4870

FLAT PANEL INDUSTRIAL TERMINAL

Printed in the United States of America
Part Number 92891-001

XYCOM
750 North Maple Road
Saline, Michigan 48176
(313) 429-4971

REVISION RECORD

REVISION	DESCRIPTION
A	Manual Released 6/88

MANUAL
P/N:
92891-001

Address comments concerning
this manual to

xycom
Technical Publications Dept
750 North Maple Rd
Saline, Michigan 48176

WARNING

Dangerous voltages are present within the Industrial Terminals and may **remain** present after electrical power is disconnected Use **caution** and avoid touching high-voltage areas when working on the terminals with their top cover removed! **Do not work alone.**

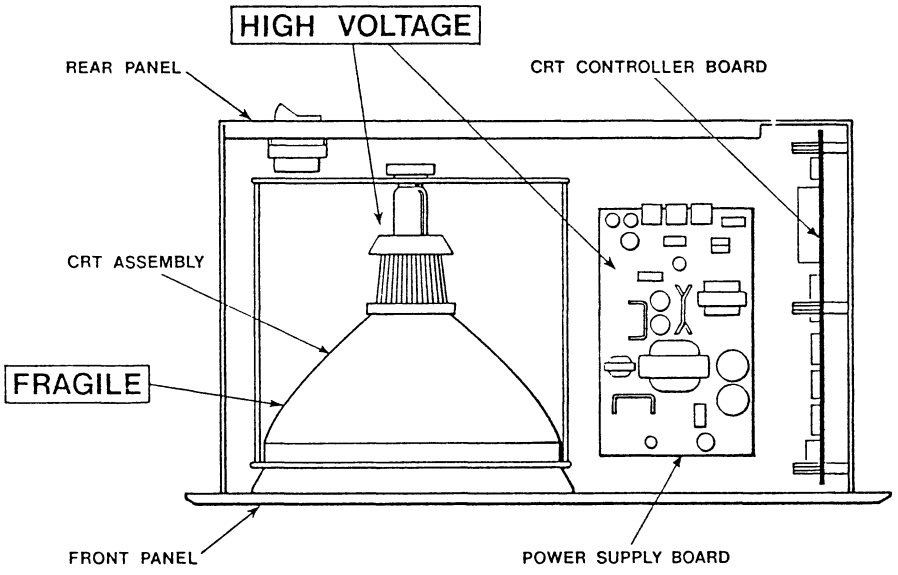


TABLE OF CONTENTS

TABLE OF CONTENTS (continued)

TABLE OF CONTENTS (continued)

TABLE OF CONTENTS (continued)

LIST OF FIGURES

TABLE OF CONTENTS (continued)

LIST OF FIGURES

LIST OF TABLES

Chapter 1

INTRODUCTION

1 1 INTRODUCTION

The Xycom 4870 Flat Panel Industrial Terminal is designed and built specifically to meet the mounting, environmental, operational, and reliability requirements of industry, especially where minimal depth is critical This rugged industrial package employs a nine-inch Electroluminescent (EL), Flat Panel amber screen

The Flat Panel Terminal serves as a rugged operator interface anywhere a person must communicate with an automated machine or process - thriving in installations where fragile office equipment simply wouldn't survive This terminal provides more meaningful information and instructions to equipment operators than indicator lights and simple message display units possibly could

Using the terminal's built-in keypad while displaying context-sensitive operator menus, this terminal can easily provide more functionality than traditional push-button, rotary, and thumbwheel switch arrangements, while giving the system designer even greater flexibility

The 4870 Flat Panel Industrial Terminal is menu-configurable to emulate either a Hazeltine 1500 terminal or an ANSI-compatible terminal (ANSI Standard X3 64-1979) When configured as an ANSI-compatible terminal, it provides DEC VT100/220 support This is an intelligent terminal, able to execute special remote commands to format the screen and draw a variety of figures, such as lines, boxes, process control symbols, and high-resolution bar graphs

The 4870 houses a 9" (diagonal) amber video display It is enclosed with the terminal control electronics and power supply The 4870 can be mounted in a standard EIA 19" equipment rack In addition, the 4870 can be panel-mounted to meet NEMA 4 and NEMA 12 specifications

