2382A

Office Display Terminal

HEWLETT
PACKARD

NOTICE

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

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

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

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 program language without the prior written consent of Hewlett-Packard Company.

Copyright © 1981 by HEWLETT-PACKARD COMPANY

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

Printed in U.S.A. 7/81

HP Computer Museum
www.hpmuseum.net

For research and education purposes only.

PREFACE

The HP 2382A Display Terminal is a versatile character, line, or block mode CRT terminal.

This reference manual contains detailed information for the HP 2382A which will allow you to configure, test, and use the terminal.

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

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

TABLE OF CONTENTS

Computer Museum

Section I — Page

GENERAL DESCRIPTION

Introduction 1-1
Keyboard 1-3
Function Keys 1-3
Configuring the Terminal 1-5
Data Communications 1-5
Self-Test 1-5

Section II — Page

CONFIGURING THE TERMINAL

Introduction 2-1
Configuration Function Keys 2-1
Terminal Configuration Menu 2-1
  Lock/Unlock Configuration Menus 2-7
Setting Configuration Parameters with
  Escape Codes 2-7
Power On Configuration Switches 2-8

Section III — Page

KEYBOARD CONTROL

Introduction 3-1
Selecting Modes 3-1
  Remote/Local Modes 3-1
  Character/Block Modes 3-1
  Format Mode 3-2
  Line Modify Mode 3-2
  Modify All Mode 3-2
  Auto Line Feed Mode 3-3
  Memory Lock Mode 3-3
    Overflow Protect 3-3
    Display Lock 3-3
  Display Functions Mode 3-3
  Caps Mode 3-4
  Caps Lock Mode 3-4
User-Definable Keys 3-4
  Defining Keys Locally 3-5
  Defining Keys Programmatically 3-6
  Controlling the User Keys Menu
    Programmatically 3-6
  Controlling the Function Key Labels
    Programmatically 3-6
ENTER Key 3-7
Send Display (ESC d) 3-9
Enable/Disable Keyboard 3-10
Soft Reset 3-10
Hard Reset 3-10
Break 3-10
Bell 3-11
Wait 3-11
Modem Disconnect 3-11

Section IV — Page

DISPLAY CONTROL

Introduction 4-1
Cursor Control 4-1
  Home Up 4-1
  Home Down 4-2
  Move Cursor Up 4-2
  Move Cursor Down 4-2
  Move Cursor Right 4-2
  Move Cursor Left 4-2
  Roll Text Up 4-2
  Roll Text Down 4-3
  Next Page/Previous Page 4-3
  Memory Addressing Scheme 4-4
  Row Addressing 4-4
  Column Addressing 4-5
  Cursor Sensing 4-5
  Cursor Positioning 4-5
  Screen Relative Addressing 4-5
  Absolute Addressing 4-6
  Cursor Relative Addressing 4-6
  Combining Absolute and Relative Addressing ... 4-7
Edit Operations 4-7
  Insert Line 4-7
  Delete Line 4-7
  Insert Character 4-8
  Delete Character 4-8
  Clear Display 4-9
  Clear Line 4-9
Setting and Clearing Margins 4-10
Setting and Clearing Tabs 4-10
Tab 4-11
Back Tab 4-11
Display Enhancements 4-11
Designing and Using Forms 4-12
  Drawing Forms 4-12
Forms Mode (Format Mode) 4-15
  Protected Fields 4-15
  Unprotected Fields 4-15

Section V — Page

DATA COMMUNICATIONS

Introduction 5-1
Hardwired Connections 5-2
Modem Connections 5-2
  Modem Considerations 5-3
Installing a Point-to-Point Configuration 5-3
  Cabling 5-3
Configuring the Terminal 5-3
Point-to-Point Programming Information 5-6
  Start and Stop Bits 5-6
  Parity Checking 5-6
  Receive Buffer 5-6
  Receive Errors 5-7

Section V, Continued Page
DATA COMMUNICATIONS
Local/Remote Modes 5-7
Full-Duplex Operation 5-7
Pacing Mechanism 5-7

Section VI Page
STATUS
Introduction 6-1
Interpreting Status 6-1
Terminal ID Status 6-2
Terminal Status 6-2
Primary Terminal Status 6-2
Secondary Terminal Status 6-2
Primary Status Bytes 6-3
Secondary Status Bytes 6-5

Section VII Page
ERROR MESSAGES AND SELF-TESTS
Errors 7-1
Terminal Self-Tests 7-1
Power-On Test 7-1
Terminal Test 7-1
Identify ROMS 7-2
Datacomm Test 7-2

Appendix A
TERMINAL MAINTENANCE PROCEDURES

Appendix B
ESCAPE CODES

Appendix C
CHARACTER SETS
7-Bit vs. 8-Bit Operation C-7
Alternate Character Set Selection C-7
ISO/USASCII Character Set C-9
Extended Roman Character Set C-9
Line Drawing Set C-10

ILLUSTRATIONS

Title	Page
HP 2382A Display Terminal	1-1
HP 2382A Keyboard	1-3
Function Keys and Screen Labels	1-4
Mode Selection Key Labels	1-4
HP 2382A Function Key Hierarchy	1-4
User Keys Definition Menu	1-5
Default User Key Labels	1-5
Sample User-Supplied User Key Labels	1-5
Terminal Configuration Menu	2-1
Power On Configuration Switches	2-8
Screen-Labeled Function Keys	3-1
User Keys Definition Menu	3-5
The "Roll" Data Functions	4-3
Previous Page and Next Page Concepts	4-4
Row Addressing	4-4
Column Addressing	4-5
Character Insert with Margins	4-8
Character Delete with Margins	4-9
Line Drawing Set Elements	4-13
Sample Data Entry Form	4-13
Completed Data Entry Form	4-14
Base Set Equivalent for Data Entry Form Illustrated in Figure 4-9	4-14
Use of Shift Shift-In and Shift-Out Codes	4-16
HP 2382A Display Terminal, Rear View	5-3
Terminal Power-On Configuration Switches	5-3
Datacomm Configuration Menu	5-4
Primary Terminal Status Example	6-2
Secondary Terminal Status Example	6-4
Screen Test Pattern	7-1
Screen Test Pattern, Option 202 (Line Drawing Character Set)	7-2
ROM Identification Listing	7-2
Swedish/Finnish Keyboard (Option 001)	C-4
Danish/Norwegian Keyboard (Option 002)	C-4
French Keyboard (Option 003), AZERTY Layout	C-5
French Keyboard (Option 003), QWERTY Layout	C-5
German Keyboard (Option 004)	C-6
United Kingdom Keyboard (Option 005)	C-6
Spanish Keyboard (Option 006)	C-7
Line Drawing Set Elements	C-10
Sample Data Entry Form	C-11

TABLES

Title	Page
Specifications	1-1
Terminal Configuration Menu Fields and Power-On Configuration Switches	2-2
Configuration Function Keys f4, f5, f7, and f8	2-7
Power-On Configuration Switch Functions	2-8
ENTER Key Operation	3-7
Data Communications Cables	5-2
Modems	5-2
Datacomm Configuration Menu Fields and Power on Configuration Switches	5-4
ASCII Status Characters	6-1
ASCII Character Set	C-2
ASCII (seven bit) Character Codes	C-3

GENERAL DESCRIPTION

SECTION I

INTRODUCTION

The HP 2382A Display Terminal (figure 1-1) is a versatile general purpose CRT terminal that offers the following features:

- CRT screen size of 114.3 mm (4.5 inches) by 162 mm (6.4 inches).
- Screen capacity of 24 lines of up to 80 characters each (1920 characters total). Two additional lines for system soft key labels.
- Eight user-programmable function keys (f1 through f8) are provided. The function of these keys is displayed in two rows of labels across the bottom of the display screen. The labels are definable (up to 16 characters each). The character strings returned when the keys are pressed can be loaded with up to 80-character strings.
- Two-character user-programmable RETURN key.
- Optional line drawing character set.
- Screen labeled system function keys (for selecting operating modes and performing other terminal control functions).
- Terminal configuration operations are performed through keyboard entries into formatted menus displayed on the screen, or by setting the switches on the back panel which are used to store the proper "power on" configuration.
- Full editing capabilities (insert/delete/clear line and insert/ delete character).

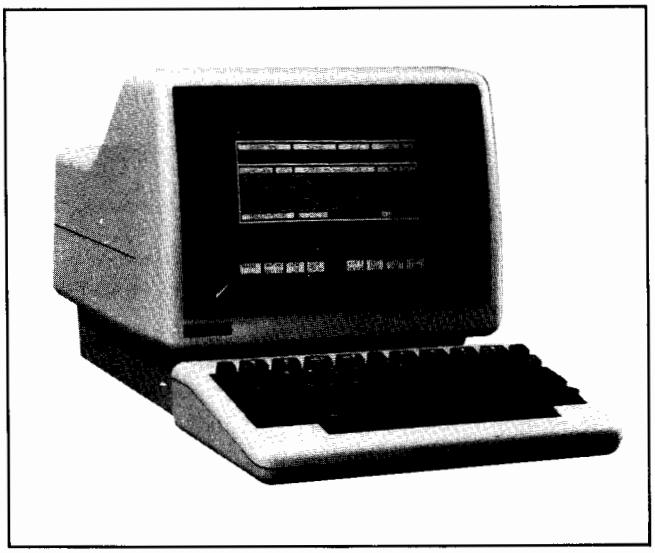


Figure 1-1. HP 2382A Display Terminal

- Display enhancements (inverse video, blinking, underlining, and half-bright).
- Adjustable margins and tab stops.
- Programmatic cursor sensing and addressing.
- Easy-to-use keyboard
- Character, line, or block mode operation.
- Extensive self-test capability.

Table 1-1 lists the specifications for the HP 2382A.

Table 1-1. Specifications

GENERAL	
Screen Size:	114.3 mm (4.5 inches) by 162 mm (6.4 inches)
Screen Capacity:	24 lines of 80 columns (1920 characters). Two additional lines for system soft key labels.
Character Generation:	7 by 11 character in 9 by 15 dot cell
Character Size:	1.83 mm (.072 inches) by 3.22 mm (.127 inches)
Character Set:	128 ASCII character set standard.

Table 1-1. Specifications (Continued)

Cursor:	Blinking-underline
Display Enhancements:	Inverse, underline, blinking, half-bright
Display Modes:	White on black
Refresh Rate:	60 Hz (50 Hz optional)
Memory:	20 K-byte ROM; 3840 bytes (2 pages) display memory; 256 bytes of RAM for data communications buffer
Keyboard:	Detached, with telephone type cable 1.8 m (6 feet fully extended); Full ASCII keyboard; 8 screen labeled keys; auto-repeat; N-key rollover
Operating Modes:	Remote; Local; Character, Line, Block; Forms, Non-Forms.
Transmission Modes:	Full duplex, asynchronous point-to- point. External data communications via modems such as the Bell 103A.
Electrical Interface:	Electrical Industry Association (EIA) Standard RS-232-C
Data Rates:	110, 134.5, 150, 300, 600, 1200, 1800, 2400, 4800, 9600 baud.
Parity:	Selectable: Even, odd, zero, one, none

PHYSICAL CHARACTERISTICS

Weight:	Approximately 22 pounds
Dimensions:	Display Monitor: 314 mm wide by 305 mm deep by 282 mm high (12.3 inches by 12.0 inches by 11.1 inches) Keyboard: 300 mm wide by 180 mm deep by 62 mm high (11.8 inches by 7.1 inches by 2.4 inches)

POWER REQUIREMENTS

Input Voltage:	240 Volts 50 Hz, 100 Volts 50/60 Hz 220 Volts 50 Hz (depending on option)
Power Consumption:	75 watts

OPTIONS

-001	Swedish/Finnish Keyboard and ROM
-002	Danish/Norwegian Keyboard and ROM
-003	French Keyboard and ROM
-004	German Keyboard and ROM
-005	United Kingdom Keyboard and ROM
-006	Spanish Keyboard and ROM
-013	50 Hz, 240 V Power
-014	60 Hz, 100 V Power
-015	50 Hz, 230 V
-016	50 Hz, 115 V
-202	Line Drawing Character Set

Computer Museum

KEYBOARD

The keyboard is similiar to that used for standard office typewriters. It has 68 keys that include eight function keys. Most of the remaining keys support the ASCII-coded character set (see figure 1-2).

An optional Line Drawing Set ROM can be installed and assigned to the keyboard keys. National character sets and keyboards are also optional.

FUNCTION KEYS

The group of 8 light colored keys in the top row of the keyboard perform different functions depending upon which keystrokes have been performed. At any given time the applicable labels for these keys appear across the bottom of the display screen (figure 1-3). When you press one of these keys and the SHIFT key simultaneously then the keys function as display control keys. This order can be reversed by setting the Shift Functions Keys field in the terminal configuration to "YES".

When you press the AIDS key and then the modes function 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, the function keys are automatically initialized as mode selection keys.

When you press the AIDS key and then the [illegible] function key, the eight function keys become general control keys that you use for configuring the terminal, setting and clearing margins and tab stops, accessing the Service Keys and so forth. The entire set of system function key labels is illustrated in figure 1-5. Pressing AIDS always reinitializes the function keys to the top level of functions (labels) shown in figure 1-5.

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

1. If a key label contains any lowercase letters, pressing the key will transfer you to another level of system function keys.

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

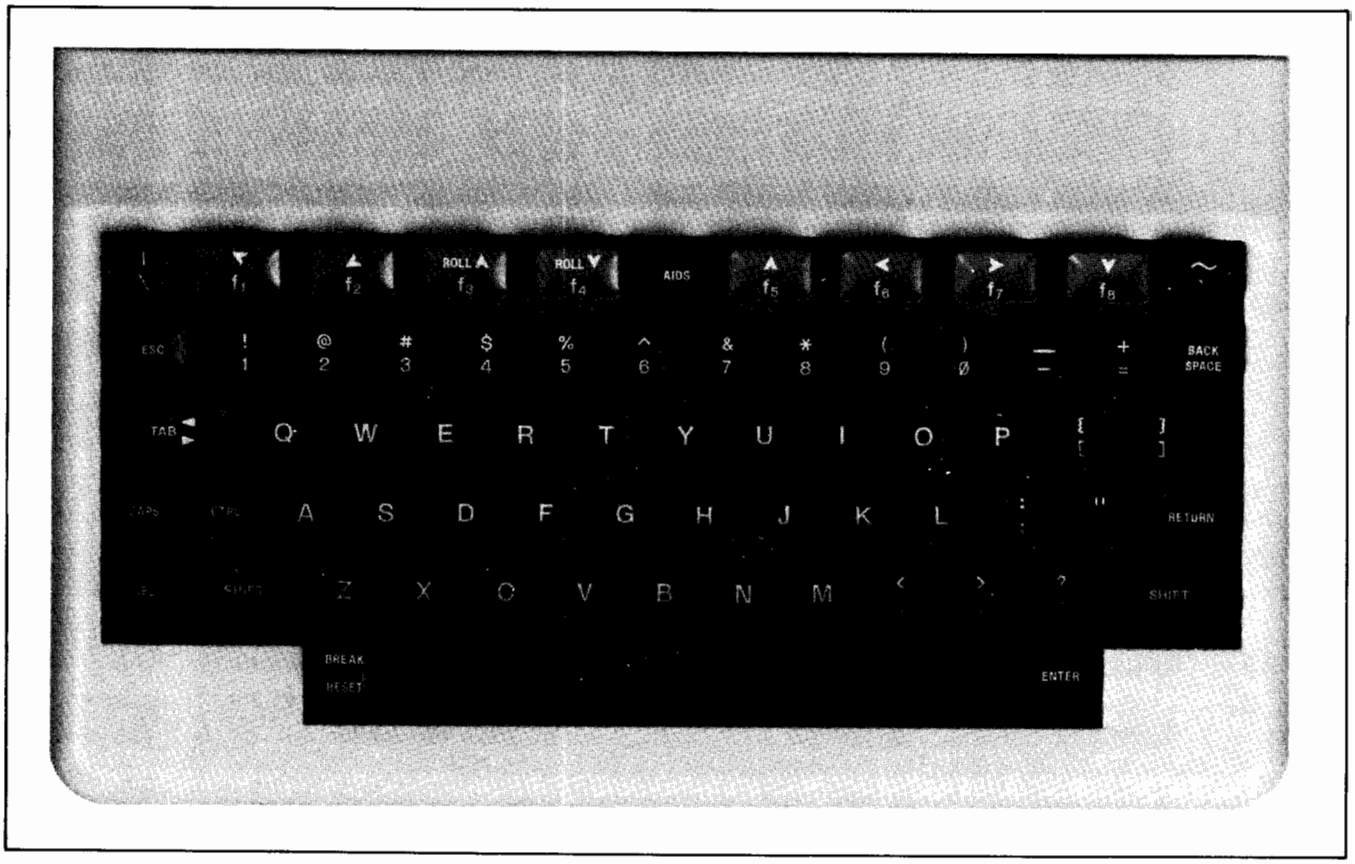


Figure 1-2. HP 2382A Keyboard

The key corresponding to [auto LF], for example, sets the automatic line feed function on; whereas the key corresponding to [config keys] transfers you to the configuration function keys. When you press the [define user key] fuction key 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 the eight user keys. To enable the eight user keys, press the [SHIFT] and [AIDS] key simultaneously. Figure 1-7 shows the default user key screen labels and figure 1-8 shows some sample user-defined user key screen labels.

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

Pressing the [CTRL], [SHIFT] and [AIDS] key disables the function keys and removes the key labels from the screen. To re-enable the keys, press the [AIDS] key with or without the [SHIFT] key.

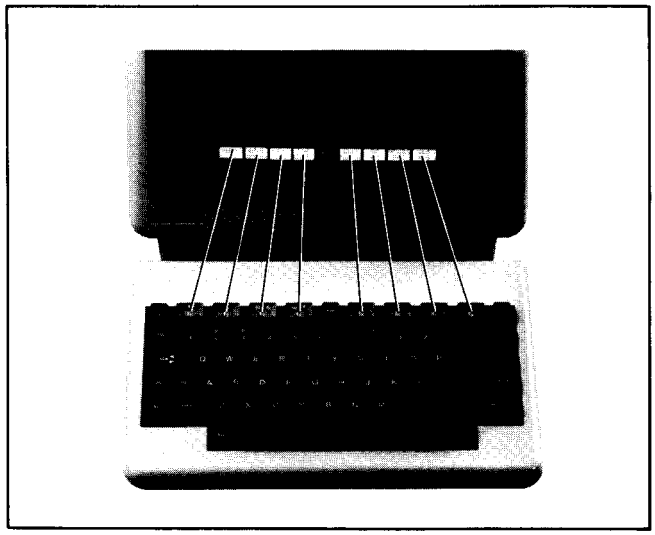


Figure 1-3. Function Keys and Screen Labels

Figure 1-4. Mode Selection Key Labels

	[f1]	[f2]	[f3]	[f4]	[f5]	[f6]	[f7]	[f8]
MODES	LINE MODIFY*	MODIFY ALL *	BLOCK MODE *	REMOTE MODE *	TERMINAL TEST	MEMORY LOCK *	DISPLAY FUNCTNS*	AUTO LF *
AIDS	modes	define user key	CLEAR LINE	CLEAR DISPLAY	aids & features	NEXT PAGE	PREVIOUS PAGE	edit keys
EDITS	LINE MODIFY*	MODIFY ALL *	CLEAR LINE	CLEAR DISPLAY	INSERT LINE	DELETE LINE	INSERT CHAR	DELETE CHAR
AIDS & FEATURES		margins/ tabs/col	service keys		enhance video	define fields		config keys
MARGINS/ TABS/COL	START COLUMN	SET TAB	CLEAR TAB	CLR ALL TABS	LEFT MARGIN	RIGHT MARGIN	CLR ALL MARGINS	
CONFIG			datacomm config		terminal config			
CONFIG	SAVE CONFIG	NEXT CHOICE	PREVIOUS CHOICE	DEFAULT VALUES	POWER ON VALUES		DISPLAY FUNCTNS*	config keys
DEFINE USER KEYS		NEXT CHOICE	PREVIOUS CHOICE	DEFAULT VALUES			DISPLAY FUNCTNS*	
ENHANCE VIDEO	define fields		SET ENHANCMT		INVERSE VIDEO *	BLINK VIDEO *	UNDRLINE VIDEO *	HALF BRIGHT*
DEFINE FIELDS	enhance video	START FIELD		STOP FIELD				FORMAT MODE *
SERVICE KEYS					TERMINAL TEST	IDENTIFY ROMS	DATACOMM TEST	

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

Figure 1-5. HP 2382A Function Key Hierarchy

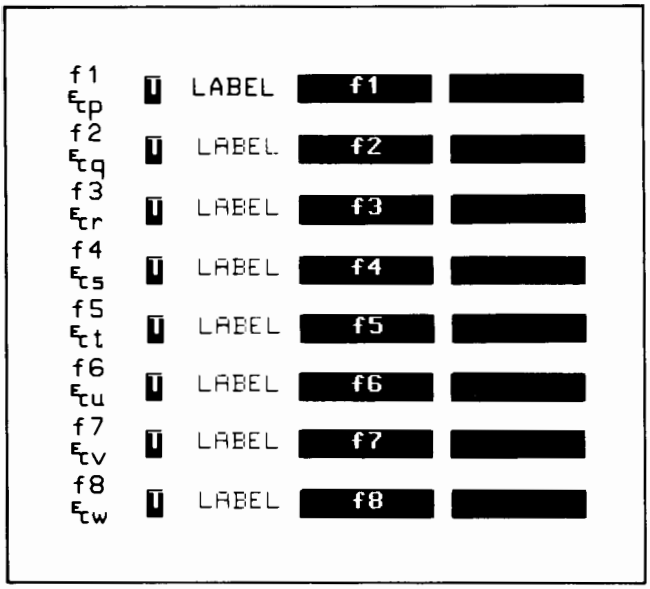


Figure 1-6. User Keys Definition Menu

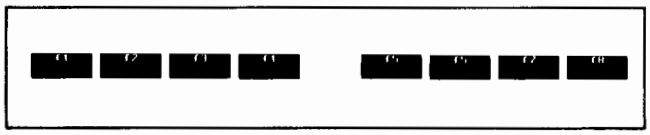


Figure 1-7. Default User Key Labels

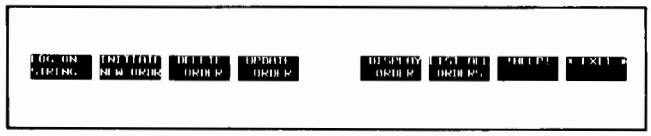


Figure 1-8. Sample User-Supplied User Key Labels

CONFIGURING THE TERMINAL

The HP 2382A is configured at "power-on" according to the setting of the switches at the back panel. After "power-on" the configuration may be modified by making keyboard entries into formatted menus displayed on the screen or by a program executing in a host computer. Refer to Section II for complete information on configuring the terminal.

