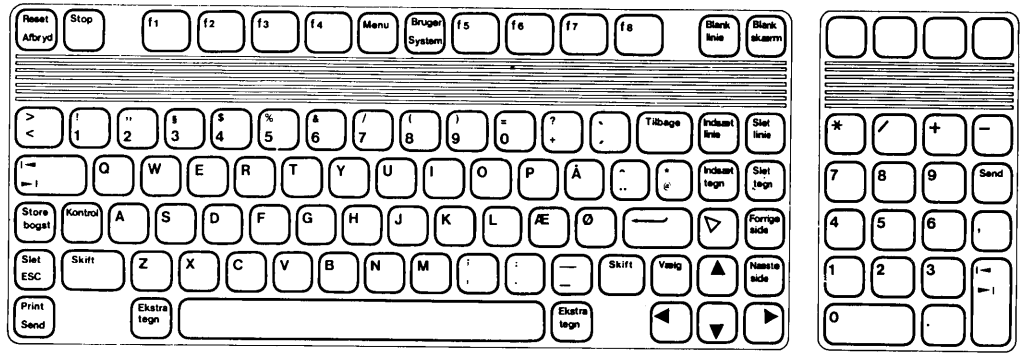


Figure B-16. Danish Keyboard (option 112)

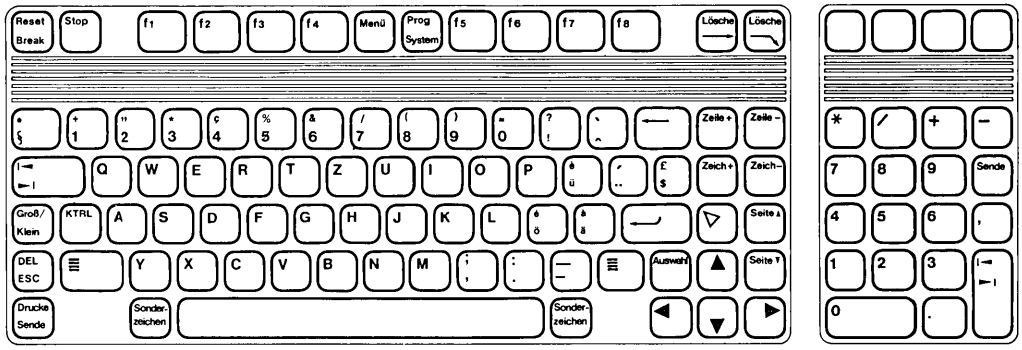


Figure B-17. Swiss (German) Keyboard (option 113)

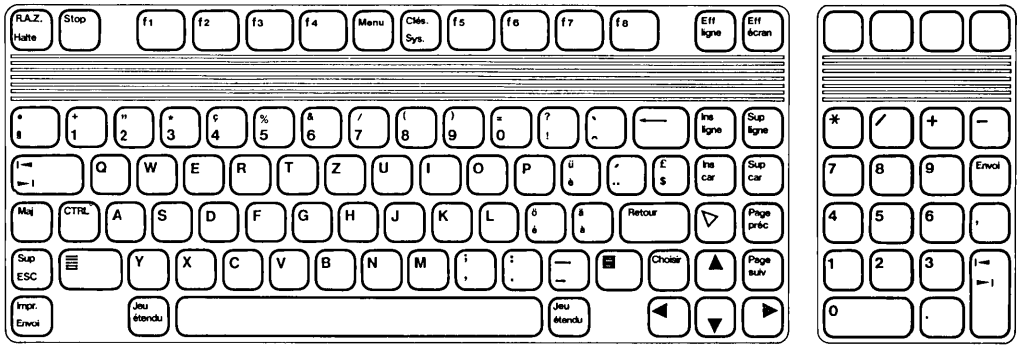


Figure B-18. Swiss (French) Keyboard (option 114)