This terminal is equipped with an RS-232C communications port and a built-in 28-key sealed membrane keypad Attachable full-size keyboards are optionally available

In addition, optional modules can be added to any terminal to add the following capabilities

- Programmable Controller Interfaces
- Data Registers
- Operator Interface Language programmability
- Time-of-Day Clock/Calendar
- Non-volatile Screen Memory
- Multidrop Networking
- Keyboards
- Auxiliary Communications Port

See your Xycom representative for details of your terminal's expansion features

1 2 MANUAL STRUCTURE

This manual consists of seven chapters which divide the various aspects of terminal specifications and operation into seven distinct areas These aspects are developed in the following progression

Chapter One - A general description of the 4870 Flat Panel Industrial Terminal and functional specifications

Chapter Two - Terminal Installation information covering configuration and start-up

Chapter Three - A description of the keypad and keyboard, including keypad codes and keyboard codes

Chapter Four - Details information on Video Display Format, cursor addressing, attributes, character size considerations, and graphics

Chapter Five - Addresses Remote Commands, Hazeltine 1500 Emulation, and ANSI Emulation

Chapter Six - Covers RS-232C communications, Full and Half-duplex operation, modem usage, RS-232C pin-out

Chapter Seven - Details the 4870 Industrial Terminal's self-diagnostics

The Appendices are designed to provide additional information in terms of the 4870's panel cutout dimensions, VT100/220 codes not supported, graphic's chart, and a Quick Reference Guide

1 3 SPECIFICATIONS

Mounting	fits in standard EIA 19" rack (EIA RS-310-C)	
Front-panel seal	meets NEMA 4 and NEMA 12 specifications	
Dimensions		
4870	Height	- 10 44" (265mm)
	Width	- 19 0" (483mm)
	Depth	- 4 25" (108mm)
Power Requirements	37 watts	@ 115 VAC, 50/60 Hz, 32A
		@ 230 VAC, 50/60 Hz, 16A
	Fuse	Slo-blo 1 5A @ 115 VAC
		75A @ 230 VAC
Temperature	0° to 50°C (32° to 122°F)	
Humidity	5% to 80% RH non-condensing (Extremely low humidity may require protection against static discharge)	

Shock (5 SINE Shock Pulse)

Operational	15g Peak Acceleration ±11 msec Duration
Non-operational	30g Peak Acceleration ±11 msec Duration

Vibration (5 to 2KHz Frequency Range)

Operational Vibration Amplitude	0 006" Peak-to-Peak Displacement 1 0g Peak Acceleration
Non-operational Vibration Amplitude	0 015" Peak-to-Peak Displacement 2 5g Peak Acceleration

Chapter 2

INSTALLATION, CONFIGURATION, AND START-UP

2 1 INSTALLATION

Installation of the industrial terminal involves

- choosing a suitable location
- connecting electrical power to the terminal
- connecting the serial port to a host computer or modem

2 1 1 Installing Expansion Modules

If you have purchased an expansion module for use with the 4870, there is a manual which accompanies it, ignore its (the expansion module's) installation procedure, and follow the procedure below The rest of the expansion module manual should be used because some modules change the operational "personality" of the terminal when installed Refer to the expansion module's manual for configuration, programming, and operating instructions that may have to be used instead of the instructions included in this base terminal manual

The 4870 has 12 screws (be sure to save the screws, they will be needed later) that need to be removed when the expansion module is installed (see Figure 2-1) Once these screws have been removed, separate the front panel from the back of the chassis Gently lay the back of the chassis on a flat surface (see Figure 2-2) Connect the expansion module to the Controller Board using hex standoffs as shown if Figure 2-2 To put the unit back together, reverse the above instructions

WARNING

Be sure the terminal's power supply is disconnected when opening the unit or shock and/or damage to the user or the components may result