DATA COMMUNICATIONS

Transmission can be performed in character mode, block line mode, or block page mode; 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)

HP's ASCII eight bits mode is also available.

Refer to Section V for complete information on data communications.

SELF-TEST

The HP 2382A 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. See Section VII for complete information concerning the various self-tests.

CONFIGURING THE TERMINAL

SECTION II

INTRODUCTION

The terminal is designed so the various terminal characteristics can be configured quickly and easily either through switches at the rear panel or by displaying configuration menus on the screen and then using the keyboard to change the content of these menus. When saved in memory, the terminal's configuration characteristics will remain active until either new values are saved, the terminal is hard reset or turned off, or the rear panel switches are changed.

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. "Power on" always initializes the configuration to that of the switches at the rear panel. For configuration menu fields that do not have associated rear panel switches, the terminal will load these fields with their default values at power on and during hard reset.

CONFIGURATION FUNCTION KEYS

To gain access to the configuration menus through the keyboard, press the AIDS key and then the aids & features function key. This causes the following label display across the bottom of the screen.

[ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ]

Pressing config keys changes the label display to

[ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ]

Pressing datacomm config or terminal config causes a configuration "menu" to appear on the screen and redefines the function keys to a set of functions that will assist you in manipulating the various parameters within the menu.

The datacomm menu is described in Section V, Data Communications. The terminal configuration menu is described in this section. In addition, information is provided which will enable you to change the terminal configuration menu from the keyboard or from the power on configuration switches.

TERMINAL CONFIGURATION MENU

When you press the terminal config function key, the menu shown in figure 2-1 appears on the screen.

NOTE

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

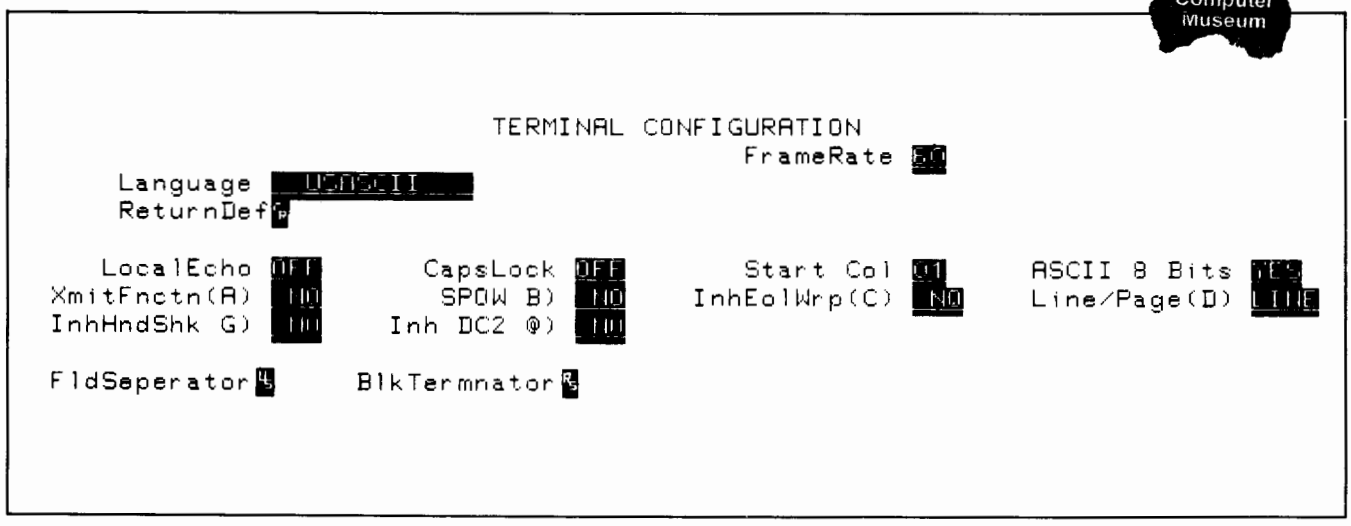


Figure 2-1. Terminal Configuration Menu

The menu contains a set of unprotected fields that you access using the TAB key. Except when the cursor is positioned in the fields labeled "RETURN Def", "FldSeparator", or "BlkTermnator", the alphanumeric keys are disabled and you select the desired parameters using the NEXT CHOICE and PREVIOUS CHOICE function keys.

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

Table 2-1. Terminal Configuration Menu Fields & Power-On Configuration Switches

FrameRate

This field specifies what line frequency (50 or 60Hz) the terminal is designed to operate at. The screen refresh rate is then synchronized to the specified frequency. If this field is set to the wrong value, the images on the screen will pulsate visibly (this will not happen until the SHIFT CONFIG key is pressed).

Values: 50 (for 50 Hz power source)
60(for 60 Hz power source)

The Frame Rate can be temporarily selected using the following escape sequences:

ESC&k1J = 50 Hz
ESC&k0J = 60 Hz

Default: 60 Hz

Switch: B8 (UP = 60 Hz, DOWN = 50 Hz)

Language

As will be described in Appendix C, Character Sets, of this manual, this terminal can be ordered with any of the following keyboards:

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

When the terminal includes any of the optional keyboards, it also automatically includes the appropriate language character set ROM. This field specifies which national keyboard format is to be used in interpreting keystrokes.

Values: USASCII (United States)
SVENSK/SUOMI (Swedish/Finnish)
DANSK/NORSK (Danish/Norwegian)
FRANCAIS azM (French AZERTY layout with mutes)
FRANCAIS qwM (French QWERTY layout with mutes)
FRANCAIS az (French AZERTY layout)
FRANCAIS qw (French QWERTY layout)
DEUTSCH (German)
UK (United Kingdom)
ESPANOL M (Spanish with mutes)
ESPANOL (Spanish)

For the French keyboard layouts, the AZERTY and QWERTY designations refer to the location of the A, Z, Q, and W keys as follows:

```
AZERTY: Row 3 = AZERTY(etc.)
        Row 2 = QSD(etc.)
        Row 1 = WXC(etc.)
QWERTY: Row 3 = QWERTY(etc.)
        Row 2 = ASD(etc.)
        Row 1 = ZXC(etc.)
```

For the French and Spanish keyboard layouts, the mutes designation refers to the manner in which certain accent character keystrokes are handled (^ and " on the French layout and ´ on the Spanish). If the mutes are enabled, those keystrokes will generate the particular accent character but will NOT move the cursor. When you then type another character, that character replaces the accent on the screen (and the ASCII code for the new character replaces the accent code in display memory). If the terminal is in remote mode, the code for the accent character is transmitted to the host computer at the point in the data stream where you pressed the accent key. Mute mode is not supported in block mode.

Default: USASCII

RETURN def

This field specifies the definition of the RETURN key. The default definition is an ASCII CR. The definition may consist of up to two characters. If the second character is a space, it is ignored.

Default: CR

Table 2-1. Terminal Configuration Menu Fields and Power-On Configuration Switches (Continued)

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

Switch: C1 (UP = ON, DOWN = OFF)

Caps Lock

This field specifies whether the terminal generates the full 128-character ASCII set or only Teletype-compatible codes.

ON (Esc&k 1C)
The terminal generates only Teletype-compatible codes: uppercase ASCII (00-5F, hex) and DEL (7F, hex). Unshifted alphabetic keys (a-z) generate the codes for their uppercase equivalents, the { , ³ and } keys generate the codes for [, \ , and] , respectively. The key for generating ~ and ' is disabled.

OFF (Esc&k 0C)
The terminal generates the full 128-character ASCII set of codes.

Default: OFF

Switch: C2 (UP = ON, DOWN = OFF)

Start Col

If the line in which you are entering data is the bottommost used line in display memory (there are no printing or nonprinting 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 a 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 memory using the **start column** 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

ASCII8Bits

When this operating mode is enabled (=YES), the terminal transmits 8-bit ASCII codes in which the eighth (high-order) bit, when set (=1), indicates that the character is from the alternate character set. This is a Hewlett-Packard convention and you will ordinarily use it only when communicating with an HP 300 Computer System or in conjuction with certain HP line printers (such as the HP 2635A/B Printing Terminal).

Values YES = 8-bit codes
NO = Standard 7-bit codes.

Default: NO

Switches: (A5 – UP, A6 – DOWN, and A7 – UP) = YES

NOTE: When ASCII 8 bits is set to "YES" by a keyboard entry in the terminal configuration menu, the parity in the datacomm configuration menu must be set to "NONE".

Table 2-1. Terminal Configuration Menu Fields and Power-On Configuration Switches (Continued)

XmitFnctn(A)

This field specifies whether escape code functions are both executed at the terminal and transmitted to the host computer.

YES (Esc &s 1A)
The escape code sequences generated by control keys such as ROLL UP and ROLL DOWN are transmitted to the host computer. If local echo is ON, the function is also performed locally.

NO (Esc &s 0A)
The escape code sequences for the major function keys are executed locally but NOT transmitted to the host computer.

Note that display functions will transmit Esc Z, Esc Y to a host computer.

Default: NO

Switch: C3 (UP = YES, DOWN = NO)

SPOW(B)

This field specifies whether or not spaces entered through the keyboard will overwrite existing characters.

NO (Esc &s 0B)
Spaces entered through the keyboard will overwrite existing characters.

YES (Esc &s 1B)
Enable SPace OverWrite (SPOW) latch. When the SPOW latch is off, overwriting occurs as normal. When the SPOW latch is on, spaces entered through the keyboard move the cursor forward but do not overwrite existing characters. Once the SPOW latch has been enabled by the above escape sequence it is turned on by a carriage return and is turned off by a line feed, home up, or back tab. It may also be turned on and off programmatically through the use of an Esc &k sequence as follows:

ON: Esc &k 1N

OFF: Esc &k 0N

Default: NO

InhEolWrp(C)

This field specifies whether or not the end-of-line wrap is inhibited.

NO (Esc &s 0C)
When the cursor reaches the right margin it automatically moves to the left margin in the next lower line (a local carriage return and line feed are generated).

YES (Esc &s 1C)
When the cursor reaches the right margin it remains in that screen column until an explicit carriage return or other cursor movement function is performed (succeeding characters overwrite the existing character in that screen column).

Default: NO

Line/Page(D)

This field specified whether or not the terminal, when operating in block mode, will transmit data a line at a time or a page at a time.

Line (Esc &s 0D)
When operating in block mode, the terminal will transmit data a line at a time.

Page (Esc &s 1D)
When operating in block mode, the terminal will transmit data a page at a time.

For a detailed description of the differences between block line and block page mode, refer to "ENTER KEY" in Section V of this manual.

Default: Line

Switch: C4 (UP = PAGE, DOWN = LINE)

InhHndShk(G) and Inh DC2(H)

Together, these fields determine what type of handshaking is to be used when transferring blocks of data from the terminal to the host computer.

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.

Table 2-1. Terminal Configuration Menu Fields and Power-On Configuration Switches (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.
3. Computer sends D_1; terminal responds with D_2; computer responds with another D_1; terminal transmits block of data.

NOTE

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

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

InhHndShk(G) = YES
Eliminates the use of the DC1 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.

If your terminal is connected to a Hewlett-Packard computer system, you will find that the default settings for these fields (both OFF) are usually adequate for your purposes. If you are concerned about the specific type of handshake to be used for one or more of the particular types of block transfers, however, you should use the following summary to verify (or alter) the settings of the InhHndShk(G) and Inh DC2(H) fields:

1. ENTER key in block mode; or

   ENTER or RETURN key in modify mode; or

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

   InhHndShk(G) (ignored)

   Inh DC2(H) = NO ---> $D_1/D_2/D_1$

   Inh DC2(H) = YES ---> no handshake

2. ENTER key in character mode.

   InhHndShk(G) = YES

   Inh DC2(H) = NO ---> $D_1/D_2/D_1$

   Any other combination no handshake

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

   Cursor sense, terminal ID status, primary status, secondary status, display transfer initiated by Etd, or device status request; or device control completion code.

   InhHndShk(G) = NO ---> D_1

   InhHndShk(G) = YES

   Inh DC2(H) = NO ---> $D_1/D_2/D_1$

   InhHndShk(G) = YES

   Inh DC2(H) = YES ---> no handshake

Table 2-1. Terminal Configuration Menu Fields and Power-On Configuration Switches (Continued)

	The G and H selection can be temporarily selected by the following escape sequences: ESC&s0G disables G ESC&s0H disables H ESC&s1G enables G ESC&s1H enables H Defaults: InhHndShk(G) = NO Inh DC2(H) = NO Switches: InhHndShk(G) is switch C5 Inh DC2(H) is switch C6 (UP = YES, DOWN = NO)
FldSeparator	When you press the ENTER key while the terminal is in block page mode and display memory contains a formatted display the terminal automatically transmits the specified field seperator character at the end of each protected field (except the final one). Value: Any ASCII character Default: ETB Switch: C7; DOWN — Field Seperator = ETX — Block Terminator = ETB UP — Field Seperator = ETB — Block Terminator = DC4
BlkTerminator	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: NAK Switch: C7; DOWN — Block Terminator = NAK — Field Seperator = ETB UP — Block Terminator = DC4 — Field Seperator = NAK
ShiftFunctnKeys	This field specifies the operation of the shifted function keys (the light colored keys in the top row of the keyboard). When "NO" is specified in this field the light colored keys act as function keys in the unshifted mode and cursor control keys in the shifted mode. When "YES" is specified the light colored keys act as cursor control keys in the unshifted mode and function keys in the shifted mode. Default: NO Switch: C8 (UP = YES, DOWN = NO) NOTE If your terminal has a terminal configuration menu field labeled "Language" refer to Appendix D for its configuration information.

Note that as you alter the fields of a configuration menu on the screen the selected values do not take effect as you enter them. When you have set all the fields to the desired values, pressing the [f1] function key will store and activate the desired settings.

While the terminal configuration menu is displayed on the screen, the function keys [f1] to [f8] perform as shown in table 2-2.

Table 2-2. Configuration Function Keys

f4	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.)
f5	Pressing this key causes fields on the screen to be filled with the switch specified value. Fields without associated switches are loaded with the default values.
f7	Pressing this key alternately enables and disables display functions mode. When enabled, an asterisk appears in the function key display. You use display functions mode for entering ASCII control characters in the `ReturnDef`, `FldSeperator`, or `BlkTerminator` fields. Note that this implementation of display functions mode is separate from that which is enabled/disabled via the mode selection keys. Enabling or disabling display functions mode using this function key does NOT alter the effect of the mode selection key (and vice versa)
f8	Pressing this key removes the menu from the screen (WITHOUT activating it) and returns the function key labels to the following:

Lock/Unlock Configuration Menus

Using an escape sequence, you can "lock" the current configuration menus (terminal config or datacomm config) so that the menu can not be accessed from the keyboard. Any attempt to access a locked menu from the keyboard will result in a "beep" from the bell. Note that when the configuration menus are locked, the and mode selection keys are also locked.

To lock the menus, use the following escape sequence:

```
Et&q 1L
```

To unlock the menus, use the following escape sequence:

```
Et&q 0L
```

SETTING CONFIGURATION PARAMETERS WITH ESCAPE CODES

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

Parameter Name As Shown in Menu	Type of Escape Sequence Used
Frame Rate	`Et&k`
LocalEcho	
CapsLock	
SPOW	
XmitFnctn(A)	`Et&s`
SPOW(B)	
InhEolWrp(C)	
Line/Page(D)	
InhHndShk(G)	
InhDC2(H)	

The `Et&k` and `Et&s` sequences alter the particular parameter in the menu, and the new setting takes effect immediately, but they are not permanently stored.

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

```
FrameRate = 60:    Et&k 0J
FrameRate = 50:    Et&k 1J

LocalEcho = OFF:   Et&k 0L
LocalEcho = ON:    Et&k 1L

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

SPOW               Et&k0N
                   Et&k1N
```

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

```
Et&k 1l 1C
Et&k 1c 1L
```

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

```
XmitFnctn(A) =  NO:   Et&s0A
XmitFnctn(A) = YES:   Et&s1A

SPOW(B) =  NO:        Et&s 0B
SPOW(B) = YES:        Et&s 1B
```

```
InhEolWrp(C) =  NO:  Et&s0C
InhEolWrp(C) = YES:  Et&s1C

Line/Page(D) = LINE: Et&s 0D
Line/Page(D) = PAGE: Et&s 1D

InhHndShk(G) =  NO:  Et&s0G
InhHndShk(G) = YES:  Et&s1G

Inh DC2(H)   =  NO:  Et&s 0H
Inh DC2(H)   = YES:  Et&s 1H
```

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

```
Et&s 1d 0g 1H
Et&s 0g 1h 1D
Et&s 1h 1d 0G
```

THE HP 2382 POWER ON CONFIGURATION SWITCHES

Power on configuration switches at the rear of the terminal are provided so that you may select the most appropriate configuration of the terminal at power on.

There are three banks of eight switches each for a total of 24 selections (see figure 2-2). The leftmost bank shall be referred to as switches A1 - A8, the middle bank B1 - B8, and the rightmost bank as switches C1 - C8. Switches A1 - A8 and B1 - B4 are used to specify the power on values of the data comm configuration menu. Switches B5, B6, B7 specify whether block mode, remote mode, and auto line feed are enabled at power on. Switches B8 and C1 - C8 specify the power on values of the terminal configuration menu.

At power on the values specified by these switches will be activated. If the switches are toggled after power on, the change in configuration will be activated immediately and the value in the associated menu field will be changed. This is unlike changes in configuration from keyboard entries into a menu which must be saved in order to be activated. Table 2-3 specifies what configuration parameter is associated with each switch. For descriptions of the terminal configuration parameters see table 2-1 and for descriptions of the data communications parameters refer to table 5-3.

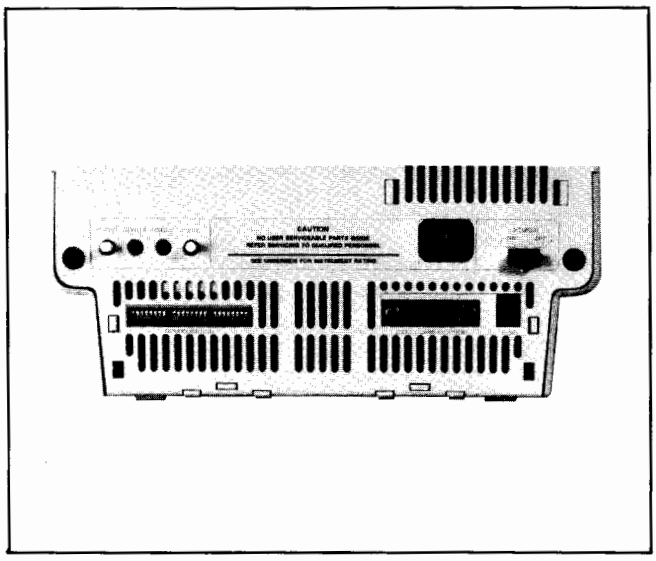


Figure 2-2. Power On Configuration Switches

Table 2-3. Power On Configuration Switch Functions

SWITCH	FUNCTION
C8	ShiftFunctnKeys
C7	BlkTerm/FldSep
C6	Inh DC2 (H)
C5	InhHndShk (G)
C4	Line/Page
C3	Xmit Functn (A)
C2	CapsLock
C1	Local Echo
B8	FrameRate
B7	Auto LF
B6	REMOTE MODE
B5	BLOCK MODE
B4	CS(CB) Xmit
B3	RecvPace
B2	SR(CH)
B1	Chk Parity
A8	EnqAck
A7	Parity/ASCII 8 Bits
A6	"
A5	"
A4	Baud Rate
A3	"
A2	"
A1	"

KEYBOARD CONTROL

SECTION III

INTRODUCTION

The HP 2382A keyboard is a separate unit that is linked to the display portion of the terminal by a telephone type cable. The keyboard layout is similiar to that used for standard office typewriters. It has 68 keys that include eight function keys (the light colored keys in the top row) and one key for specifying the family of functions to be assigned to the function keys (the AIDS key). Most of the remaining keys support the ASCII coded character set.