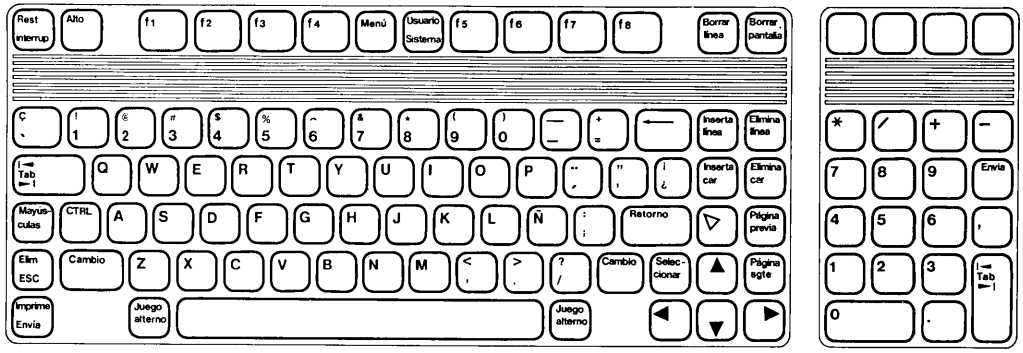


Figure B-19. Spanish (Latin American) Keyboard (option 115)

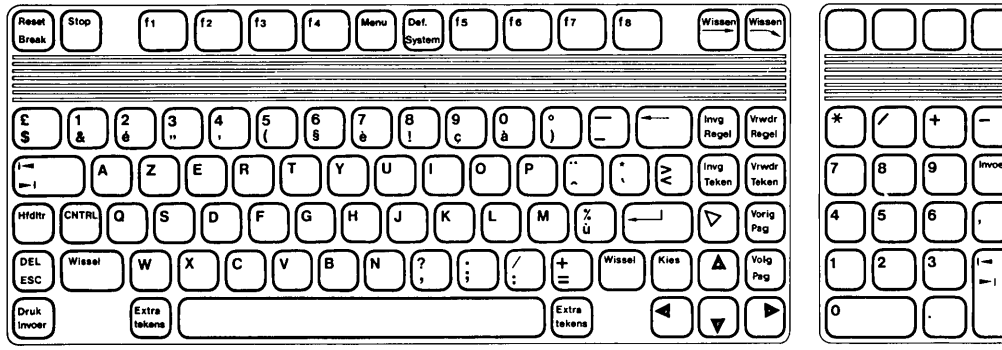


Figure B-20. Flemish Keyboard (option 116)

Index

Subject Page

A

absolute addressing 4-9
addressing:
  absolute (memory) 4-9
  combining absolute and relative 4-11
  cursor relative 4-10
  screen relative 4-9
advance line 5-4
advance page 5-4
alternate character set:
  extended character B-6
  line drawing 4-24, B-1
ASCII character set B-2
ASCII transfers 5-7
asterisk:
  field, datacomm configuration menu .. 6-8, 6-9
  screen labels 1-8
  screen status line 1-10
attribute field, user key 3-11
auto keyboard lock mode 3-10
auto line feed mode 3-7

B

backtab 4-19
battery maintenance 9-1
baud rate:
  field, datacomm configuration menu 6-8
  field, external device
  configuration menu 5-11, 5-12
bell:
  code 3-24
  terminal configuration menu field 2-2, 2-4
binary transfers 5-7
blinking video 4-20
block terminator field,
terminal configuration menu 2-2, 2-9
block mode 3-3
bottom logging 5-3
break 3-23
[Reset Break] key 3-22
brightness control 1-2
buffer:
  overflow 6-12
  receive 6-12

Subject Page

C

cables:
  data communications 6-3
  external device 5-9
caps mode ([Caps] key) 3-8
caps lock mode
(terminal configuration menu field) 2-4, 3-8
character mode:
  line modify mode 3-1
  modify all mode 3-2
  selection 3-3
character set:
  available B-1
  alternate line drawing 4-24, B-1
  extended characters B-6
check parity field,
datacomm configuration menu 6-8, 6-9
checking parity 6-11
cleaning the terminal 9-1
clear display 4-16
clear line 4-17
clearing margins 4-17
clearing tabs 4-18
combining absolute and relative addressing .. 4-11
completion codes, device control 5-6
computer port 1-12
config keys label 2-1
configuration menus:
  datacomm 6-8
  external device 5-12
  lock/unlock 2-11
  programming 2-12, A-4
  terminal 2-3
  user-keys definition 3-12
configuring the terminal 1-10, 2-2, 5-8, 6-7
control characters A-9
control characters A-10
copy all 5-5
copy memory to device 5-6
copy menu 5-6
copy page 5-5
CS (CB) Xmit field,
datacomm configuration menu 6-8, 6-10
CS (CB) Xmit field,
external device configuration menu ... 5-11, 5-13