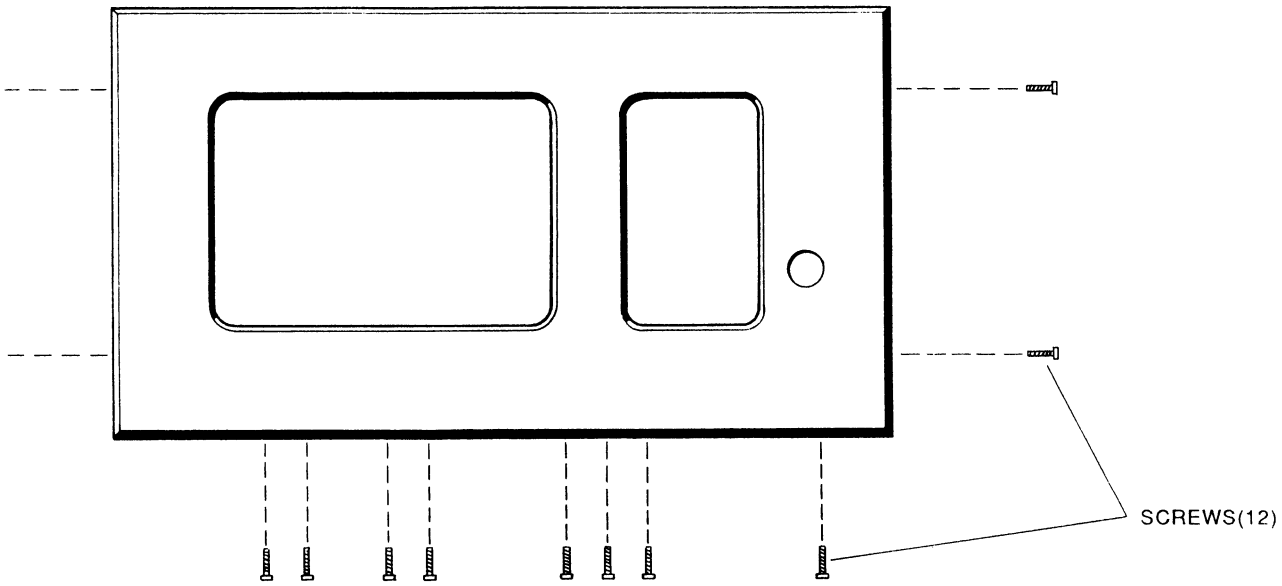


Figure 2-1 Removing the Screws
(Save ALL Screws)