SELECTING MODES

Pressing the AIDS and then the modes function key enables the mode selection keys and changes the function keys screen labels to the following:

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

Remote/Local Modes

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

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

From the keyboard, you switch the terminal back and forth between local and remote modes using the REMOTE MODE function key. From the keyboard or a user-definable key, you can switch the terminal from local to remote (and vice versa) using the following escape sequences:

Local: Esc&k0R
Remote: Esc&k1R

The power on configuration switch B6 determines whether or not the terminal is in remote mode at power on. When you change modes using the escape sequences, however, the power-on state or state after hard reset is NOT altered. After turning off the power, the terminal reverts to the mode specified by the remote/local designator in switch B6.

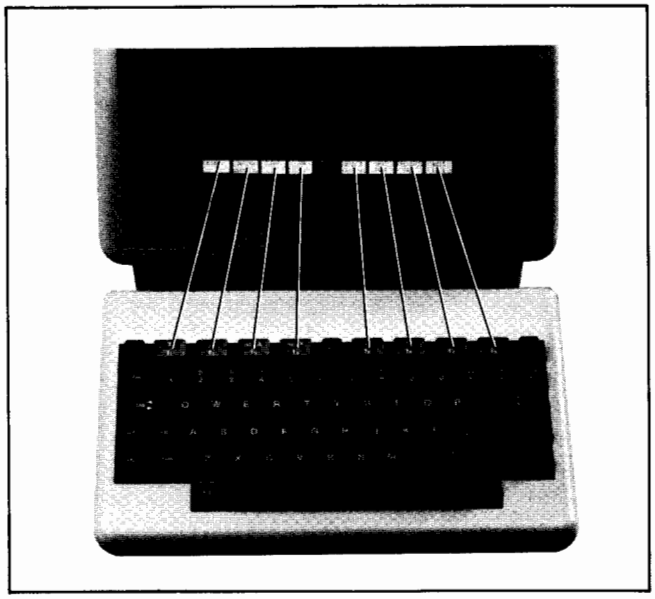


Figure 3-1. Screen-Labeled Function Keys

Character/Block Modes

Computer Museum

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

- Character Mode. In this mode, data is transmitted a character at a time as it is entered through the keyboard. ASCII control codes (such as CR and LF) are transmitted.
- Block Mode. In this mode, data is NOT transmitted at the time it is entered through the keyboard. Instead, you transmit an entire block of data by first typing the data (after initially typing the data you can move the cursor around and edit the data as desired) and then press the ENTER key.

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

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

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

ENABLE: ESC&k1B
DISABLE: ESC&k0B

The power on configuration switch B5 determines whether or not the terminal is in block mode after power on. Hard reset does not effect whether block mode is on or off. When you change modes using the escape sequences the power on or hard reset state is NOT altered.

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

Format Mode

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

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

The designing of forms and the use of format mode are described in Section IV. From the keyboard, you enable and disable format mode using the FORMAT MODE key. From a program executing in a host computer, you enable and disable format mode using the following escape sequences:

ENABLE: ESCW
DISABLE: ESCX

Line Modify Mode

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

Note that while line modify mode results in a block transmission, it is completely independent of the block mode function described earlier in this section (you do NOT have to first enable block mode). In fact, line modify mode is a feature that was specifically designed for use when the terminal is operating in character mode.
From the keyboard, you enable line modify mode using the LINE MODIFY key. Line modify mode is automatically disabled when you press either the 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. Hard reset and power down also disable Line Modify mode.

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

Modify All Mode

When the terminal is in character mode, you can enable modify all mode, which switches the terminal to a special form of block mode. Modify all mode is the same as line modify mode except that it is NOT disabled when you press RETURN or ENTER. From the keyboard, you enable and disable modify all mode using the MODIFY ALL key. From a program executing in a host computer, you enable and disable modify all mode using the following escape sequences:

ENABLE: ESC&k1M
DISABLE: ESC&k0M

Hard reset does not affect the state of modify all mode. At power on modify all mode is always off.

NOTE

When using modify mode you will usually want the data block (NOT a P_2 handshake control code) to be sent when you press ENTER or RETURN. The default configuration parameters, however, enable the P_1/P_2 P_1 handshake. Therefore, in most cases you will first need to disable the $P_1/P_2/P_1$ handshake before using modify mode. You do so by setting the `Inh DC2(H)` field in the terminal configuration menu to "YES", or by sending ESC&s 1H. The modify line and modify all modes are ignored if the terminal is in format or block mode.

Auto Line Feed Mode

When auto line feed mode is enabled, an ASCII line feed control code is automatically appended to each ASCII carriage return control code generated through the keyboard. That is, every ↵ code generated through the keyboard becomes a ↵LF.
ASCII carriage return control codes can be generated through the keyboard in any of the following ways:

- By pressing the RETURN key, provided that a ↵ 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 ↵ 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 ↵ code is transmitted as the line terminator).

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

ENABLE: ESC&k1A
DISABLE: ESC&k0A

After turning off the power, the terminal reverts to the mode specified by AUTO LF power on configuration switch (B7). Hard reset does not effect the state of auto line feed.

Memory Lock Mode

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

OVERFLOW PROTECT. If you home the cursor and then enable memory lock mode, display memory becomes "protected" so that no data can be lost off the top. In such a case, when you have used all available lines in display memory, any attempt to use more memory is rejected with an audible "beep". You may, however, use the cursor control keys to go back and alter any of the existing data. To continue entering new data, merely disable memory lock mode and reposition the cursor immediately below the last line.

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

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

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

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

Normal editing can be performed within the locked rows; that is, the rows are locked by row number only, so if lines are inserted among the locked rows, they become locked but the total number of locked rows does not increase.

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

ENABLE: ESCl
DISABLE: ESCm

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

Display Functions Mode

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

- In local mode, it displays ASCII control codes and escape sequences but does not execute them. For example, if you press the ← key the terminal displays ESCD on the screen but does not perform the "cursor left" function.

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

There are two exceptions to the foregoing descriptions:

1. Pressing the DISPLAY FUNCTIONS function key which enables/disables display functions mode, is executed but not displayed.

2. A CR (or CR LF if auto line feed mode is enabled) is executed in addition to being transmitted and/or displayed.

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

ENABLE: ESC Y
DISABLE: ESC Z

NOTE

There is interaction between display functions and the XmitFnctn(A) field of the configuration menu. If XmitFnctn(A) is on, the DISPLAY FUNCTIONS key transmits ESC Y, ESC Z. Note that, as a result of this, the receiver of the transmission can never exit display functions.

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

Caps Mode

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

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

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

ENABLE: ESC&k1P
DISABLE: ESC&k0P

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

Caps Lock Mode

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

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

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

ENABLE: ESC&k1C
DISABLE: ESC&k0C

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, or through the Caps Lock switch at the back panel the state you save will remain active until you explicitly change it from the keyboard, change it by an escape sequence, or turn off the terminal. When you enable or disable the mode using the escape sequence you only change the active value of the Caps Lock field in the terminal configuration menu.

At power on the terminal reverts to the mode specified by the Caps Lock power on configuration switch (C2).

USER-DEFINABLE KEYS

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

1. You can assign to each key a string of ASCII alphanumeric characters and/or control codes (such as CR or LF).

2. You can specify each key's operational attribute: whether its content is to be executed locally at the terminal, transmitted to a host computer, or both.

3. You can assign to each key an alphanumeric label (up to 16 characters) which, in user keys mode, is displayed across the bottom of the screen.

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

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

Defining Keys Locally

To define one or more keys from the keyboard, first press the AIDS key and then the DEFINE USER KEY key, or use ESC j. The user keys menu shown in figure 3-2 then appears on the screen. Note that the menu in figure 3-2 contains the default values for all of the fields. While the menu is displayed on the screen, you can reset the entire menu to the default values by pressing the DEFAULT VALUES function key.

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 in the same manner as the alphanumeric keys (N). If the terminal is in local mode, the content of the key is executed locally. If the terminal is in remote mode and local echo is disabled (OFF), the content of the key is transmitted to the host computer. If the terminal is in remote mode and local echo is enable (ON), the content of the key is both transmitted to the host computer and executed locally.

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

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 (eighty 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.

When entering the label and key definition you may access display functions mode by way of the DISPLAY FUNCTNS key. Note that this implementation of display functions mode is separate from that which is enabled/disabled via the mode selection keys.

The RETURN can be used for including 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.

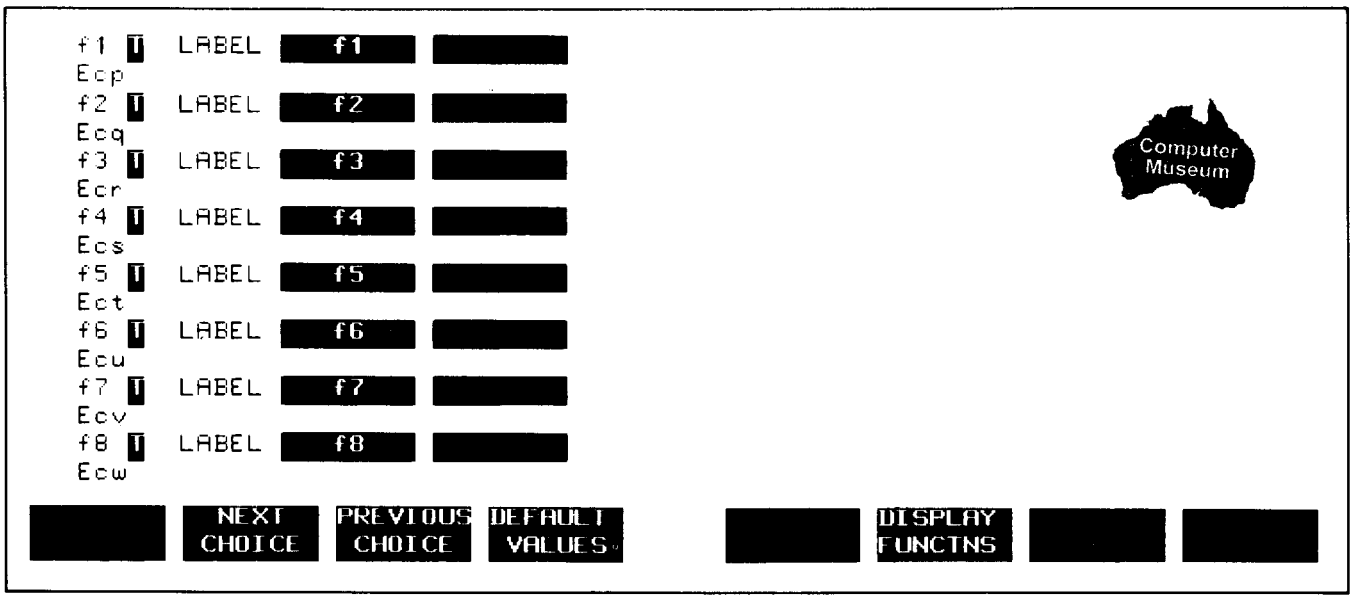


Figure 3-2. User Keys Definition Menu

When you are finished defining all the desired keys, press the key and the user keys menu will disappear from the screen. When you press and simultaneously the defined user key labels are displayed across the bottom of the screen and the through 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:

ESC&f <attribute><key><label length>
<string length><label><string>

where:

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

The <attribute>, <key>, <label length>, and <string length> parameters may appear in any sequence but must precede the label and key definition strings. You must use an uppercase identifier (A, K, D, or L) for the final parameter and a lowercase identifier (a, k, d, or l) for all preceding parameters. Following the parameters, the first 0 through 16 characters, as designated by <label length>, constitute the key's label and the next 0 through 80 characters, as designated by <string length>, constitute the key's definition string. The total number of displayable characters (alphanumeric data, ASCII control codes such as ESC R and ESC F, and explicit escape sequence characters) in the label string must not exceed 16, and in the definition string must not exceed 80.

Example: Assign LOG-ON as the label and HELLO USER.ACCOUNT as the definition for the user key. The key is to have the attribute "N".

```
ESC&f5k6d19LLOG-ONHELLO USER.ACCOUNT^F
```

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

```
f5 N LABEL LOG ON
HELLO USER.ACCOUNT<CR>
```

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

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

Controlling The User Keys Menu Programmatically

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

DISPLAY MENU: ESC j
REMOVE MENU: ESC k

Controlling The Function Key Labels Programmatically

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

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

The escape sequences are as follows:

ESC&j Disable the function keys entirely and remove all key labels from the screen. Note: If a user key through is hit while the the terminal is in remote the function key is transmitted (whether or not the function key labels are displayed on the screen).

ESC&jA Enable the mode selection keys.

ESC&jB Enable the user keys.

`Esc&j` Unlock screen labels.

`Esc&jS` Lock screen labels.

ENTER KEY

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

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

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

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

Table 3-1. ENTER Key Operation

CHARACTER MODE

The cursor is repositioned to column 1.

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

ASCII control codes, video enhancement escape sequences, alternate character set escape sequences, and field definition escape sequences all are transmitted if encountered. If the operation is terminated by encountering the end of the line, the terminal sends a CR (or a CR LF) if auto line feed mode is enabled). The cursor is repositioned to column 1 and a line feed is performed if auto line feed mode is enabled.

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

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

The type of handshaking used is determined as follows:

```
InhHndShk(G) = YES
Inh DC2(H) = NO ---> ACK1/ACK2/ACK3
Any other combination ---> no handshake
```

CHARACTER MODE, FORMAT MODE

If the cursor is within an unprotected field, all characters from the current cursor position through the end of the field are transmitted to the host computer as a block. Otherwise, the terminal searches for the next subsequent unprotected field and transmits the content of that field.

ASCII control codes within the field are transmitted.

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

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

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

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

The type of handshaking used is determined as follows:

```
InhHndShk(G) = YES
Inh DC2(H) = NO ---> ACK1 ACK2 ACK3
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 the terminal menu is set to "line".

`Inh DC2(H) = YES`

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

`Inh DC2(H) = NO`

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

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

Table 3-1. ENTER Key Operation (Continued)

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

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

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

The type of handshaking used is determined as follows:

```
InhHndShk(G) is ignored.
Inh DC2(H) = NO ---> P1/P2/P3
```

BLOCK LINE MODE, FORMAT MODE

Block line mode means that block Mode is on and the line/page (D) field of the terminal configuration is set to "line". ASCII control codes within the field are transmitted.

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

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

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

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

The type of handshaking used is determined as follows:

```
InhHndShk(G) (ignored)
Inh DC2(H) = NO ---> P1/P2/P3
Inh DC2(H) = YES ---> no handshake
```

BLOCK PAGE MODE

Block page mode means that the terminal is in Block Mode and the `line/page` field in the terminal configuration is set to "page".

```
Inh DC2(H) = YES
```

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

```
Inh DC2(H) = NO
```

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

ASCII control codes, 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 ↵LF. If the operation is terminated by encountering the end of display memory, the terminal sends a ↵LF followed by a block terminator after the last line. If the operation is terminated by encountering a block terminator, the terminal sends only a block terminator after the last line.

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

The type of handshaking used is determined as follows:

```
InhHndShk(G) (ignored)
Inh DC2(H) = NO ---> P1P2P3
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.

Table 3-1. ENTER Key Operation (Continued)

ASCII control codes within the fields are transmitted.

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

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

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

The type of handshaking used is determined as follows:

```
InhHndShk(G)  (ignored)
Inh DC2(H) = NO ^1/^3/^1
Inh DC2(H) = YES ---> no handshake
```

MODIFY MODE

Note that modify line and modify all modes are functional only when the terminal is configured for character mode operation. When either block mode or format mode are 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 and Power On Configuration Switches in Section II.

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

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

If the operation is terminated by encountering the end of the line, the terminal sends a ^M (or a ^M^J 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 ^M (or a ^M^J if auto line feed mode is enabled). The cursor remains positioned immediately following the terminator.

The type of handshaking used is determined as follows:

```
InhHndShk(G) is ignored.
Inh DC2(H) = NO  ---> ^1/^3/^1
Inh DC2(H) = YES --->no handshake
```

Send Display (Esc d)

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:

```
Esc d
```

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

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

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

```
InhHndShk(G)    = NO  ---> ^1

InhHndShk(G)    = YES ---> ^1/^3/^1
Inh DC2(H)      = NO  ---> ^1/^3/^1

InhHndShk(G)    = YES ---> no handshake
Inh DC2(H)      = YES ---> no handshake
```

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

Note that an Esc d sequence resets the "block trigger received" flag. This means, for example, that if you are using the ^1 handshake and the terminal receives a ^1 followed by the Esc d, it "forgets" that a block trigger was just received and thus will NOT send the data immediately. The terminal must receive another ^1 before it will start the data transfer.

Computer Museum

The amount of data transferred depends on the following terminal settings: page/line field in terminal configuration, and whether Block Mode, Modify All or Line Modify is enabled. For more detailed information, refer to Table 3-1 " ENTER Key Operation" in this section.

ENABLE/DISABLE KEYBOARD

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

SHIFT RESET

CLEAR BREAK

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

ENABLE: ESC b
DISABLE: ESC c

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

SOFT RESET

A soft reset does the following:

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

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

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

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

ESC 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 and configuration parameters saved in memory are used rather than the power on switch configuration.

More specifically, a hard reset does the following:

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

   Disables display functions mode, caps mode 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.

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:

ESC E

BREAK

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

BELL

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

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

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

WAIT

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

Esc●

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

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

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

🔔 Esc● Esc● 🔔

MODEM DISCONNECT

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

DISPLAY CONTROL

SECTION IV

INTRODUCTION

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

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

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

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

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

CURSOR CONTROL

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

The light colored keys in the top row of the keyboard, in addition to to their use as function keys, are used as the cursor control keys. To achieve this dual purpose these keys are used in conjunction with the SHIFT key.

The `ShiftFunctnKeys` field of the terminal configuration menu determines which of these purposes is served in the "shifted" mode. When the `ShiftFunctnKeys` field is saved with a value of "NO", the keys f1 through f8 act as cursor control keys in the "shifted" mode and as function keys in the "unshifted" mode. When the `ShiftFunctnKeys` field is saved with "YES" the opposite is true; in the "shifted" mode the keys act as function keys and in the "unshifted" mode they act as cursor control keys.

See Table 2-1 for further discussion.

Computer Museum

Home Up

Pressing the ↖ key moves the cursor to the left margin in the top row of the screen and rolls the text in display memory down as far as possible so that the first line in display memory appears in the top row of the screen. When format mode is enabled, the ↖ key also rolls the text down as far as possible but leaves the cursor positioned at the beginning of the first unprotected field. If no fields have been defined, the cursor will appear at the first column of the first row on the screen.

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

ESC H

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

NOTE

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

Home Down

Pressing the key 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:

```
ESCF
```

Move Cursor Up

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:

```
ESCA
```

Move Cursor Down

Each time you press the key the cursor moves downward one row in the current column position. If you hold the key down, the cursor movement continues row-by-row until the key is released. When the cursor is in the bottom row of the screen, pressing this key moves the cursor to the same column position in the top row of the screen.

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

```
ESCB
```

Move Cursor Right

The key, moves the cursor 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:

```
ESCC
```

Move Cursor Left

The key, moves the cursor 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:

```
ESCD
```

Roll Text Up

The key, rolls the text in display memory up one row on the screen. The top row rolls off the screen, the remaining data rolls up one line on the screen, and a new line of data rolls from display memory into the bottom line of the screen. If you hold this key down, the text continues to roll upward until you release the key or until the final line of data in display memory appears in the top row of the screen. In the latter case, pressing or continuing to hold down the key has no further effect. The "roll up" function is illustrated in figure 4-1a.

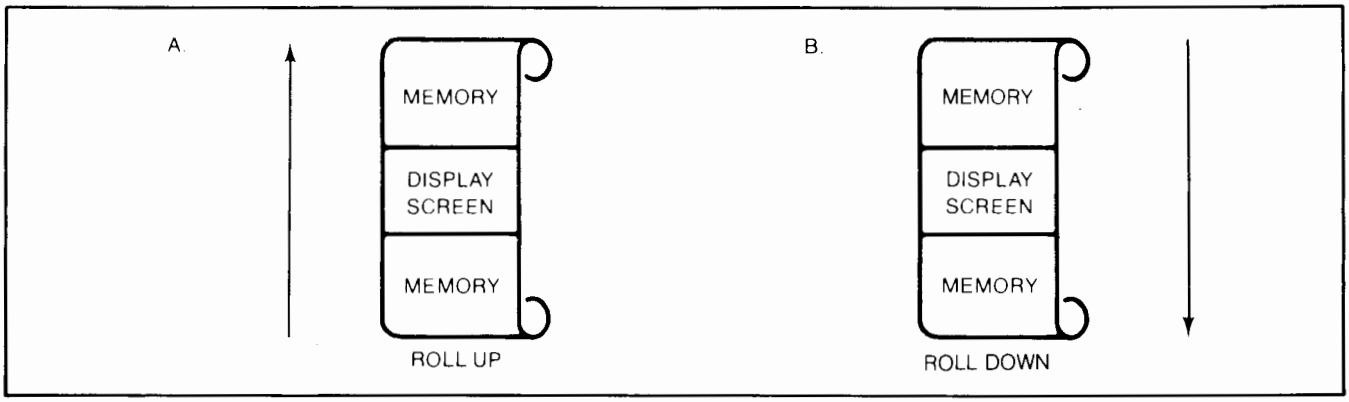


Figure 4-1. The "Roll" Data Functions

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

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

ESC t S

Roll Text Down

The ROLL ▼ key, rolls the text in display memory down one row on the screen. The bottom row rolls off the screen, the remaining data rolls down one line on the screen, and a new line of data rolls from display memory into the top line of the screen. If you hold this key down, the text continues to roll downward until you release the key or until the first line of data in display memory appears in the top row of the screen. In the latter case, pressing or continuing to hold down the key has no further effect. The "roll down" function is illustrated in figure 4-1b.
In the configuration and user softkey definition menus, this key is disabled. In memory lock mode, the unlocked text rolls behind the locked text, as if the bottom line of the locked text is at the bottom of the screen. The cursor position relative to the screen remains unchanged after this operation.