Index (Continued)

Subject	Page
cursor control	4-2
cursor control operation	A-4
cursor down	4-3
cursor home down	4-3
cursor home up	4-2
cursor left	4-4
cursor relative addressing	4-10
cursor right	4-3
cursor up	4-3
cursor position mode	3-9
cursor sensing	4-8

D

DC1/DC2 handshake	2-6, 3-16, 5-13, 6-15
datacomm/extdev field, terminal configuration menu	2-2, 2-5
datacomm module	1-5
data communications:	
cables	6-3
configuration menu	6-8
description	6-2
test hood	8-5
data bits: field, datacomm configuration menu	6-8, 6-9
field, external device configuration menu	5-11, 5-12
start/stop	6-11
data logging:	
bottom	5-3
top	5-3
data transfers, computer to device	5-7
data transfers, display to device	5-4
defining user keys locally	3-11
defining user keys programmatically	3-12
delete character	4-15
delete line	4-13
designing forms	4-23
device configuration menu	5-11, 5-12
device control completion codes	5-6
device control labels	1-9, 5-4
device modes labels	1-9, 5-2
device port	1-12
device status	7-12
diacritic characters	B-8

Subject	Page
disable display	4-20, A-7, A-9
disable keyboard	3-10, 3-14, A-2
display enhancements	4-20
display functions mode	3-6
display lock	3-5, A-9
drawing forms	4-24

E

echo	2-5
edit operations	4-12
enable keyboard	3-10, 3-14, A-2
enhancements, display	4-20
EnqAck field, datacomm configuration menu	6-8, 6-9
EnqAck handshake	5-7, 6-14
[Print Enter] key	3-15
error messages	8-1
errors, receive	6-12
escape sequences:	
available	A-1
transfer to computer	2-6
transfer to printer	2-10
EscXfer (N) field, terminal configuration menu	2-2, 2-10
[Extend char] key	3-23
external device:	
cabling	5-9
configuration menu	5-11, 5-12
control labels	5-4
port	1-12
status	7-12

F

fields, protected and unprotected	4-26
FldSeparator field, terminal configuration menu	2-2, 2-9
forms (format) mode	4-26
forms, designing and using	4-23
full duplex operations	6-13
function key hierarchy	1-9
function keys	1-8

Index (Continued)

Subject Page

H

half bright 4-20
handshaking 2-6, 6-15
hard reset 3-22
hierarchy, function keys 1-9
home down 4-3
home up 4-2

I

ID status 7-2
identify ROMs 8-4
InhDC2 (H) field,
terminal configuration menu 2-2, 2-6
InhEolWrp (C) field,
terminal configuration menu 2-2, 2-5
InhHndShk (G) field,
terminal configuration menu 2-2, 2-6
insert character 4-13
insert line 4-12
installing a point-to-point configuration 6-6
inverse video 4-20

K

key definition field, user key 3-11
keyboard:
  auto-lock mode 3-10
  cleaning 9-3
  control 3-1
  enabling/disabling 3-10, 3-14, A-2
  field, terminal configuration menu 2-2, 2-3
  groups of keys 1-6
  lock 3-10, 3-14, A-2
  national B-7
keys:
  [Extend char] 3-23
  function keys 1-7
  [Menu] 1-10, 3-1, 3-11
  [Print Enter] 3-15
  [Select] 3-23
  [Stop] 3-23
  [User System] 1-7, 3-1

Subject Page

L

label field, user key 3-11
labels, hierarchy function key 1-9
line modify mode 3-1
Language field,
terminal configuration menu 2-2, 2-3
Line/Page (D) field,
terminal configuration menu 2-2, 2-9
Local echo field,
terminal configuration menu 2-2, 2-5
local mode 3-4, 6-13
lock:
  configuration 2-11
  display 3-5, A-9
  keyboard 1-10, 3-10, 3-14
logging:
  bottom 5-3
  top 5-3