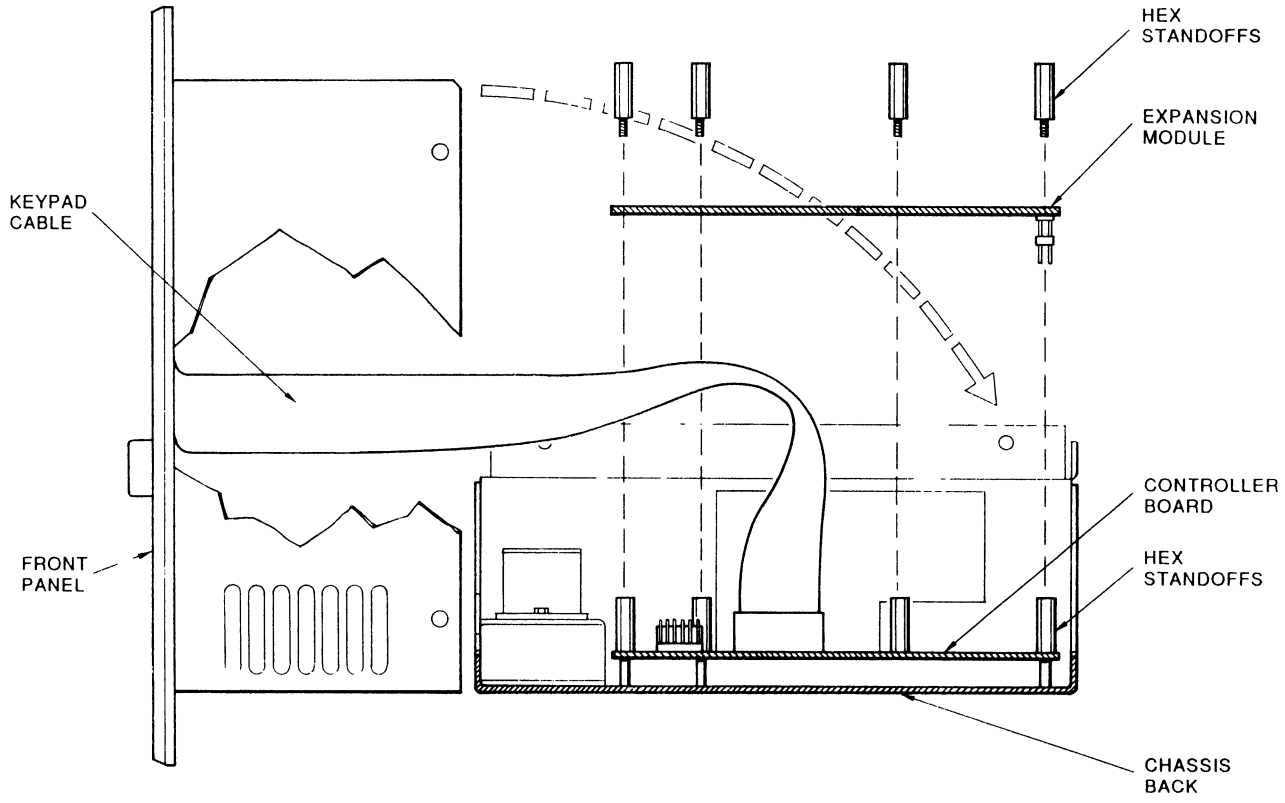


Figure 2-2 Expansion Module Installation

2 1 2 Connecting Electrical Power

The 4870 terminal must be connected to 120 VAC, 50/60 Hz, or 220 VAC, 50/60 Hz Before connecting electrical power to the terminal, put the terminal's **ON/OFF** switch in the **OFF** position by pressing the side of the switch marked "O" The ON/OFF switch is located on the rear panel of the terminal (refer to Figure 2-3)

Before connecting to electrical supply, decide whether terminal is to be operated at 220 VAC or 120 VAC If 220 volts is selected, the orientation of the line voltage card (within the terminal's fuse receptacle, see Figure 2-3) must be changed

The orientation of the Voltage Line card (within the terminal's fuse receptacle, see Figure 2-3) must be changed The edge of the card is labeled with the possible voltages The Voltage Post must be opposite to the chosen voltage Therefore, if 120 VAC is chosen, the post will be opposite the 120 VAC label (**NOTE:** The Voltage Line card is labeled 100, 120, 220, and 240 VAC - **ONLY** the 120 and 220 VAC voltages are supported!) Insert the card into the fuse receptacle with the Voltage Post **UP** Replace the fuse receptacle cover, the Voltage post should protrude slightly through the hole labeled 120V (see Figure 2-3)

Replace the fuse slot's plastic cover once the fuse is in place Next, make sure the power switch is in the "OFF" position Now connect the female end of the terminal's power cable to the terminal's electric power receptacle The terminal's power cable should now be connected to a properly grounded outlet that can supply the required power Do not use an adapter plug that prevents the terminal from being properly grounded through its power cable

2 1 3 Interfacing to Host Device

Each industrial terminal has a single RS-232C communications port (Figure 2-4)
A second serial port is available as an option (e g , options 4800-E1 thru 4800-E8)

The terminal is interfaced to the host device by installing a data cable from the terminal's communications port to the host device's RS-232C communications port Alternately, the terminal can be connected to a modem which is in turn connected to the host device

The type of cable connector required for the terminal's RS-232C communications port is a 25-pin male D-type connector See Section 6 6 for communication port pin definitions