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

ESC t T

Next Page/Previous Page

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

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

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

When a configuration or user softkey definition menu is displayed this feature is disabled.

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

ESC t U

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

When a configuration or user softkey definition menu is displayed this feature is disabled.

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

ESC t 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.

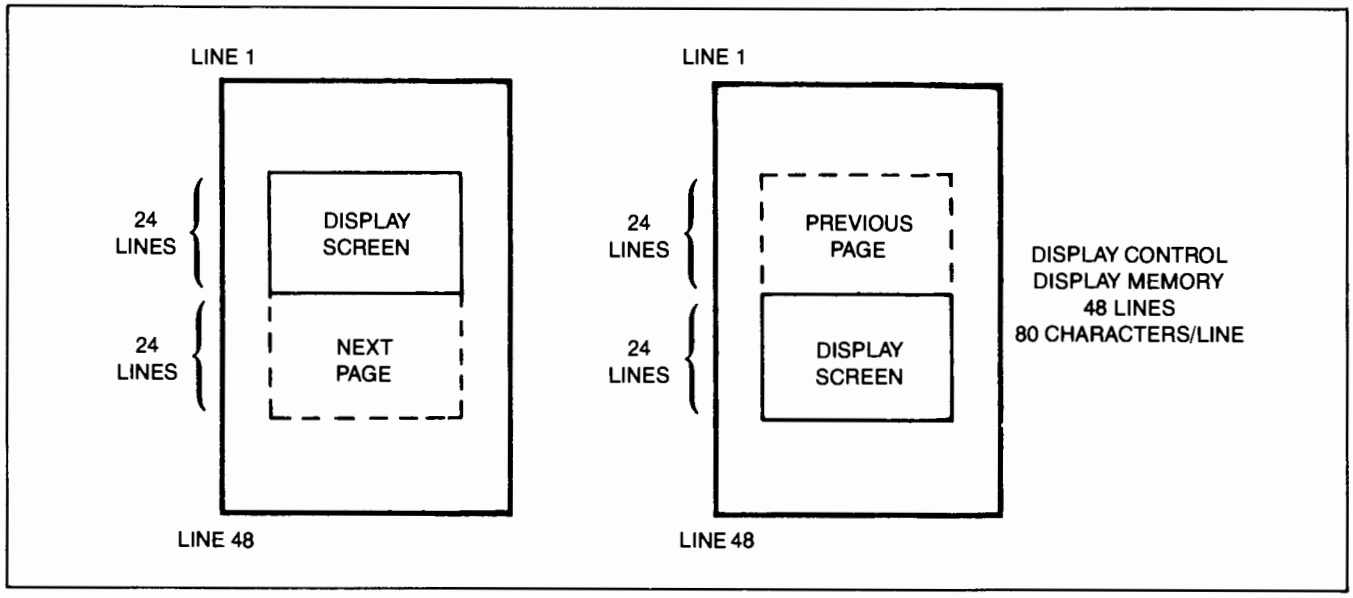


Figure 4-2. Previous Page and Next Page Concepts

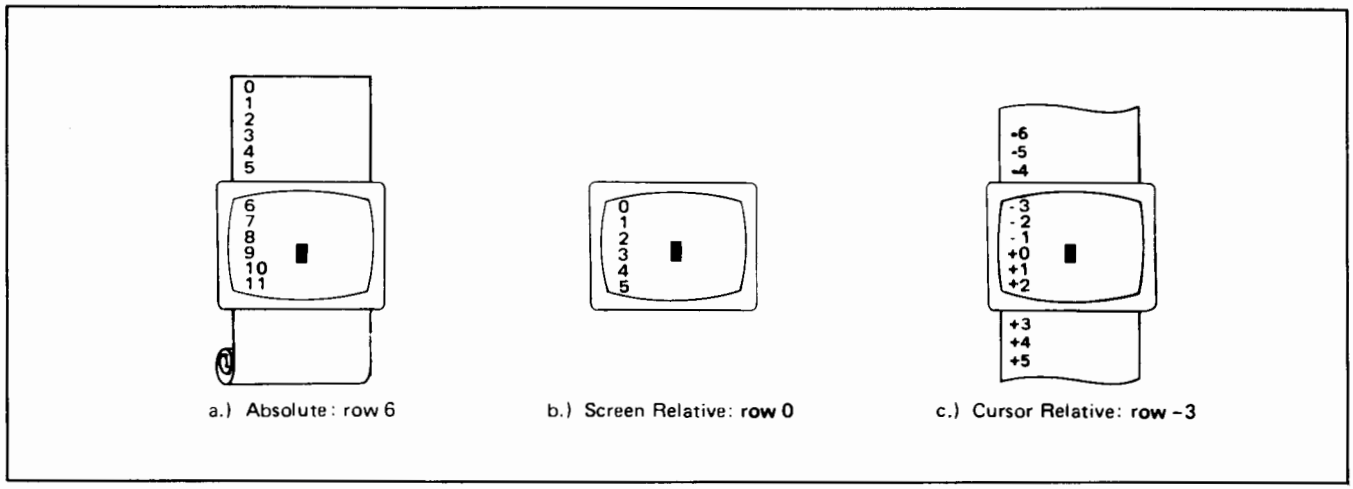


Figure 4-3. Row Addressing

MEMORY ADDRESSING SCHEME

Display memory positions can be addressed using absolute or relative coordinate values. Display memory is made up of 80 columns (0-79) and 48 lines (0-47). There can be as many as 48 lines of 80 characters (2 screens). The amount of memory in the terminal can be determined from byte 0 of the primary terminal status (refer to section VI). The types of addressing available are:

- Absolute
- Screen Relative
- Cursor Relative

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

a. Absolute: row 6

b. Screen Relative: row 0

c. Cursor relative: row −3

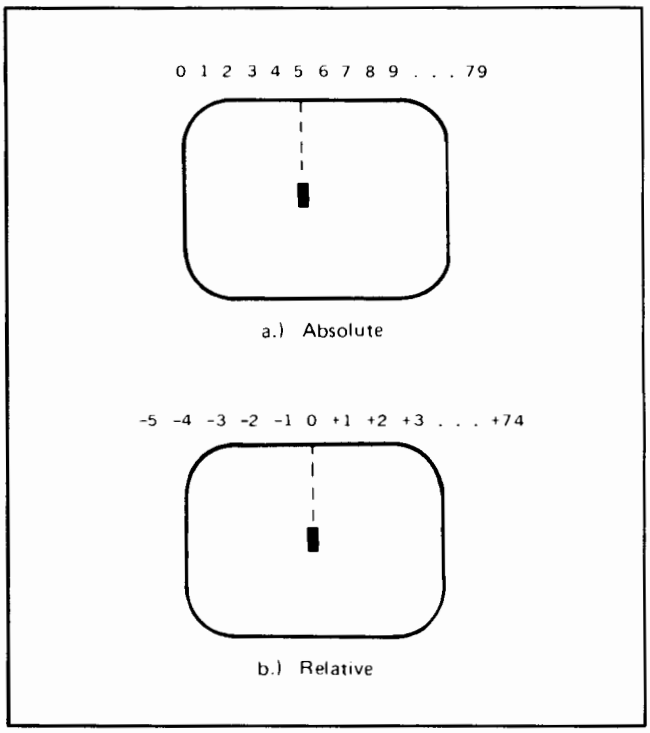


Figure 4-4. Column Addressing.

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

Whenever the row or column addresses exceed those available, the largest possible value is substituted. In screen relative addressing, the cursor can not be moved to a row position that is not currently displayed. For example, in figure 4-3c a relative row address −10 would cause the cursor to be positioned at the top of the current screen (relative row −3). Column positions are limited to available screen positions (0 to 79 in figure 4-4a and −5 to 74 in figure 4-4b). The cursor can not be wrapped around from column 0 to column 79 by specifying large negative values for relative column positions.

CURSOR SENSING

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

Absolute Sensing: Eca

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

computer: Eca
terminal: Eca020c040R

Relative Sensing: Ec'

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

computer: Ec'
terminal: Ec&a020c005Y

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 operations.

Screen Relative Addressing

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

```
Ec&a <column number> c <row number> Y

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

Ec&a <column number> C

Ec&a <row number> Y
```

where:

`<column number>` — is a decimal number specifying the screen column to which you wish to move the cursor. Zero specifies the leftmost column.

`<row number>` — is a decimal number specifying the screen row (0-23) to which you wish to move the cursor. Zero specifies the top row of the screen; 23 specifies the bottom row.

When using the above escape sequences, the data on the screen always remains unchanged. If you specify only a `<column number>`, the cursor remains in the current row. Similarly, if you specify only a `<row number>`, the cursor remains in the current column.

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

```
Ec&a6y19C
```

Absolute Addressing

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

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

where:

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

`<row number>` — is a decimal number specifying the row coordinate (within display memory) of the character at which you want the cursor positioned. Zero specifies the first (top) row in display memory, 47 specifies the last.

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

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

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

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

```
Et&a26r59C
```

Cursor Relative Addressing

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

```
Et&a +/- <column number> c
     +/- <row number> R
Et&a +/- <1 row number> r
     +/- <column number> C
Et&a +/- <column number> C
Et&a +/- <row number> R
```

where:

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

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

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

- If a specified character position lies within the boundaries of the screen, the cursor moves to that position and the data on the screen remains unchanged.
- If the specified cursor relative row precedes the top line currently visible on the screen, the cursor moves to the specified column in the top row of the screen and the text rolls downward until the specified row appears in the top line of the screen.

- If the specified cursor relative row follows the bottom line currently visible on the screen, the cursor moves to the specified column in the bottom row of the screen and the text rolls upward until the specified row appears in the bottom line of the screen.

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

Example: The following escape sequence moves the cursor (and rolls the text if necessary) so that it is positioned at the character residing 15 columns to the right and 25 rows above the current cursor position within display memory:

```
Ec&a+15c-25R
```

Combining Absolute and Relative Addressing

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

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

```
Ec&a69c+18R
```

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

```
Ec&a-15c3Y
```

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

```
Ec&a9c47R
```

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 function keys or programmatically by using escape sequences:

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

Insert Line

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

From the keyboard, each time you press the INSERT LINE key under the edit keys level of function keys 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 function is disabled in format mode, and is not accessible in the configuration and user softkeys definition menus.

NOTE

When display memory (48 rows) is full, inserting a line will cause data to be lost. The first data row will always be the one to be released unless it happens to be on the screen, in which case the last data row 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:

```
EcL
```

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 function key under the edit keys level of function keys 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 function is disabled in format mode, and is not accessible in the configuration and user softkeys definition menus.

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

ESC M

Insert Character

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

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

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

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

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

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

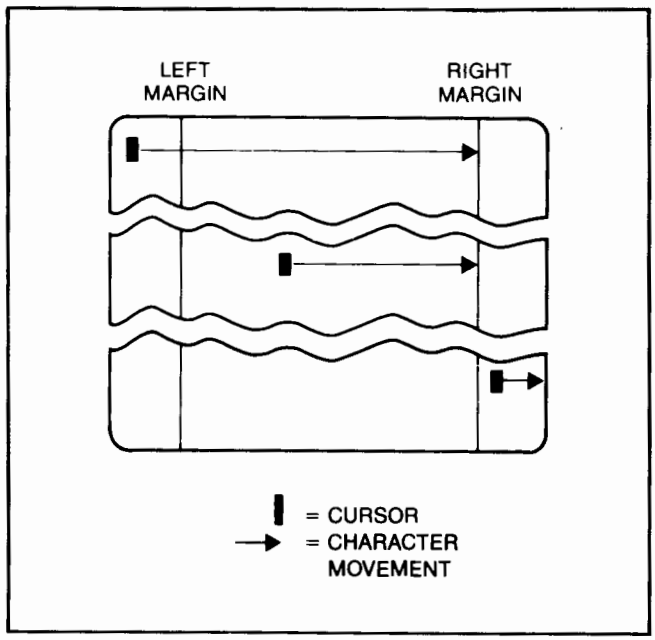


Figure 4-5. Character Insert with Margins

From the keyboard, you enable and disable the insert character editing function by pressing the INSERT CHAR function key under the [redacted] level of function keys. When enabled, the characters, "IC" are displayed in the status line at the bottom of the screen and an asterisk appears in the INSERT CHAR screen label.

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

ENABLE: ESC Q
DISABLE: ESC R

Delete Character

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

This edit function is meant to be used within that portion of the screen delineated by the left and right margins. If you position the cursor to the left of the left margin, the delete character function works as described above. If you position the cursor beyond the right margin, however, the delete character function affects those characters from the current cursor position through the right boundary of the screen. The movement of existing characters during a "delete character" editing operation is illustrated in figure 4-6.

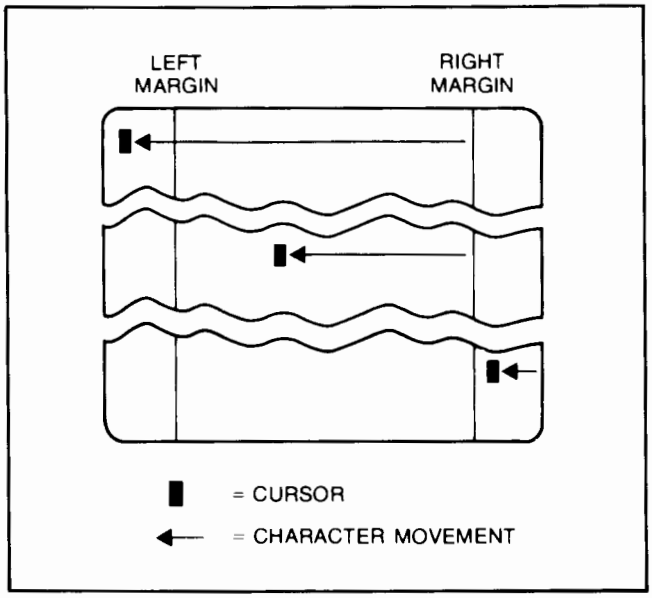


Figure 4-6. Character Delete with Margins

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

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

In the user softkeys definition menu, the DELETE CHAR function key is not accessible, however, the delete character escape code can be used. When the delete character escape sequence is used in the softkey definition menu it works the same as in format mode. The delete character feature is disabled in the configuration menus. From the keyboard, each time you press the DELETE CHAR function 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

When format mode is off, pressing the CLEAR DISPLAY function key deletes 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 function 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 feature is disabled in the user softkeys definition and configuration menus.

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

$^{E}_{c}J$

Clear Line

When format mode is off, pressing the CLEAR LINE function 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 function 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 function key has no effect.

In the user softkeys definition menu and configuration menus the CLEAR LINE function key is not accessible, however the escape code may be used. When the clear line escape sequence is used in the softkey definition menu it acts as in format mode. The clear line feature is disabled in the configuration menus. 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 are performed without regard to margins. Format mode, when enabled, clears the margins, creating an 80 character line (1-80).

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

When data is being received from a host computer, it enters display memory only within the defined margins. When the cursor reaches the right margin, it automatically moves to the left margin in the next lower line (as mentioned above, this operating characteristic can be disabled through the use of the "InhEolWrp" configuration parameter). When an ASCII ␍ control code is received, the cursor always moves to the left margin in the current line regardless of whether or not auto line feed mode is enabled.

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

This changes the function key labels to the following:

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

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

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

SET LEFT MARGIN: ␍t4

SET RIGHT MARGIN: ␍t5

CLEAR ALL MARGINS: ␍t9

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

SETTING AND CLEARING TABS

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

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

This changes the function key labels to the following:

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

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

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

SET TAB: ␍t1

CLEAR TAB: ␍t2

CLEAR ALL TABS: ␍t3

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

TAB

From the keyboard, you can move the cursor ahead to the next subsequent tab stop using the TAB key. In format mode, tab moves the cursor to the beginning of the next unprotected field. The last field wraps around the beginning of the first field. Tab acts similarly in the user softkeys definition menu and the configuration menu.

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

Esc I

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

Note that the left margin is treated as a tab stop. When the cursor is positioned at or to the right of the rightmost tab stop, the tab function moves the cursor to the left margin in the next lower line. When the cursor is positioned to the left of the left margin, however, the tab function advances the cursor to the first explicit tab stop in the line (or to the left margin in the next lower line if no explicit tab stops are defined). Note that tabbing the cursor to the next line is the equivalent of a linefeed.

BACK TAB

From the keyboard you can move the cursor backward to the previous tab stop using the SHIFT and TAB keys.

In format mode, configuration menus, and user keys definition menu, the cursor, if within a field, will move to the beginning of the field; otherwise it will move to the first character of the previous unprotected field.

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

Esc i

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

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

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

DISPLAY ENHANCEMENTS

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

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

You use these enhancements on a field basis. They may be used separately or in any combination. When used, they cause control bits to be set within display memory. If the content of display memory is subsequently transmitted in block mode to a host computer, these control bits are translated into escape sequences which are transmitted along with the displayable text characters. From the keyboard, you enable and disable the various video enhancements using the enhance video set of function keys. To get to that set, use the following keystroke sequence:

This changes the function key labels to the following:

To cause a particular string of text characters to be displayed using one or more of the enhancements, perform the following:

1. Enable the desired enhancement(s) by pressing the associated function key (f5, f6, f7, and/or f8). When an enhancement is enabled, an asterisk is displayed in the label.
2. Position the cursor at the first character in the string.
3. Press f3. The selected enhancements take effect immediately. You will notice the enhancements begin at the cursor position and continue through the end of the line (or through the next subsequent column in which another display enhancement begins). You will also notice that when you press SET ENHANCMT f3, the asterisk(s) disappear from the screen labels (all enhancements are disabled until you once again explicitly enable them).
4. Position the cursor immediately to the right of the final character in the string

5. Press f3 . The enhancements disappear from the cursor position through the end of the line (or through the next subsequent column in which another display enhancement begins).

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

```
Ec&d <enhancement code>
```

where the enhancement code is an @ or one of the uppercase letters A through O specifying the desired enhancements as follows:

Enhancement Character

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

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

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

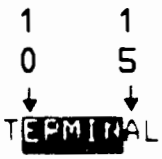
Step 1. Position the cursor in column 10 of line 5.

Step 2. Enter Ec&dC.

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

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

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



DESIGNING AND USING FORMS

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

When format mode is enabled, the cursor automatically moves to the start of the first unprotected field in the form. Henceforth the terminal can only enter data into those portions of the display screen which lie within unprotected fields. When the operator enters a character into the last position of a field, the cursor advances to the start of the next unprotected field. In addition the TAB key can be used to advance the cursor to the beginning of the next unprotected field and the SHIFT and TAB keys can be used to move the cursor to the beginning of the previous 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.

There are three major steps in designing data entry forms:

1. Create the linear structure of the form on the screen using the line drawing set (if your terminal has Option 202).

2. Define the various unprotected fields within the form.

3. Programmatically read the completed form and store it in the host computer for future use.

DRAWING FORMS

In order to draw lines with your terminal, you need option 202 of the HP 2382A terminal. Option 202 adds the Line Drawing character set so that you may create "paper-like" images on the screen.

With the terminal configured to its default state, the standard USASCII set is defined as the Base set, and the Line Drawing set is defined as the alternate character set. The elements of the Line Drawing set and their relationship to the terminal's keyboard are illustrated in figure 4-7.

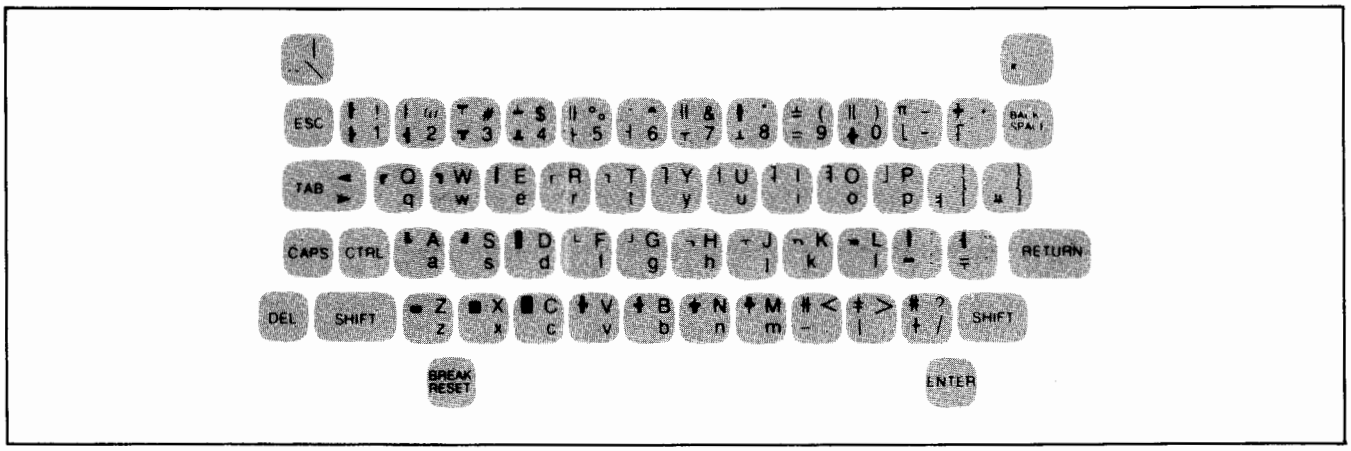


Figure 4-7. Line Drawing Set Elements

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

When you are designing a form through the keyboard, you use ASCII $^{5}_{6}$ (control-N) and $^{5}_{1}$ (control-O) codes to switch back and forth between the Line Drawing and standard USASCII character sets.