M

margins, setting and clearing 4-17
memory 1-3, 1-4
memory addressing 4-6
memory lock mode 3-5
menu:
  copying to a printer 5-6
  datacomm configuration 6-8
  external device configuration 5-11, 5-12
  terminal 2-2, 2-3
  user key definition 3-12
[Menu] key 1-10, 3-1, 3-11
messages, terminal error 8-1
mode:
  auto keyboard lock 3-10
  auto line feed 3-7
  block 3-3
  caps 3-8
  caps lock 3-8
  character 3-3
  display functions 3-6
  display lock/overflow protect 3-5
  forms (format) 3-9

Index (Continued)

Subject	Page
line modify	3-1
local	3-4
logging (bottom/top)	5-3
memory lock	3-5
modify all	3-2
record	5-1
remote	3-4
send cursor position	3-9
smooth scroll	3-5
modem	6-5
modem indicator	1-10, 6-9
modem disconnect	3-24
modes label	1-9
modify all mode	3-2
modify mode, line	3-1

N

national keyboards	B-7
next page	4-5

O

options	1-4
overflow protect	3-5

P

pacing mechanism	6-14
page, next/previous	4-5
parity checking	6-11
Parity/Databits field, datacomm configuration menu	6-8, 6-9
Parity/Databits field, external device configuration menu	5-11, 5-12
point-to-point configuration, installation	6-7
point-to-point programming information	6-11
port test	8-5
power-on test	8-2
previous page	4-5
primary terminal status	7-3
[Print Enter] key	3-15
printer:	
cabling	5-9
control	5-1
copying to	5-5

Subject	Page
data transfers to	5-4, A-6
port	1-12
receiving data from computer	5-7
receiving data from display	5-4
selecting as destination	5-4
transferring escape sequences	2-10
Printer Nulls field, external device configuration menu	5-11, 5-12
Printer Type field, external device configuration menu	5-11, 5-12, B-7
priority, data transfers	6-16
protected fields	4-26

R

receive buffer	6-12
receive errors	6-12
record mode	5-1
RecvPace field, datacomm configuration menu	6-8, 6-10
relative addressing:	
cursor	4-10
screen	4-9
remote mode	3-4, 6-13
reset:	
hard	3-22
soft	3-22
[Reset Break] key	3-22
ReturnDefault field, terminal configuration menu	2-2, 2-10
roll text:	
down	4-5
up	4-4
ROMs, identify	8-4

S

screen:	
cleaning	9-1
labels	1-9
relative addressing	4-9
scrolling	4-4, 4-5
secondary status	7-3
[Select] key	3-23
self-tests:	
identify ROMs	8-4
ports	8-5, 8-6

Index (Continued)

Subject	Page
power-on	8-2
send display	3-16
setting margins	4-17
setting tabs	4-18
shift-in code	B-1
shift-out code	B-1
smooth scroll mode	3-5
soft reset	3-22
specifications	1-3
SPOW (B) field, terminal configuration menu	2-2, 2-4, 2-12
SR (CH) field, datacomm configuration menu	6-8, 6-9
SRRInvert field, external device configuration menu	5-11, 5-13
SRRXmit field, external device configuration menu	5-11, 5-13
Start Col field, terminal configuration menu	2-2, 2-6
status:	
device	7-12
ID	7-2
primary	7-3
secondary	7-3
stop/start bits	6-11
[Stop] key	3-23
[User System] key	1-7

T

tab:	
back	4-19
forward	4-19
setting/clearing	4-18
terminal:	
cleaning	9-1
configuration menu	2-3
ports	1-10
status	7-1
tests	8-2
Terminal Id field, terminal configuration menu	2-2, 2-3
text editing	4-12
"to" device	5-4
top logging	5-3

U

underline enhancement	4-20
unlock configuration	2-11
unprotected fields	4-26
user-definable keys	3-10
user keys, defining locally	3-11
user keys, defining programmatically	3-12
user key definition menu	3-12
[User System] key	1-7

V

video enhancements	4-20

W

wait code	3-24

X

Xmit Pace field, datacomm configuration menu	6-8, 6-10
Xmit Pace field, external device configuration menu	5-11, 5-13
XmitFnctn (A) field, terminal configuration menu	2-2, 2-6
X-ON/X-OFF handshaking	6-14

02390-90001

Printed in U.S.A. 04/1984
Manual part number : 02390-90001E0484