You switch from the base set (USASCII) to the Line Drawing set by issuing an ASCII $^{5}_{6}$ code (control-N) and you switch from the Line Drawing set back to the base set by issuing an ASCII $^{5}_{1}$ code (control-O). Note that the $^{5}_{6}$ code affects only those characters from the cursor position through the next $^{5}_{6}$ or $^{5}_{1}$ code or through the end of the line, whichever occurs first. Consequently if the Line Drawing set is enabled at the end of one line on the screen and you also want it enabled at the start of the next line, you will have to explicitly issue another $^{5}_{6}$ at the start of the second line.

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

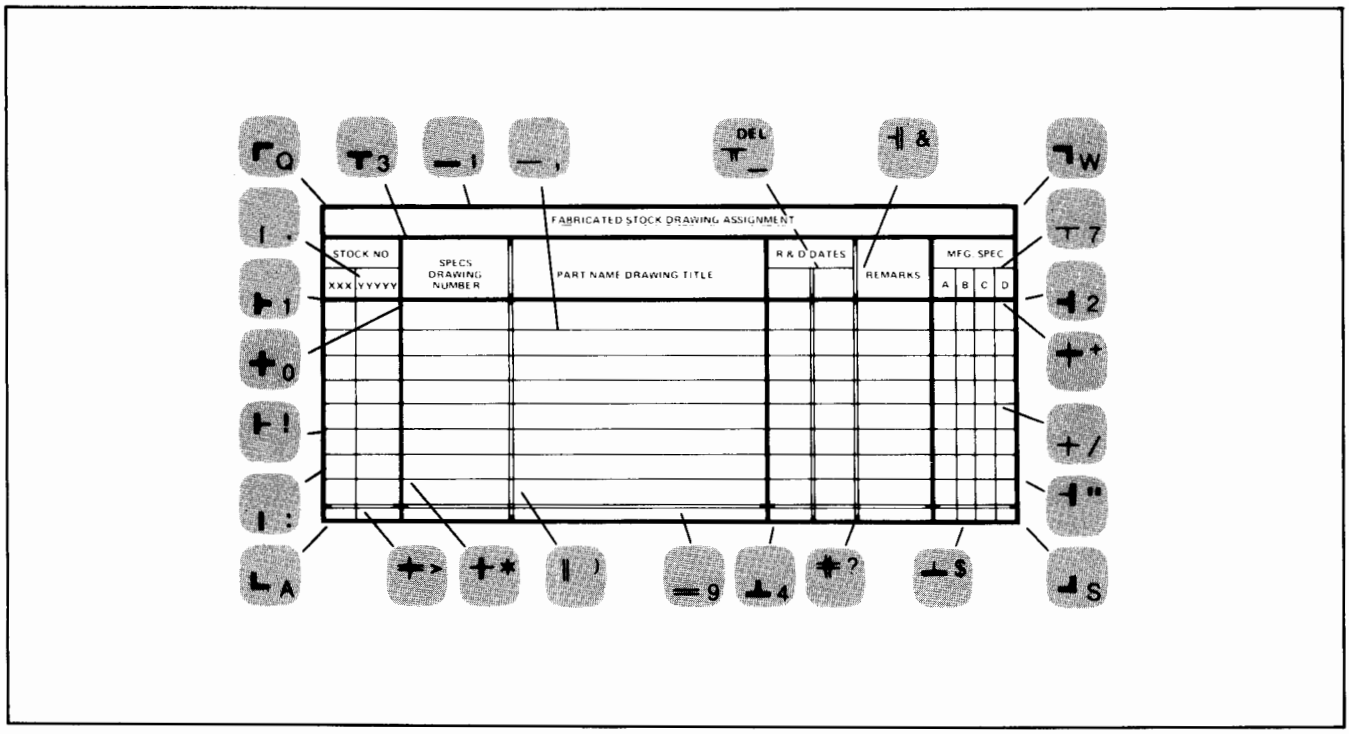
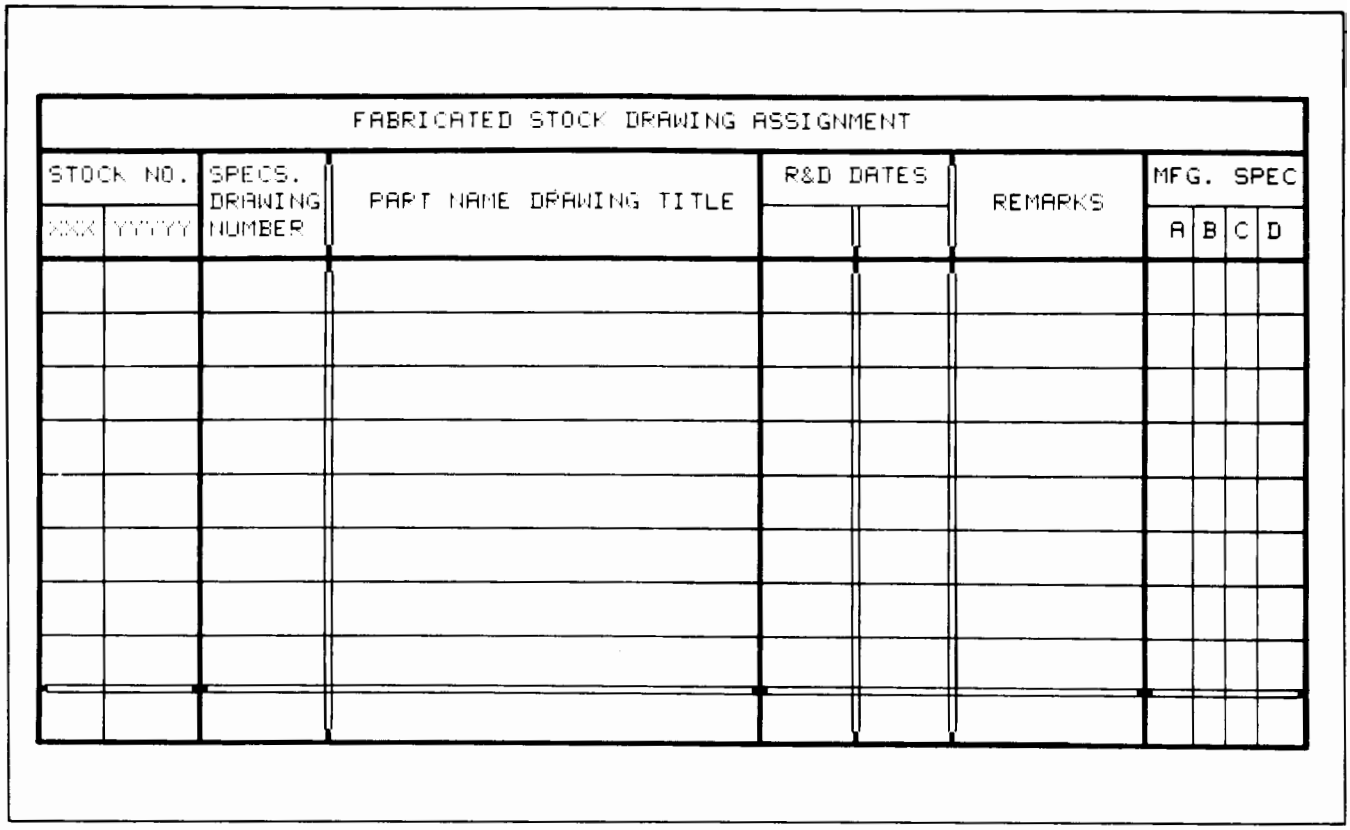


Figure 4-8. Sample Data Entry Form



FABRICATED STOCK DRAWING ASSIGNMENT

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

Figure 4-9. Completed Data Entry Form

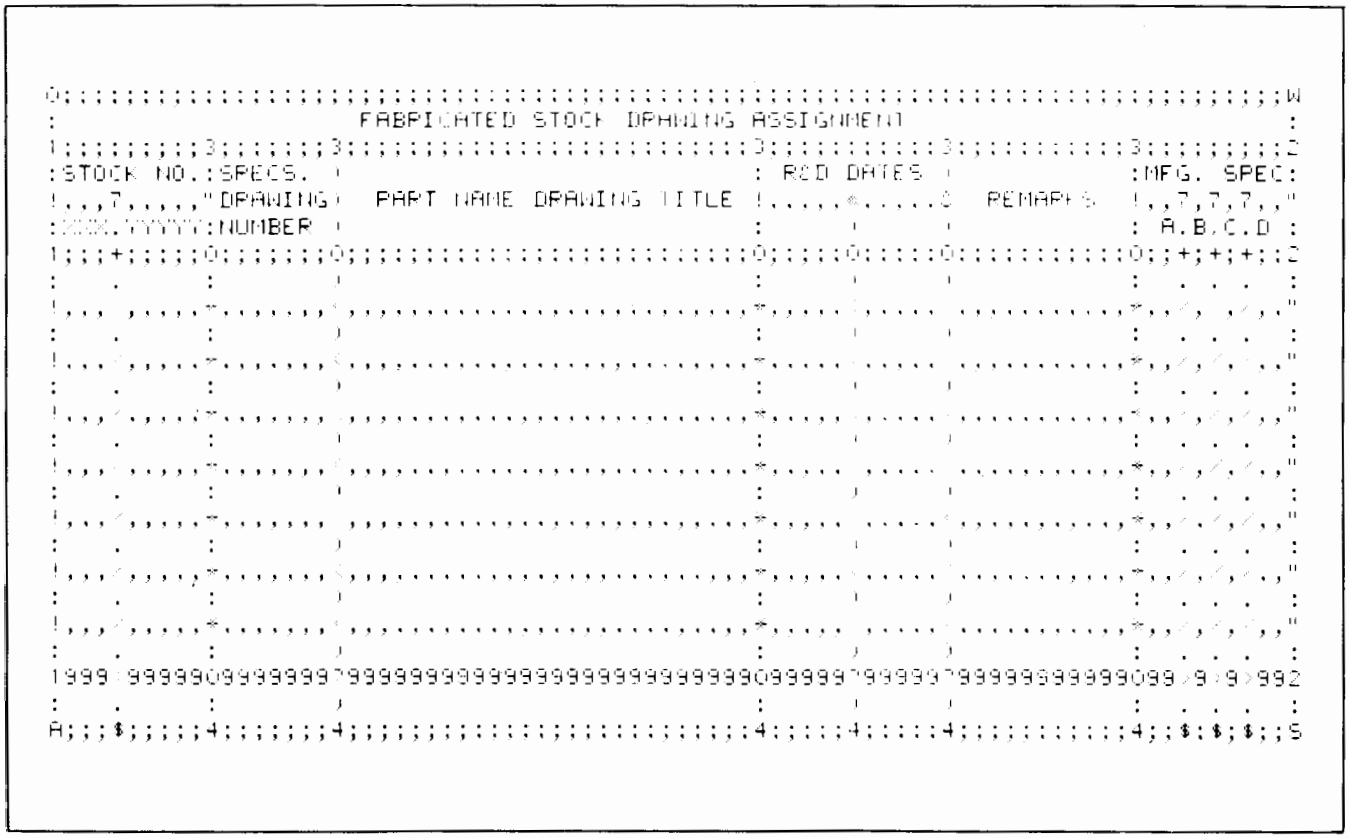


Figure 4-10. Base Set Equivalents for Data Entry Form Illustrated in Figure 4-9.

One approach to generating a form structure through the keyboard is to load two of the user keys with the ^N and ^O codes (control-N and control-O, respectively, with display functions mode enabled), define both keys as Local keys, define their labels as "Line Draw" and "Base Set", respectively, and enable them by pressing SHIFT and A|DS. Then draw the form structure and alphanumeric annotations using the Base set as illustrated in figure 4-10. As the form is evolving use the cursor control keys to switch the linear structure portions of the form to the Line Drawing character set. When doing this, however, be sure that those portions of the form that will be used for data fields are set to the Base set (figure 4-11 shows three lines of the sample form and the various point at which you would use the ^N and ^O codes). You may also, if you wish, load some of the more repetitive line definition (such as the second and third lines in figure 4-11) into user keys to speed up the process of drawing the main body of the form.

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

The program must first define the Line Drawing character set by sending an ESC)B to the terminal.

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

FORMS MODE (FORMAT MODE)

In Forms Mode, the terminal prevents you from overwriting or transmitting data in protected fields. Forms mode is normally entered under control of the computer and is normally used in conjunction with Block Mode operation.

To enable or disable format mode programmatically, use the following escape sequences:

ENABLE: ESC W
DISABLE: ESC X

Format Mode can also be enabled from the keyboard at the `define fields` level of the function keys. To enable/disable Format Mode in this manner perform the following keystrokes:

The following "define fields" level of function keys should now be displayed:

The FORMAT MODE function key (f8) can be toggled on and off to enable and disable Format Mode. Note that an asterisk appears in the FORMAT MODE function key label when format mode is enabled. When Format Mode is turned on the cursor is homed to the beginning of the first unprotected field.

Protected Fields

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

Unprotected Fields

Data can be written into unprotected fields in the normal manner. After reaching the end of an unprotected field, the cursor moves to the beginnng of the next unprotected field. The function can be used to move from one unprotected field to the beginning of the next unprotected field.

From a host computer "unprotected" fields can be defined using the following escape sequences:

Start Unprotected Field: ESC [
End Unprotected Field: ESC]

Computer Museum

Unprotected fields are also terminated at the end of every line.

Unprotected fields can also be defined from the keyboard using the define fields level of function keys.

To use these keys, first position the cursor to where you want the unprotected field to begin, then press the START FIELD function key (f2).

Then position the cursor to the column you want the field to end at and press the STOP FIELD function key (f4).

The enhance video function key (f1) of the "define fields" function key level gives you access to the video enhancements function keys, so you can define unprotected fields with video enhancements.

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

FORM#1876R

VENDOR NAME	ADDRESS	CITY	ZIP
Pacific Tools	45 Dun St	San Jose	95123

Voucher Date	Quantity	Item	COST
5/12/81	3	Hammers	$36.00
5/27/81	5	Pliers	$25.00

NOTE

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

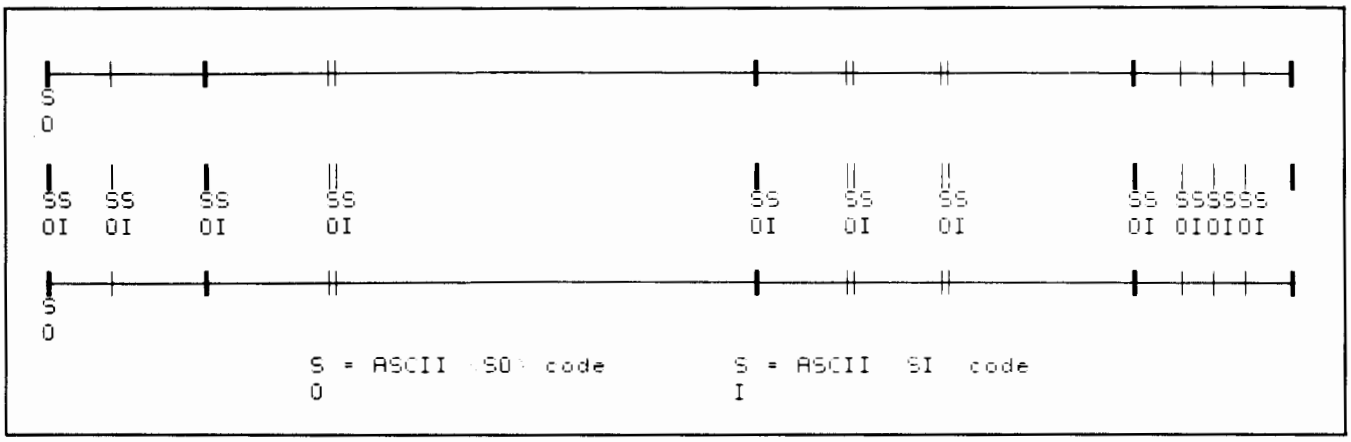


Figure 4-11. Use of Shift-In and Shift-Out Codes

DATA COMMUNICATIONS

SECTION V

INTRODUCTION

The term "data communications" (or "datacomm") refers to the transfer of data between the terminal and a host computer.

There are several ways to connect the terminal to a computer. To arrive at a particular way you must compare a number of factors and make a series of decisions. After selecting the necessary equipment and cables, you must then physically connect the terminal to the computer (or to the modem, if that is what you have chosen) and configure the terminal for use with the particular type of data communications link.

This section is divided into four parts:

1. The first is a general discussion that should help you decide what type of equipment and cabling you need for the data link.

2. The second tells you how to physically install the terminal.

3. The third tells you how to configure the terminal to operate properly with the selected type of data link.

4. The final part provides programming reference material for someone who is writing a datacomm driver or controller program to communicate with the terminal in a point-to-point environment.

Before proceeding with the decision making process, it may help to briefly define the most important terminology as it pertains to data communications.

Data Link: The means by which a terminal is connected to a host computer. This always includes some type of communications line (a coaxial cable, the public telephone network, or a leased telephone line), and it may also include a pair of modems (one at each end of the line).

Point-to-Point: A data communications configuration in which a single terminal is connected to a host computer over a data link. The terminal is designed for use with a point-to-point data link.

Asynchronous: A mode of transmission in which each data character is framed by a "start bit" and one or more "stop bits". The interval between successive data characters is random. The terminal is designed for use with an asynchronous data link.

Full Duplex: A data link in which data can be transmitted in both directions simultaneously.

Character Mode: When the terminal is operating in character mode, it sends data characters to the computer one at a time as they are typed into the keyboard.

Block Mode: When the terminal is operating in block mode, data characters typed into the keyboard are merely stored in display memory. When a block transfer is subsequently triggered (by the host computer or by pressing the ENTER key, a group of data characters is sent from the terminal to the computer as a block.

A point-to-point configuration is the standard form of data communications within the industry (it is sometimes referred to as a "Teletype-compatible" data link). Point-to-point is supported by most computers. At any given time, it accomodates only one terminal per data link; it may, however, operate in either character mode or block mode.

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

Another consideration is the desired availability of the particular computer port. If you wish to have it available (at different times) to terminals in diverse and/or varying locations, then you should choose a modem connection with dial-up capability.

HARDWIRED CONNECTIONS

If you decide on a point-to-point hardwired connection, the only decision that remains to be made is the type of cable to be used. The available cables are summarized in table 5-1. Please note that a hardwired connection for your terminal is always full duplex (the terminal does not support half-duplex connections).

MODEM CONNECTIONS

If you decide on a point-to-point modem connection, you must now decide what type of modem to use. Note that point-to-point as supported by the terminal always employs asynchronous transmission. You will therefore be limiting your choice of modem to the asynchronous variety. Refer to table 5-2 for help in selecting the proper modem.

Table 5-1. Data Communications Cables

CABLE NO.	HP PART NO.	DESCRIPTION
13242M	13242-60002	EUROPEAN MODEM CABLE Male RS-232-C 25-pin connector for interfacing the terminal to the European telephone system via Bell 103 or 202C type European modems. Length: 5 meters (16.7 feet)
13242N	13242-60001	U.S. MODEM CABLE Male RS-232-C 25-pin connector for interfacing the terminal to an HP 1000, 2000, or 3000 Multiplexer; to a Bell 103A, 202C/D/S/T, 212A, or VADIC 3400 modem; or to an acoustic coupler (signal compatible only). Length: 5 meters (16.7 feet)
13242Y	13242-60005	EMP PROTECT (MALE) Male RS-232-C 25-pin connector for interfacing the terminal to an HP 1000, 2000, or 3000 Multiplexer. Provides protection from lightning-induced transients. For use in hardwired configurations only. Length: 5 meters (16.7 feet)
13232U	5061-2403	Modem bypass cable with a female RS-232-C 25-pin connector on both ends. It crosses the signals so that two terminals can communicate with one another. Length: 1.5 meters (5 feet)

Table 5-2. Modems

MODEM	DATA RATE (BITS/SEC)	DUPLEX FULL/HALF	DIALED/ LEASED	REVERSE CHANNEL
Bell 103A	300	F/H	D/L	No
Bell 202T	1200	F/H	L	Option
Bell 202D	(See note 2)			
Vadic VA3400 (see notes 1, 3)	1200	F	D	No

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

Computer Museum

Modem Considerations

If you are communicating with the host computer through a modem, it may be necessary for you to turn on a modem power switch or make modem parity setting changes. The modem's baud rate and parity settings should be the same as those configured in the terminal.

The terminal supports the Bell 103A, 202A, or equivalent type of modem.

Whenever the modem line (Data Set Ready) is active, an asterisk appears between the fourth and fifth screen label at the bottom of the screen. If your facility requires the display of this "active modem" indicator, do not shut off the screen labels display.

The asterisk between the fourth and fifth screen label is controlled by an LED which tracks the Data Set Ready (DSR) input to the terminal. When a modem is connected the DSR line is low (active) and the modem indicator (asterisk) is on. When the DSR line is high, this signals a modem disconnect and the asterisk disappears from the screen.

INSTALLING A POINT-TO-POINT CONFIGURATION

The terminal datacomm port may be connected to a computer via a 25-pin, female RS-232-C compatible connector provided on the back of the terminal (see figure 5-1).

Terminal Cabling

The HP 13242 cables listed in table 5-1 have male 25-pin RS-232-C connectors on both ends (except the 13232U which has two female RS-232-C connectors). You attach a male RS-232-C cable connector to the connector on the rear panel of the terminal. The other RS-232-C end attaches to the modem, computer or multiplexer panel.

CONFIGURING THE TERMINAL

Once the the physical connections between the terminal and the computer or modem are complete, the terminal can be configured. The power on configuration switches at the rear panel of the terminal should be set for the most appropriate configuration to meet your particular needs. Switches A1-A8 and B1-B4 pertain to your terminals data communications power on configuration (see figure 5-2). The values you set in these switches will be the values found in the associated fields in the terminal's Datacomm Configuration menu at power on.

These switches work on an ON/OFF basis. When the switch is pointing down towards the table, it is in the OFF position. When the switch is pointing up it is in the ON position. Table 5-3 describes how to set these switches for your needs.

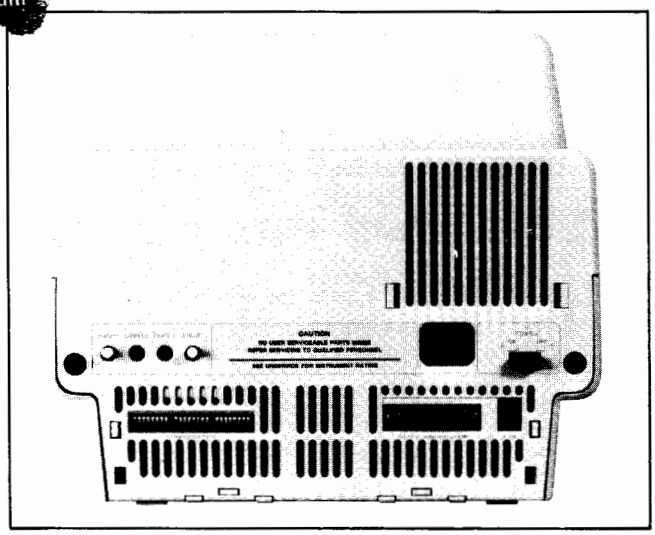


Figure 5-1. HP 2382A Display Terminal, Rear View

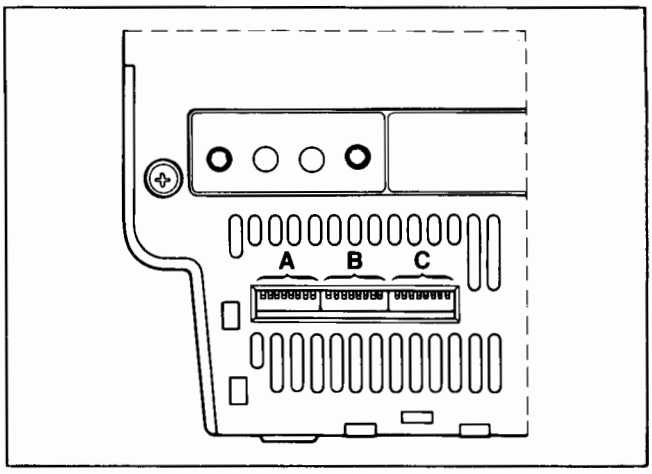


Figure 5-2. Terminal Power On Configuration Switches

The only terminal DataComm Configuration menu field without an associated power on configuration switch is the `XmitPace` field. This field will always power on with a value of "NONE" meaning that `XmitPace` is disabled.

The values set in the switches remain valid after the terminals power has been turned off and is turned on again. The values in these switches will probably remain constant over time.

The terminal's data communications parameters can also be configured through the keyboard by saving the desired entries in the terminal's Datacomm Configuration menu. At power on the menu will reflect the settings of the power on configuration switches. Subsequent changes to the menu from the keyboard will be lost when the terminal is turned off. To access the Datacomm Configuration menu press use the following keystroke sequence:

aids + aids & features

This changes the function key labels to the following:

To access the configuration keys press the config keys key. This changes the function key labels to the following:

Now press the datacomm config key to display the currently active values in the Datacomm Configuration menu.

The function key labels now read:

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 and PREVIOUS CHOICE function keys.

The meanings of the various fields in datacomm menu are described in table 5-3.

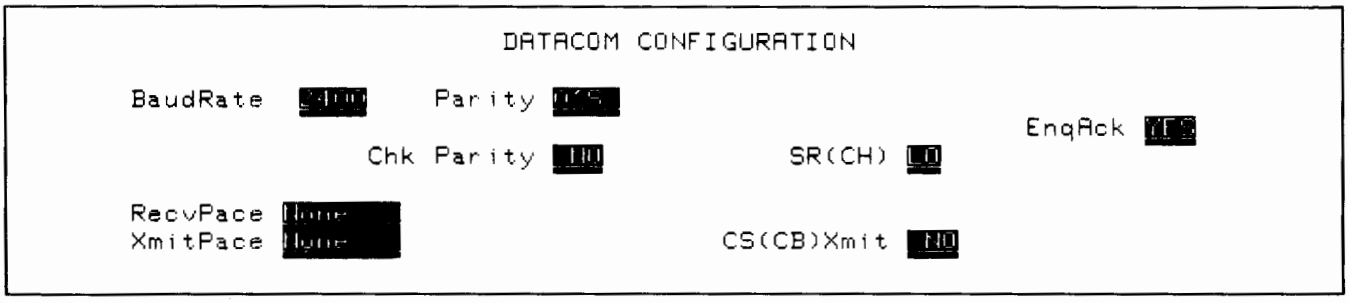


Figure 5-3. Datacomm Configuration Menu

Table 5-3. Datacomm Configuration Menu Fields & Power On Configuration Switches

BaudRate This field specifies at what speed you want data transmission to take place (in bits per second).

Values: 110, 134.5, 150, 300, 600, 1200, 1800, 2400 (default), 4800, 9600

Switches A1 - A4: (0=down/off, 1=up/on)

Switch A1	A2	A3	A4	Baud Rate
0	0	0	0	110
0	0	0	1	134.5
0	0	1	0	150
0	0	1	1	300
0	1	0	0	600
0	1	0	1	1200
0	1	1	0	1800
0	1	1	1	2400
1	0	0	0	4800
1	0	0	1	9600

Other switch combinations will not change (or affect) the Baud Rate and at power on the default baud rate of 2400 baud will be active. If all four switches are up when the terminal is powered on, the terminal will automatically enter Manufacturing Test mode.

Parity This field specifies what type of parity generation and checking you wish used with each data character.

Values: NONE (no parity bit)
0'S (parity bit always zero)(default)
ODD (odd parity)
1'S (parity bit always one)
EVEN (even parity)

Switches: A5, A6, and A7 (0=down/off, 1=up/on)

Switch A5	A6	A7	Parity	
0	0	0	0's	(7 bit)
0	0	1	ODD	(7 bit)
0	1	0	1's	(7 bit)
0	1	1	EVEN	(7 bit)
1	0	0	NONE	(7 bit)
1	0	1	NONE	ASCII 8 Bits enabled

Other switch combinations will not change or affect the parity and at power on the default (0's) parity will be active.

Note: The parity power on configuration switches also effect the power on configuration of the ASCII 8 Bits mode of the Terminal Configuration Menu. To have ASCII 8 Bits mode enabled at power on set the switches as follows: A5-up, A6-down and A7 up.

Table 5-3. Datacomm Configuration Menu Fields & Power-On Configuration Switches (Continued)

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)

Switch: A8 (UP=YES, DOWN=NO)

Chk Parity This field is used for enabling or disabling the parity check feature for data characters received over the datacomm line. Note that if the Parity field (above) is set to NONE, then this field is ignored.

Values: YES (enable)
NO (disable) (default)

Switch: B1: (UP=YES, DOWN=NO)

SR(CH) This field specifies the desired state of the RS-232-C SR line when the terminal's power is first turned on or when the terminal is reset. The SR line, RS-232-C pin number 23, is defined as the Data Signal Rate Detector (DTE Source). It is normally used on dual speed modems to select the appropriate speed (single speed modems merely ignore this line).

Values: HI
LO (default)

Switch: B2 (UP=HI, DOWN=LO)

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 means of performing receive pacing: by using the XON and XOFF control codes.

If this field is set to XonXoff, the terminal will automatically perform receive pacing using XON (ASCII D_1) and XOFF (ASCII D_3) control codes. With this type of receive pacing, the terminal causes the remote device to halt transmission by sending an XOFF code and to resume transmission by sending an XON code. For this type of receive pacing to work, the remote device must of course be configured to start and stop transmission in response to XON and XOFF codes.

Note that if the remote device recognizes XON and XOFF codes and your terminal is operating in character mode, you can issue the codes through the keyboard regardless of the setting of this field. The CTRL and Q keys (when pressed simultaneously) generate an XON code and the CTRL and S keys generate XOFF.

Values: NONE (default)
XON/XOFF

Switch: B3 (UP=NONE, DOWN=Xon/Xoff)

CS(CB)Xmit This field specifies whether or not a true state (−12V) on the RS-232-C Clear to Send (CS/CB) control line is a required condition for transmitting data. For a modem configuration, it is recommended that you set this field to "YES".

Values: YES
NO (default)

Switch: B4 (UP=YES, DOWN=NO)

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/XOFF control codes. When the terminal receives an XOFF code (ASCII D_3), it stops transmitting data. When the terminal subsequently receives an XON code (ASCII D_1) it resumes transmitting data. This should not be used with D_1/P_2 handshaking. Therefore, if Xmit pacing is enabled, InhHndShk(G) and InhD2(H) in terminal configuration must both be set to "ON", which disables D_1/P_2 handshaking.

If this field is set to "NONE", the terminal does not recognize the ASCII D_1 and D_3 codes as XON and XOFF.

For another form of transmit pacing, refer to the description of the CS(CB)Xmit field above.

Values: NONE (default)
XON/XOFF

Switch: There is no power on configuration switch associated with this field. At power on this field is always set to the default (NONE).

When you have set all the switches to the desired values, every time you power on the terminal these values will be active and displayed in the terminal Datacomm Configuration menu. You may change the settings in the menu by using the NEXT CHOICE or PREVIOUS CHOICE keys and then pressing the SAVE CONFIG to store and activate the change in configuration. This change in configuration will be lost however when the terminal is turned off.

While the datacomm configuration menu is displayed on the screen, the DEFAULT VALUES , POWER ON VALUES and config keys keys have the effects described below:

DEFAULT VALUES Pressing this key causes all fields in the menu to be filled with their default values. These default values are listed in table 5-3.

POWER ON VALUES Pressing this key causes all fields with associated power on configuration switches to be set to the value specified by the switches. The XmitPace field has no associated switch and is set to its default value (NONE).

config keys Pressing this key removes the menu from the screen (WITHOUT activating it or saving it in memory) and returns the function key labels to the following:

POINT-TO-POINT PROGRAMMING INFORMATION

This topic discusses programming information of interest to someone who is writing a data communications driver or controller program to communicate with the terminal in an asynchronous point-to-point environment.

An asynchronous point-to-point data communications environment is characterized by a flow of characters that have been produced over random time intervals. In order to achieve hardware synchronization, each character is delimited by a "start bit" and one or more "stop bits".

Start And Stop Bits

These hardware-generated bits are used for synchronizing the transmit and receive devices in an asynchronous environment. A start bit is a "zero" line state (+12V) that lasts for 1.0 bit time; it is affixed to the beginning of a serial character bit stream (which may also include a parity bit). A stop bit is a mark or a "one" line state (−12V) that lasts for 1.0 bit time; it is appended to the end of each serial character bit stream. After the stop bit, the line remains in the mark state until the next character, signified by a start bit, is transmitted.

The start and stop bits are not configurable. For 110 baud, the terminal is automatically configured to transmit 2 stop bits 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 the data and 1 stop bit is expected to be received with data.

Parity Checking

In an asynchronous point-to-point environment, the terminal provides a vertical redundancy check (VRC), which is a character-based error checking mechanism for non-binary data. With VRC, an additional bit is affixed to each character to provide an expected high-order bit state for each character. This type of parity generation and checking is a means of determining the validity of data transfer on a character-by-character basis.

Note that when 8-bit data is being exchanged, parity cannot be used and the `Parity` field in the datacomm configuration menu must be set to "NONE".

The 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.

Receive Buffer

The terminal's receive buffer is a first in/first out (FIFO) storage area for accepting data from the remote device. When you are using any type of receive pacing, this buffer is partitioned into a working buffer and a 40-byte overrun area. For example, the specified buffer size is always 256 bytes, thus if receive pacing is being used, the working buffer is 216 bytes long and the overrun area is 40 bytes long. When the data being received exceeds the working buffer and intrudes on the overrun area, the terminal will exercise its receive pacing mechanism (send an XOFF, for example, if XON/XOFF receive pacing is enabled) at that time to temporarily halt the flow of data from the remote device. When enough data has been processed so that the receive buffer is only half full, the

terminal then signals the remote device to resume transmission (by sending an XON, for example, if XON/XOFF receive pacing is enabled).

There is no equivalent overrun area for transmitting data from the terminal to the remote device.

Receive Errors

When receiving data from the remote device, the terminal can detect the following three types of error conditions (in addition to parity errors):

1. Character overruns — a character is received before the preceding character was processed by the terminal's datacomm firmware.

2. Framing errors — no stop bit was detected at the end of a character.

3. Buffer overflows — the entire allocated buffer space is filled (both the working buffer and the overrun area).

Receive errors, when detected, are reported to the remote device by way of byte 5 of the primary terminal status bytes. The remote device will not be able to determine which type of error occurred. If multiple receive errors occur simultaneously, only one will be reported.

When a datacomm error occurs a delete character (■) is placed in the datacomm queue and later it is displayed on the screen.

Local/Remote Modes

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

Full-Duplex Operation

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

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

Request to Send (RS/CA)
Clear to Send (CS/CB)
Data Terminal Ready (CD/TR)

Pacing Mechanisms

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

1. Hardware Handshake. The computer can temporarily restrain the terminal from transmitting by lowering the Clear to Send (CB) line. Note that this type of transmit pacing can only be used in a hardwired configuration.

2. XON/XOFF Handshake. The host computer or printer uses the ASCII control codes XON(P_1) and XOFF (P_3) to start and stop the terminal from transmitting. Note that a single XON code cancels any number of preceding XOFF codes.

Note that this type of transmit pacing can only be used in a hard wired configuration. In a full-duplex environment, the terminal can also participate in the XON/XOFF handshake form of receive pacing, in which the terminal uses the ASCII control codes XON (P_1) and XOFF (P_3) to start and stop the host computer from transmitting.

Note that a single XON code cancels any number of XOFF codes.

The terminal can also participate in an ENQ/ACK handshake (which is a Hewlett-Packard handshaking mechanism). With this form of handshaking, the host computer transmits a block of data and then sends an ASCII <ENQ> control code. The terminal responds to the <ENQ> by sending back an ASCII <ACK> control code when it has processed all of the data preceding the <ENQ>. The general interpretation of these two control codes is as follows:

ENQ: "Have you processed the data up to this point?"
ACK: "Yes, I have."

The above pacing mechanisms are responded to by the terminal in the following order of precedence:

1. Hardware handshaking pacing (highest priority)

2. XON/XOFF transmit pacing

3. XON/XOFF receive pacing

4. ENQ/ACK pacing (lowest priority)

STATUS

SECTION VI

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 three 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 toggle keys, various error and pending flags, and the following configuration menu fields (see Section II, table 2-1 for descriptions of the fields):

   Xmit Fnctn

   SPOW

   Inh Eol Wrp

   Line/Page

   Inh Hndshk

   Inh DC2

3. Secondary Terminal Status. This request returns seven bytes that report the status of the memory lock, buffer memory, and I/O firmware.

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: $_{CR}$ or $_{CR}$ $_{LF}$

Block Line Mode: $_{CR}$ or $_{CR}$ $_{LF}$

Block Pace Mode: BlkTermnator

In either character mode or block line mode, the $_{CR}$ $_{LF}$ is used if auto line feed mode is enabled. In block page mode, the block terminator is BlkTerminator ($_{ES}$).

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

```
InhHndShk(G) = YES        No handshake
Inh DC2(H)   = YES

InhHndShk(G) = NO          P1
Inh DC2(H)   = YES or NO

InhHndShk(G) = YES         P1/P2/P1
Inh DC2(H)   = NO
```

Computer Museum

INTERPRETING STATUS

For primary and secondary status requests, the terminal returns an escape sequence followed by a string of bytes. The status information is contained in the lower four bits of each byte. The upper four bits are set so that the byte translates into one of the 16 ASCII characters shown in table 6-1.

For a terminal ID request, the terminal returns the 5-character ASCII string "2382A".

Table 6-1. ASCII Status Characters

ASCII CHARACTER	BINARY
0	0011 0000
1	0011 0001
2	0011 0010
3	0011 0011
4	0011 0100
5	0011 0101
6	0011 0110
7	0011 0111
8	0011 1000
9	0011 1001
:	0011 1010
;	0011 1011
<	0011 1100
=	0011 1101
>	0011 1110
?	0011 1111

Terminal ID Status

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

ESC * 5 ^

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

2382A

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" or "modes" level of function keys) is pressed. There are two terminal status requests: primary and secondary. Each returns a set of 7 status bytes.

PRIMARY TERMINAL STATUS. You request the first set of terminal status bytes (bytes 0-6) by issuing the following escape sequence:

ESC ^

The terminal responds with an ESC \, and seven status bytes followed by a terminator. A typical primary terminal status request and response is illustrated in figure 6-1. The example assumes that the DC1 handshake is being used and that the appropriate terminator is a CR.

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

ESC ~

The terminal responds with an ESC :, and seven status bytes followed by a terminator. A typical secondary terminal status request and response is illustrated in figure 6-2. The example assumes that the DC1 handshake is being used and that the appropriate terminator is a CR.

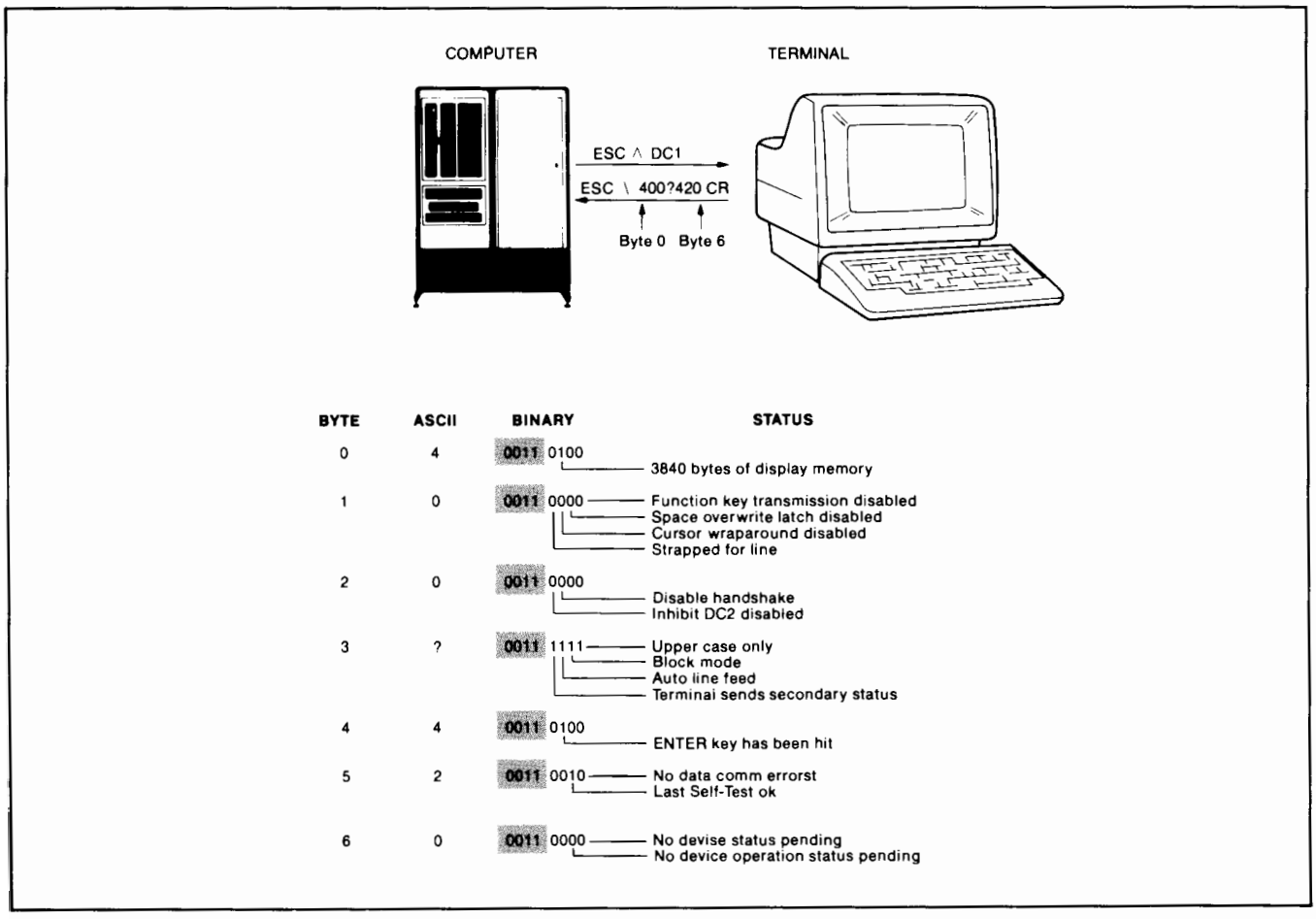


Figure 6-1. Primary Terminal Status Example

PRIMARY STATUS BYTES

BYTE 0 DISPLAY MEMORY SIZE

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

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

This byte specifies the amount of display memory available in the terminal. Note that the HP 2622A and HP 2623A terminals always return 4K bytes.

BYTE 1 CONFIGURATION STRAPS A–D

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

- Strap D — Page Line: 1 = Page, 0 = Line
- Strap C — (Inhibit End-of-Line Wraparound): 1 = yes (Enabled), 0 = no (Disabled)
- Strap 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.

BYTE 2 CONFIGURATION STRAPS E–H

8	7	6	5	4	3	2	1
0	0	1	1	1/0	1/0	C	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 TOGGLE KEYS

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

- Terminal sends secondary status
- AUTO LF Key: 1 = auto LF, 0 = no auto LF
- CAPS LOCK Key: 1 upper case only, 0 upper and lower case
- BLOCK MODE Key: 1 block mode, 0 character mode

BYTE 4 TRANSFER PENDING FLAGS

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

- Secondary Status Pending: 1 yes, 0 no
- ENTER Key Pending: 1 yes, 0 no
- Cursor Sense Pending: 1 yes, 0 no
- Function Key Pending: 1 yes, 0 no

BYTE 5 ERROR FLAGS

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

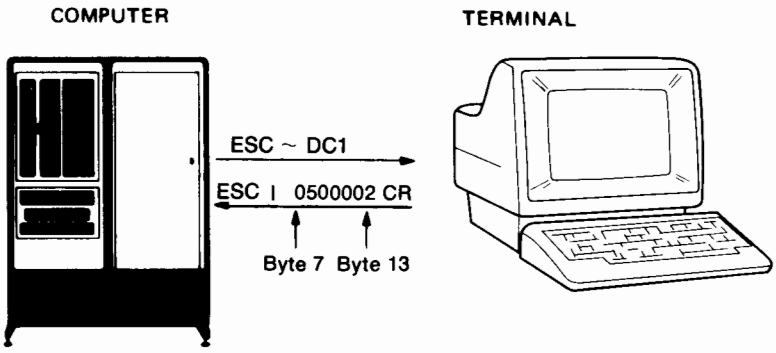
- Data Comm: 1 parity or buffer overflow error or framing overrun error; 0 no error
- Self-Test: 1 no error, 0 error

BYTE 6 DEVICE TRANSFER PENDING FLAGS

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

(tracks "S", "F", or "U" completion codes associated with &p device control sequences.)

Figure 6-1. Primary Terminal Status Example (Continued)

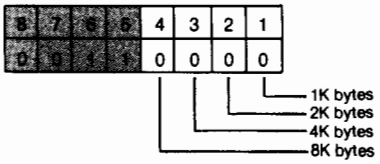


BYTE	ASCII	BINARY	STATUS
7	0	0011 0000	
8	5	0011 0100	Terminal identifies self
9	0	0011 0000	
10	0	0011 0000	
11	0	0011 0000	
12	0	0011 0000	
13	2	0011 0010	Memory locked

Figure 6-2. Secondary Terminal Status Example

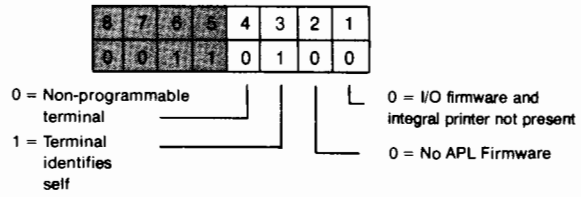
SECONDARY STATUS BYTES

BYTE 7 BUFFER MEMORY (always zero)



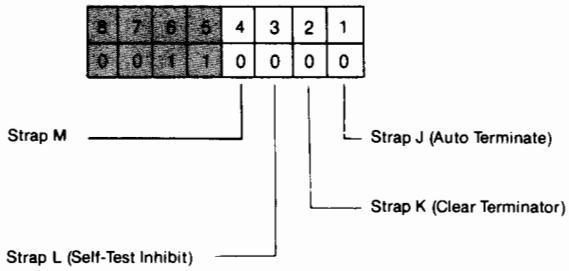
Memory installed in addition to display memory that is available for use as data buffers. Note that the HP 2622A and HP 2623A terminals always return a 0 value.

BYTE 8 TERMINAL FIRMWARE CONFIGURATION



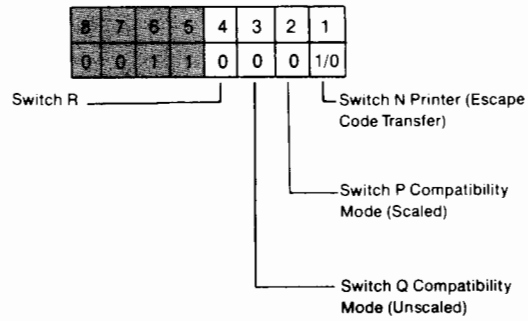
APL Firmware does not apply.

BYTE 9 CONFIGURATION STRAPS J–M (always zero)



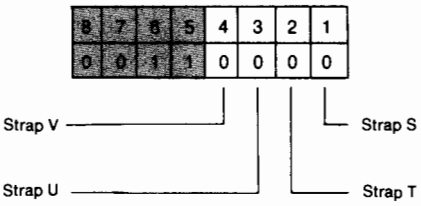
Straps J–M do not apply to the terminal.

BYTE 10 KEYBOARD INTERFACE KEYS (N–R)



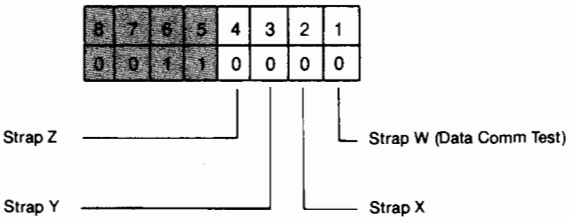
Straps N–R do not apply to the HP 2622A terminal.

BYTE 11 CONFIGURATION STRAPS S–V (always zero)



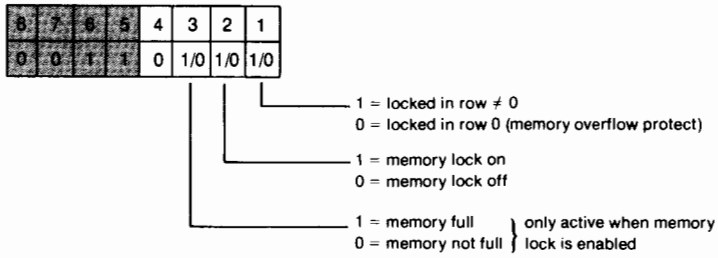
Straps S–V do not apply to the teminal.

BYTE 12 CONFIGURATION STRAPS W–Z (always zero)



Straps W, X, Y, and Z do not apply to the terminal.

BYTE 13 MEMORY LOCK MODE



Computer Museum

ERRORS AND SELF-TESTS

SECTION VII

ERRORS

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) along with the message "Press RETURN to clear". Pressing RETURN unlocks the keyboard, clears the message, and reinstates the function key labels.

TERMINAL SELF-TESTS

The HP 2382A Terminal tests itself. This section describes the types of self-tests that are incorporated into the terminal.

The HP 2382A includes four types of self-tests:

- Power-On Test
- Terminal Test
- Identify ROMs
- Datacomm Test

The Power-On Test is automatically initiated as the result of a power-on sequence. TERMINAL TEST , IDENTIFY ROMS and DATACOMM TEST are initiated using the "service keys" level of function keys. The "Terminal Test" can also be initiated programmatically or by using a "MODES" function key.

Power-On Test

The Power-On Test does the following:

1. Tests the processor and verifies the integrity of all RAM (Random-Access Memory) chips and ROM (Read-Only Memory) chips within the terminal.

If the Power-On Test results are normal:

1. at power on, the terminal beeps once
2. does the test for about 15 seconds
3. beeps once again
4. brings up the modes level of function key labels on the screen

If an error is found, one of the following will occur when the terminal is turned on:

a. The terminal will fail to beep at all.

b. The terminal will beep continuously.

c. After the first beep the terminal will beep 1 to 11 times and no softkey labels will appear on the screen.

If one of the above 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. Non-destructively verifies the integrity of all RAM chips within the terminal.
3. Verifies the integrity of all ROM chips within the terminal.
4. Displays the test pattern shown in figure 7-1 or 7-2 depending on whether or not the optional line drawing character set is installed.

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

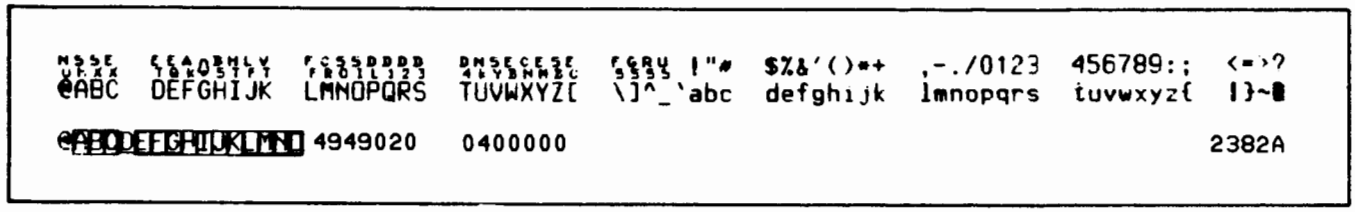


Figure 7-1. Screen Test Pattern, Standard Terminal

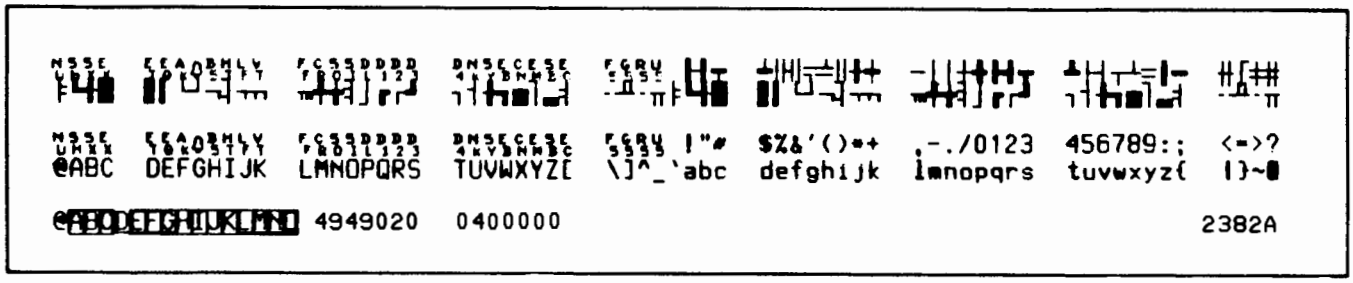


Figure 7-2. Screen Test Pattern, Option 202 (Line Drawing Set)

If a ROM error is detected, the following message is displayed across the bottom of the screen:

ROM ERR#x
Press RETURN to clear

where "x" is a number from 1 to 3. This message contains information identifying the bad ROM chip(s) and describing the nature of the detected error condition. In such a case, or for any other error message, write down the message so you can relate it accurately to your HP Service Representative over the telephone (this allows him to arrive prepared with the proper replacement parts).

If a RAM error is detected, the folowing message is displayed across the bottom of the screen:

RAM ERR#x

where "x" is a number from 1 to 8. The message contains information identifying the bad RAM chip(s) and describing the nature of the detected error condition. Write down the error message and call your nearest HP Service Representative.

To clear the error message press the RETURN key.

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:

AIDS [f5] aids & features [f3] service keys [f7] DATACOMM TEST

A list similar to the one shown in figure 7-3 is displayed on the screen.

```
FIRMWARE ROMS
1818-1796 2128
1818-1797 2128
1818-1798 2128
```

Figure 7-3. ROM Identification Listing

Datacomm Test

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

AIDS [f5] aids & features [f3] service keys [f8] IDENTIFY ROMS

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

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

Datacomm Error 1 will appear if no hood is used.

This test hood is available in a set of hoods which can be ordered as the HP 13259A Data Comm Self-Test Connector Kit.

The loopback test consists of a data loopback operation (which checks to see that the character sent is the same as the character received), a baud rate test (which verifies that the baud rate mechanism is functioning properly within ± 2% of the configured baud rate), and a modem control line test.

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

The `Err#` field contains a numeric error code which is interpreted as follows:

1 = Test connector not present

2 = Baud rate too fast

3 = Baud rate too slow

4 = Error in control lines

5 = Character did not loop back

6 = Received character NOT the same as one transmitted

7 = Framing error in character

8 = A character was overrun

TERMINAL MAINTENANCE PROCEDURES

APPENDIX
A

This section provides information on preventive maintenance for your HP terminal.

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.

ESCAPE CODES

APPENDIX B

KEY(S)		CODE	FUNCTION
TERMINAL CONTROL FUNCTION			
[redacted]	(as used in Local mode)	↯ 0	Copy memory to destination(s)
[redacted]	margins / tabs / col SET TAB	↯ 1	Set tab
[redacted]	margins tabs / col CLEAR TAB	↯ 2	Clear tab
[redacted]	margins / tabs / col CLR ALL TABS	↯ 3	Clear all tabs
[redacted]	margins / tabs / col LEFT MARGIN	↯ 4	Set left margin
[redacted]	margins / tabs / col RIGHT MARGIN	↯ 5	Set right margin
[redacted]	margins / tabs / col CLR ALL MARGINS	↯ 9	Clear all margins
		↯ :	Delay one second
[redacted]		↯ A	Cursor up
[redacted]		↯ B	Cursor down
[redacted]		↯ C	Cursor right
[redacted]		↯ D	Cursor left

KEY(S)		CODE	FUNCTION
[redacted]		↯ E	Hard reset (power on reset)
[redacted]		↯ F	Cursor home down
[redacted]	(with Auto LF disabled)	↯ G	Move cursor to left margin
[redacted]		↯ H	Cursor home up
[redacted]		↯ I	Horizontal tab
[redacted]		↯ J	Clear display from cursor to end of memory
[redacted]		↯ K	Clear line from cursor to end of line
[redacted]		↯ L	Insert line
[redacted]		↯ M	Delete line
[redacted]		↯ P	Delete character
[redacted]		↯ Q	Start insert character mode
[redacted]	*	↯ R	End insert character (↯ Q)
[redacted]		↯ S	Roll up
[redacted]		↯ T	Roll down
[redacted]		↯ U	Next page
[redacted]		↯ V	Previous page

KEY(S)			CODE	FUNCTION
TERMINAL CONTROL FUNCTION (continued)				
			↯ W	Format mode on
			↯ X	Format mode off
[ALPHA]	MODES	DISPLAY FUNCTION	↯ Y	Display Functions mode on
[ALPHA]	MODES	DISPLAY FUNCTNS •	↯ Z	Display Functions mode off
			↯ [	Start unprotected field
			↯]	End unprotected/transmit-only field
			↯ ^	Primary terminal status request
			↯ \	Sense cursor position (relative)
			↯ a	Sense cursor position (absolute)
			↯ b	Unlock keyboard
			↯ c	Lock keyboard
			↯ d	Transmit a block of text to computer
			↯ f	Modem disconnect
[CNTL]	[RESET / BREAK]		↯ g	Soft reset

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

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

CURSOR CONTROL OPERATIONS

NOTE

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

↯ &a <col >c <rows >Y	Moves the cursor to column "col" and screen row "row" of the screen (screen relative addressing).
↯ &a <col >c <row >R	Moves the cursor to column "col" and row "row" in memory (absolute addressing).
↯ &a ± <col >c ± <row > Y	Moves the cursor to column "col" and row "row" (on the screen) relative to its present position ("col" and "row" are signed integers). A positive number indicates right or downward movement and a negative number indicates left or upward movement.
↯ &a ± <col >c ± <row > R	Moves the curso r to column "col" and row "row" relative to its present cursor position in memory ("col" and "row" are signed integers). A positive number indicates right or downward movement and a negative number indicates left or upward movement.

Computer Museum

CONFIGURATION OPERATIONS

ESC &q 0L	Unlock configuration
ESC &q 1L	Lock configuration

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

FORMAT MODE

ESC {	Starts a field
ESC }	Ends the field

FUNCTION KEY AND ERROR MESSAGE OPERATIONS

ESC &j <string length>L <message>

To enable and disable the function keys (F1 thru F8) use the following escape sequence:

ESC &j <x>

x	MEANING
A	Display the Modes set of function key labels.
B	Enable the User function keys. (The user key labels are displayed.)
C	Clear message and return function key label to the screen.
@	Disable the function keys and remove the function key labels from the screen.

To enable or disable the Function Control keys:

S	Disables the [redacted] key.
R	Enables the [redacted] key.

To define functions for the function keys:

ESC &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	[redacted] function key	
	2	[redacted] function key	
	3	[redacted] function key	
	4	[redacted] function key	
	5	[redacted] function key	
	6	[redacted] function key	
	7	[redacted] function key	
	8	[redacted] function key	
Label length (d)	0 thru 16	Number of characters in the label.	0
String length (l)	0 thru 80	Number of characters in the string.	1
	-1	Clears the content of the string.	
Label	(none)	The label is entered at this point in the sequence.	
String	(none)	The character string is entered at this point in the sequence.	

DISPLAY ENHANCEMENTS OPERATIONS

To start and end display enhancements

ESC &d <char> Selects the display enhancement indicated by "char" to begin at the present cursor position

	"char"														
	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Half-Bright									X	X	X	X	X	X	X	X
Under-line					X	X	X	X					X	X	X	X
Inverse Video			X	X			X	X			X	X			X	X
Blinking		X		X		X		X		X		X		X		X
End Enhance-ment	X															

DISPLAY WINDOW ON/OFF

To turn off the display window (the top 24 lines of the screen):

```
ESC & w13F - turns off display

ESC & w12F - turns on display
```

Possible use of these escape sequences, is for dumping a page of data to display memory with the user seeing it download.

ALTERNATE CHARACTER SET SELECTION

ESC) B	Selects line drawing as alternate character set.
ESC) 0	Selects base character set
ESC) A	Selects Extended Roman as alternate character set

KEYBOARDS AND CHARACTER SETS

APPENDIX C

Depending on the option selected, a terminal comes with one of three types of character ROM installed, as shown below:

OPTION	CHARACTER SET(S)
Standard	USASCII
001–006	USASCII, Extended Roman, and Line-Drawing sets
202	USASCII and Line-Drawing sets

Figures C-1 through C-7 show the national keyboards which are available as options 001 through 006. These options also include the extended Roman character set ROM. This ROM contains many characters used in the languages represented in options 001 through 006 but not found in the USASCII character set (see tables C-1 and C-2).

Computer Museum

Table C-1. Standard ISO/ASCII Character Codes

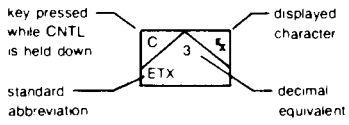
BIT 7 6 5 / 4 3 2 1	CONTROL (CNTL) CHARACTERS 0 0 0	0 0 1	DISPLAYABLE CHARACTERS 0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1
0000	@ 0 $^{N}_{U}$ NUL	P 16 DL DLE	SP	0	●	P	'	p
0001	A 1 $^{S}_{H}$ SOH	Q 17 $^{D}_{1}$ DC1	!	1	A	Q	a	q
0010	B 2 $^{S}_{X}$ STX	R 18 $^{D}_{2}$ DC2	"	2	B	R	b	r
0011	C 3 $^{E}_{X}$ ETX	S 19 $^{D}_{3}$ DC3	#	3	C	S	c	s
0100	D 4 $^{E}_{T}$ EOT	T 20 $^{D}_{4}$ DC4	$	4	D	T	d	t
0101	E 5 $^{E}_{Q}$ ENQ	U 21 $^{N}_{K}$ NAK	%	5	E	U	e	u
0110	F 6 $^{A}_{K}$ ACK	V 22 $^{S}_{Y}$ SYN	&	6	F	V	f	v
0111	G 7 BEL	W 23 $^{E}_{B}$ ETB	'	7	G	W	g	w
1000	H 8 $^{B}_{S}$ BS	X 24 $^{C}_{N}$ CAN	(	8	H	X	h	x
1001	I 9 $^{H}_{T}$ HT	Y 25 $^{E}_{M}$ EM	)	9	I	Y	i	y
1010	J 10 $^{L}_{F}$ LF	Z 26 $^{S}_{B}$ SUB	*	:	J	Z	j	z
1011	K 11 $^{V}_{T}$ VT	[27 $^{E}_{C}$ ESC	+	;	K	[	k	{
1100	L 12 $^{F}_{F}$ FF	\ 28 $^{F}_{S}$ FS	,	<	L	\	l	|
1101	M 13 $^{C}_{R}$ CH	] 29 $^{G}_{S}$ GS	-	=	M	]	m	}
1110	N 14 $^{S}_{O}$ SO	^ 30 $^{R}_{S}$ RS	.	>	N	^	n	~
1111	O 15 $^{S}_{I}$ SI	_ 31 $^{U}_{S}$ US	/	?	O	_	o	● DEL

Table C-2. Extended Roman Character Codes

BIT 4321 \ 765	000	001	010	011	100	101	110	111
	$B_8 = 1$ EXTENDED ROMAN CHARACTERS							
0000				–/–	â	Â		
0001					ê	î		
0010					ô	Ø		
0011				•	û	Æ		
0100					á	å		
0101				Ç	é	Í		
0110				Ñ	ó	ø		
0111				ñ	ú	æ		
1000			. '	¡	à	Ä		
1001			`	¿	è	ì		
1010			^	¤	ò	Ö		
1011			¨	£	ù	ü		
1100			~		ä	É		
1101				§	ë	ï		
1110					ö	ß		
1111			£		ü			

— ACKNOWLEDGE
— BELL
— BACKSPACE
— CANCEL LINE
— CARRIAGE RETURN
— DATA LINK ESCAPE
— DEVICE CONTROL 1
— DEVICE CONTROL 2
— DEVICE CONTROL 3
— DEVICE CONTROL 4
— DELETE
— END OF MEDIUM
— ENQUIRY
— END OF TRANSMISSION
— ESCAPE
— END OF BLOCK
— END OF TEXT
— FORM FEED
— FILE SEPARATOR
— GROUP SEPARATOR
— HORIZONTAL TAB
— LINE FEED
— NEGATIVE ACKNOWLEDGE
— RECORD SEPARATOR
— SHIFT IN
— SHIFT OUT
SP — SPACE
— START OF HEADING
— START OF TEXT
— SUBSTITUTE
— SYNCHRONOUS IDLE
— UNIT SEPARATOR
— VERTICAL TAB

Control Character Legend



The French keyboard (option 003), when delivered, is physically arranged in the AZERTY layout; a keycap extraction tool comes with it. To change the keyboard to the QWERTY layout, the A, Z, Q, and W keys must be rearranged to be as shown in figure C-4.

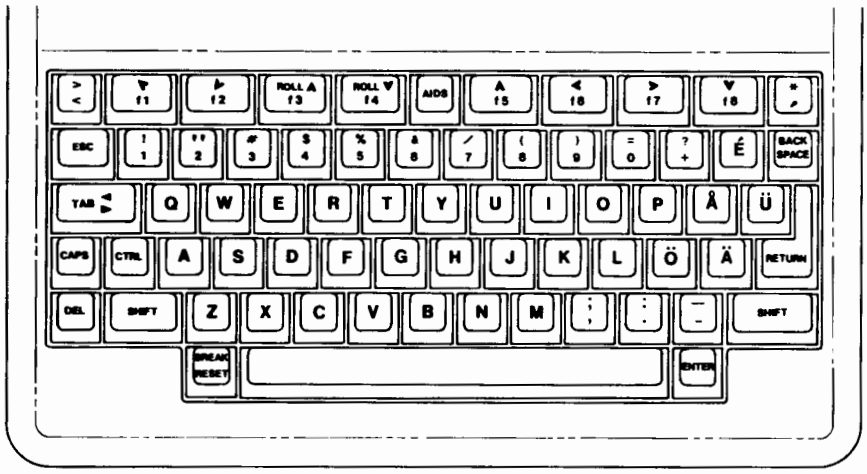


Figure C-1. Swedish/Finnish Keyboard (Option 001)

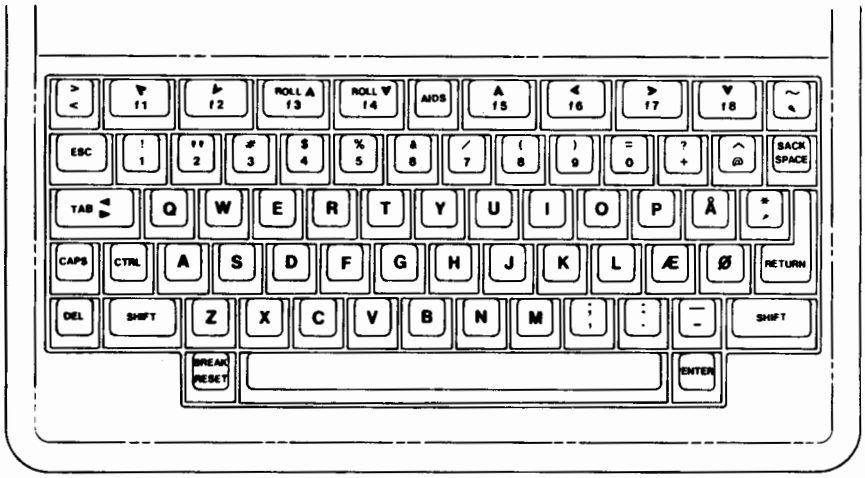


Figure C-2. Danish/Norwegian Keyboard (Option 002)

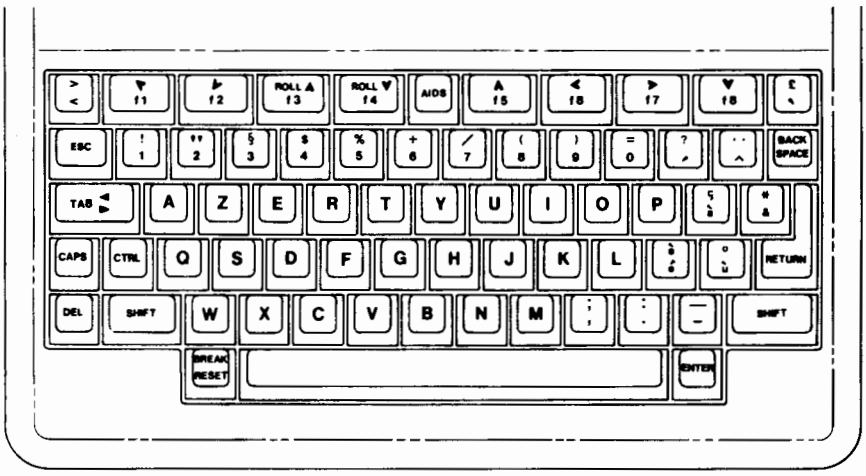


Figure C-3. French Keyboard (Option 003), AZERTY Layout

Computer Museum

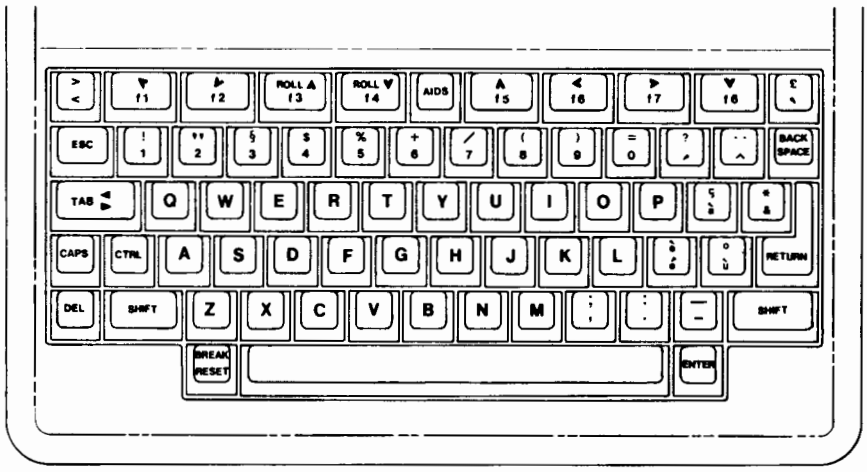


Figure C-4. French Keyboard (Option 003), QWERTY Layout

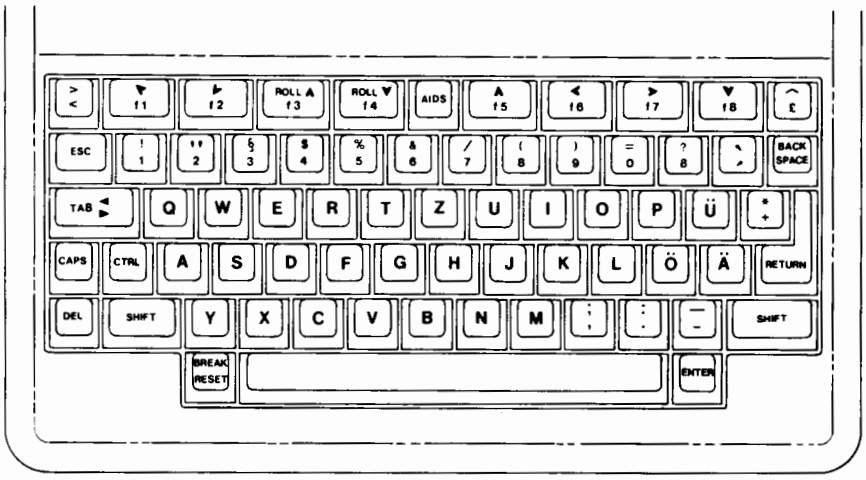


Figure C-5. German Keyboard (Option 004)

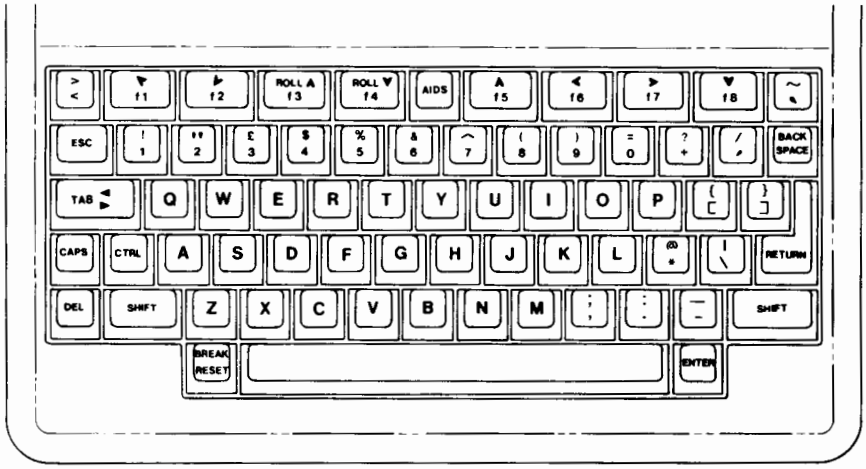


Figure C-6. United Kingdom Keyboard (Option 005)

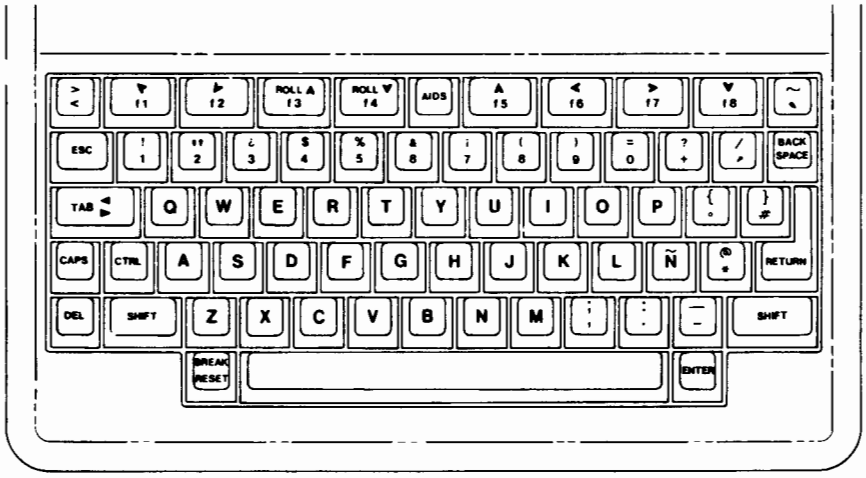


Figure C-7. Spanish Keyboard (Option 006)

7-Bit Vs. 8-Bit Operation

The terminal can operate in 7-bit or 8-bit mode. It is configured for 7-bit operation by setting switches 5, 6, 7 in group A to 1, 0, 0 respectively, or by setting the ASCII 8 bit field of the Terminal Configuration menu to "NO". The terminal is configured for 8-bit operation by setting switches 5, 6, 7 in group A to 1, 0, 1 respectively, or by setting the ASCII 8 bit field of the Terminal Configuration menu to "YES". Parity operation is allowed only in 7-bit mode.

Alternate Character Set Selection

The "base" language for the terminal is the language selected on the Terminal Configuration menu. The alternate character set can be the line-drawing set (which comes with options 001 through 006 and option 202) or the extended Roman character set (which comes with options 001 through 006).

One of these two sets is selected as the "active" alternate character set using an escape sequence, which can be entered from either a program or the keyboard. Here is a list of the appropriate escape sequences:

ESC) B	Base set
ESC) A	Extended Roman Set (only in 8 bit mode)
ESC) B	Line Drawing Set

Note that in 7 bit mode, the extended Roman set cannot be chosen as the "active" alternate character set.

At power-up time or after a hard reset, the terminal defaults to the extended Roman character set as the active alternate character set if the terminal is configured for 8-bit operation and to the line-drawing set if the terminal is configured for 7-bit operation.

Pressing the CTRL and N keys simultaneously selects the active alternate character set as the character set accessed by the keyboard keys. Pressing the CTRL and O keys simultaneously returns the terminal to the base language.

When the terminal is in 7-bit mode and the active alternate character set is selected, it remains the character set accessed by the keyboard through the end of the current line. The alternate character set must be selected anew for each line.

When the terminal is in 8-bit mode and the active alternate character set is selected, it remains the character set accessed by the keyboard until the base set is reselected with a CTRL O.

ISO/USASCII Character Set

If the terminal is equipped with the ROM containing the extended Roman character set (table C-2), and is configured for 7-bit operation, the characters #, @, [, \,], ^, `, {, |, }, and ~ in table C-1 are replaced on the screen with one of the following sets of characters, depending on the national language selected on the Terminal Configuration menu.

	KEYBOARD	DECIMAL VALUE										
LANGUAGE	OPTION #	35	64	91	92	93	94	96	123	124	125	126
USASCII	(standard)	#	@	[	\	]	^	`	{	|	}	~
Swedish/Finnish	001	#	É	Ä	Ö	Å	Ü	é	ä	ö	å	ü
Danish/Norwegian	002	#	@	Æ	Ø	Å	^	`	æ	ø	å	~
French	003	£	à	°	ç	§	^	`	é	ù	è	¨
German	004	£	§	Ä	Ö	Ü	^	`	ä	ö	ü	ß
United Kingdom	005	£	@	[	\	]	^	`	{	|	}	~
Spanish	006	#	@	¡	Ñ	¿	^	`	{	ñ	}	~

Extended Roman Character Set

All USASCII (table C-1) and extended Roman characters (table C-2) are accessible using the keyboard keys under the following conditions:

a. The terminal is configured for 8-bit operation.

b. The extended Roman character set is selected as the active alternate character set.

c. The ASCII <SO> code (or CTRL N from the keyboard) and the ASCII <SI> code (or CTRL O) are used to flip back and forth between the USASCII and Extended Roman character sets.

In 8-bit mode, the eighth data bit is used by the firmware code to select either the USASCII or extended Roman character set. If the terminal is configured for 8-bit mode but does not include the 8K character ROM which contains the extended Roman character set, the eighth bit is still used for selection of the USASCII character set, but any keystrokes which would access the extended Roman characters, if the appropriate ROM was present, display spaces.

If the terminal is configured for 7-bit operation, and the extended Roman character set is selected as the active alternate character set, the terminal defaults to the base character set.

Line-Drawing Set

The line-drawing set consists of various line setments. Using it, you can construct data entry forms by combining different line segments. Each line segment is associated with one of the keyboard keys as shown in figures C-8 and C-9.

The line-drawing set line segments are accessed by selecting the line-drawing set as the active alternate character set using the escape sequence ESC) B and then selecting it as the character set accessed by the keyboard keys by pressing the CTRL and N keys, simultaneously.

Figure C-8. Line-Drawing Set Elements

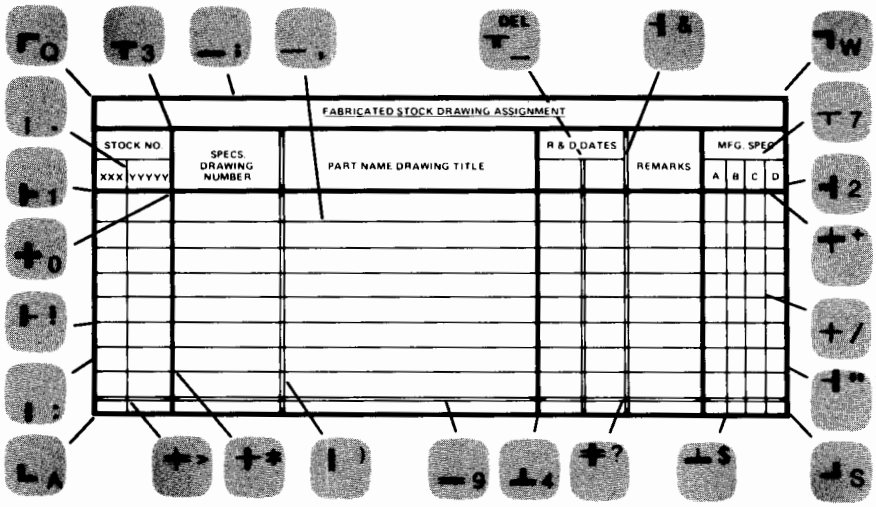


Figure C-9. Sample Data Entry Form

A

Absolute Addressing 4-6
Addressing, Absolute 4-6
  Combining Absolute and Relative 4-7
  Cursor Relative 4-6
  Screen Relative 4-5
AIDS Key 1-3
ASCII Character Set C-2
Attribute Field, User Key 3-5
Auto Line Feed Mode 3-3

B

Back Tab 4-11
BaudRate Configuration 5-4
BlkTermnator Configuration 2-5
Block Mode 3-1
BREAK Key 3-10
Buffer, Receive 5-6

C

Caps Lock Mode 3-4
Caps Mode 3-4
ChkParity Configuration 5-5
Cleaning the Screen and Keyboard A-1
Clear Display 4-9
Clear Line 4-9
Clearing Margins 4-10
Clearing Tabs 4-10
Combining Absolute and Relative Addressing..... 4-7
"configkeys" Function Keys 2-1
Configurtion Lock/Unlock 2-7
Configuring the Terminal 1-5,2-1,5-3
Cursor Control 4-1
Cursor Down 4-2
Cursor Left 4-2
cursor Relative Addrressing 4-6
Cursor Right 4-2
Cursor Up 4-2

D

Datacomm Configuration Menu 5-4
Datacomm Test 7-2
Data Communications 1-6,5-1
Data Entry Forms,Designing and Using 4-12
Defining User Keys Locally 3-5
Defining User Keys Programmatically 3-6
Delete Line 4-7
Designing Forms 4-12
Disable Keyboard 3-10
Display Enhancements 4-11
Display Functions Mode 3-3
Display Lock 3-3
Drawing Forms 4-12

E

Edit Operations 4-7
Enable Keyboard 3-10
Enhancements,Display 4-11
EnqAck Configuration 5-5
ENTER Key 3-7
Error Messages 7-1
Errors,Receive 5-7

Computer Museum

F

Fields,Protected and Unprotected 4-15
FldSeparator Configuration 2-6
Format Mode 3-2,4-15
Forms,Designing and Using 4-12
FrameRate Configuration 2-2
Full Duplex Operation 5-7
Function Key Hierarchy 1-4
Function Keys 1-3
Function Keys,"config keys" 2-1

H

Hard Reset 3-11
Hierarchy,Function Key 1-3
Home Down 4-2
Home Up 4-1

I

ID Status 6-2
Identify ROMs 7-2
Inh DC2(H) Configuration 2-4
IhnEo1Wrp(C) Configuration 2-4
InhHndShk(G) Configuration 2-4
Insert Character 4-8
Insert Line 4-7
Installing a Point-to-Point Configuration 5-3

K

Key Definition Field, User Key 3-5
Keyboard 1-3
Keyboard Control 3-1
Keyboard,Cleaning A-1
  Enable/Disable 3-10
Keyboards,National C-1

L

Label Field,User Key 3-5
Line Modify Mode 3-2
Line/Page(D) Configuration 2-4
Local Mode 3-1,5-7
Lock Configuration 2-7

M

Margins,Setting and Clearing 4-10
Memory Lock Mode 3-3
Menu,Datacomm Configuration 5-4
  Terminal Configuration 2-1
  User Keys Definition 1-5,3-5
Messages,Error 7-1
MODES Function Key 1-3
Modify All Mode 3-2
Modify Mode,Line 3-2

N

National Keyboards C-1
Next Page 4-3

O

Options,Keyboard 1-2
Overflow Protect 3-3

P

Pacing Mechanisms 5-7
Page,Next/Previous 4-3
Parity Checking 5-6
Parity Configuration 5-4
Point-to-Point Configuration, Installing 5-3
Point-to-Point Programming Information 5-6
Power-On Configuration Switches 2-8
Power-On Test 7-1
Previous Page 4-3
Primary Terminal Status 6-2
Programming Configuration 2-7
Protected Fields 4-15

R

Receive Buffer 5-6
Receive Errors 5-7
RecvPace Configuration 5-5
Relative Addressing,Cursor 4-6
  Screen 4-5
Remote Mode 3-1
Reset 3-10
RETURN Def Configuration 2-2
Roll Text Down 4-3
Roll Text Up 4-2
ROMs,Identify 7-2

S

Screen Relative Addressing 4-5
Screen,Cleaning A-1
Secondary Terminal Status 6-2
Self-Tests 1-6,7-1
Send Display (ecd) 3-9
Setting Margins 4-10
Setting Tabs 4-10
Soft Reset 3-10
SPOW(B) Configuration 2-4
SR(CH) Configuration 5-5
Start Bits 5-6
StartCol Configuration 2-3
Status,Terminal 6-1
Stop Bits 5-6
Switches,Configuration 2-8
  Terminal Configuration 2-2
  Datacomm Configuration 5-4

T

Tab 4-11
Tabs,Setting and Clearing 4-10
Terminal Configuration Menu 2-1
Terminal Maintenance Procedures A-1
Terminal Status 6-1
Terminal Test 7-1

U

Unlock Configuration 2-7
Unprotected Fields 4-15
User-Definable Keys 3-4
User Keys 3-4
User Keys Definition Menu 1-5,3-5
User Keys,Enabling 1-4

V

Video Enhancements 4-11

X

XmitFnctn(A) Configuration 2-4
XmitPace Configuration 5-5