For detailed information on the terminal's communications port, see Chapter 7

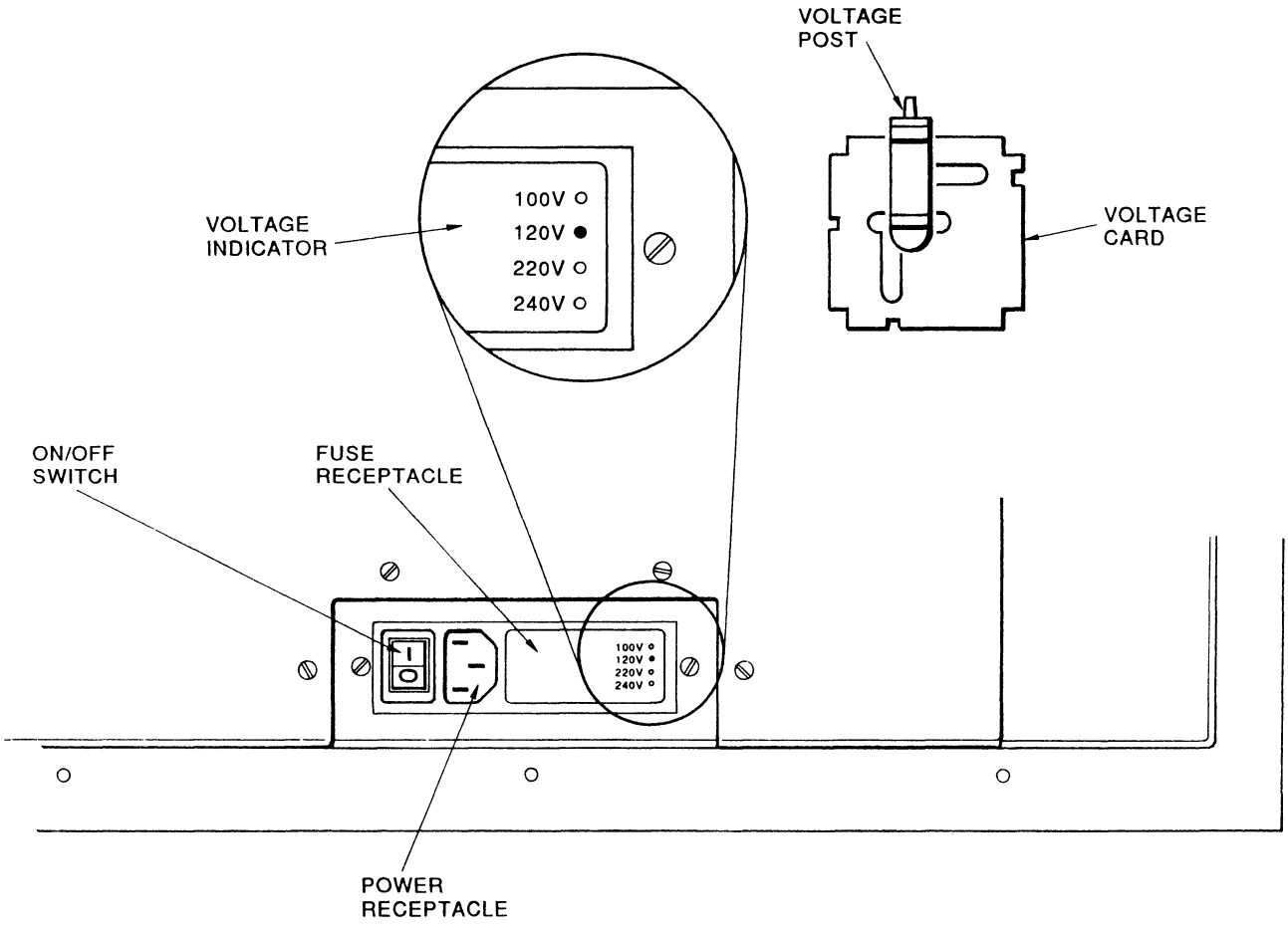


Figure 2-3 Line Voltage Selection Card

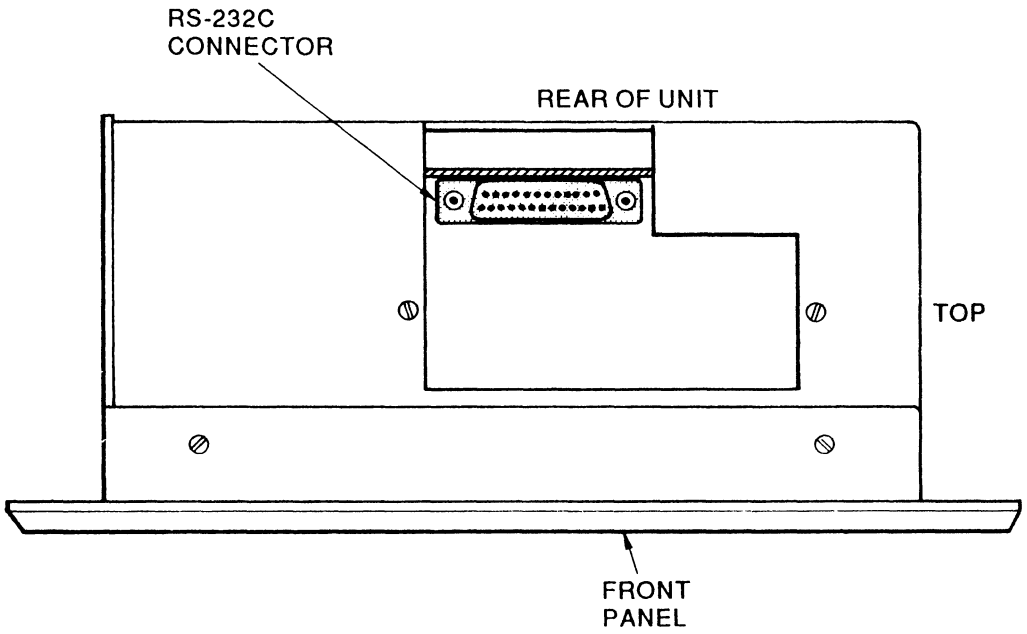


Figure 2-4 Location of RS-232C Communications Port

2 1 4 Connecting the Optional Keyboard

To install an optional full-size XYCOM 4810-KYB or IBM PC/XT[1] compatible keyboard, connect the cable that is attached to the keyboard to the connector on the right-hand corner of the terminal's front panel (Figure 2-5) You must first unscrew the protective cap

When the full-size keyboard is not being used, the protective cap should be screwed on again

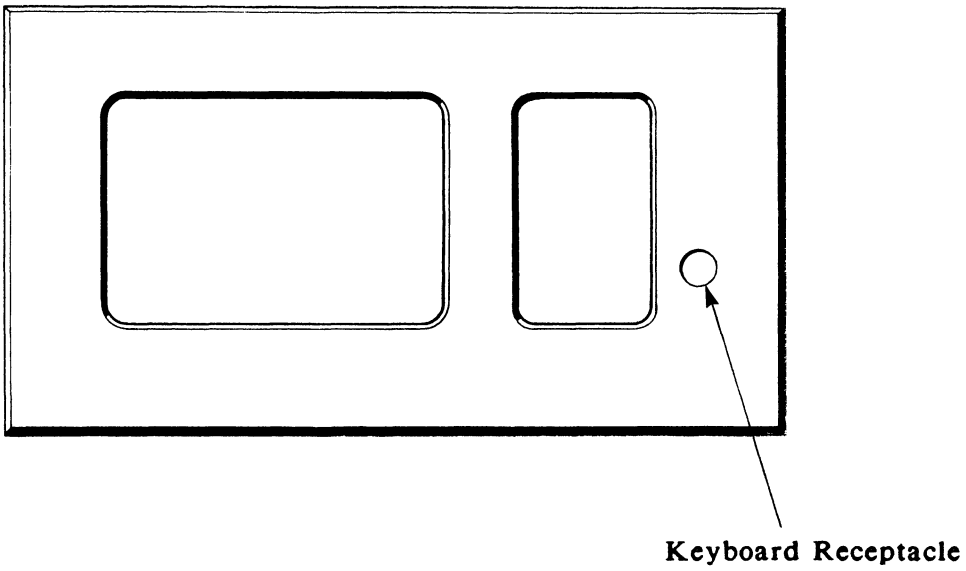


Figure 2-5 Location of Keyboard Receptacle

[1] IBM PC, and PC XT are trademarks of International Business Machines Corporation

A switch on the terminal's controller circuit board is used to specify which type (XYCOM 4810-KYB or IBM PC/XT compatible) of keyboard is to be used (see Figure 2-6) Placing the switch in the "key" position configures the terminal for operation with the 4810-KYB The "PC" position is for an IBM PC/XT keyboard The labels "Key" and "PC" are silkscreened on the controller board which is under the matrix interface card (Plugging in the wrong style keyboard for the switch position will not harm either the keyboard or the terminal -- it will simply give improper data until the switch position is corrected)

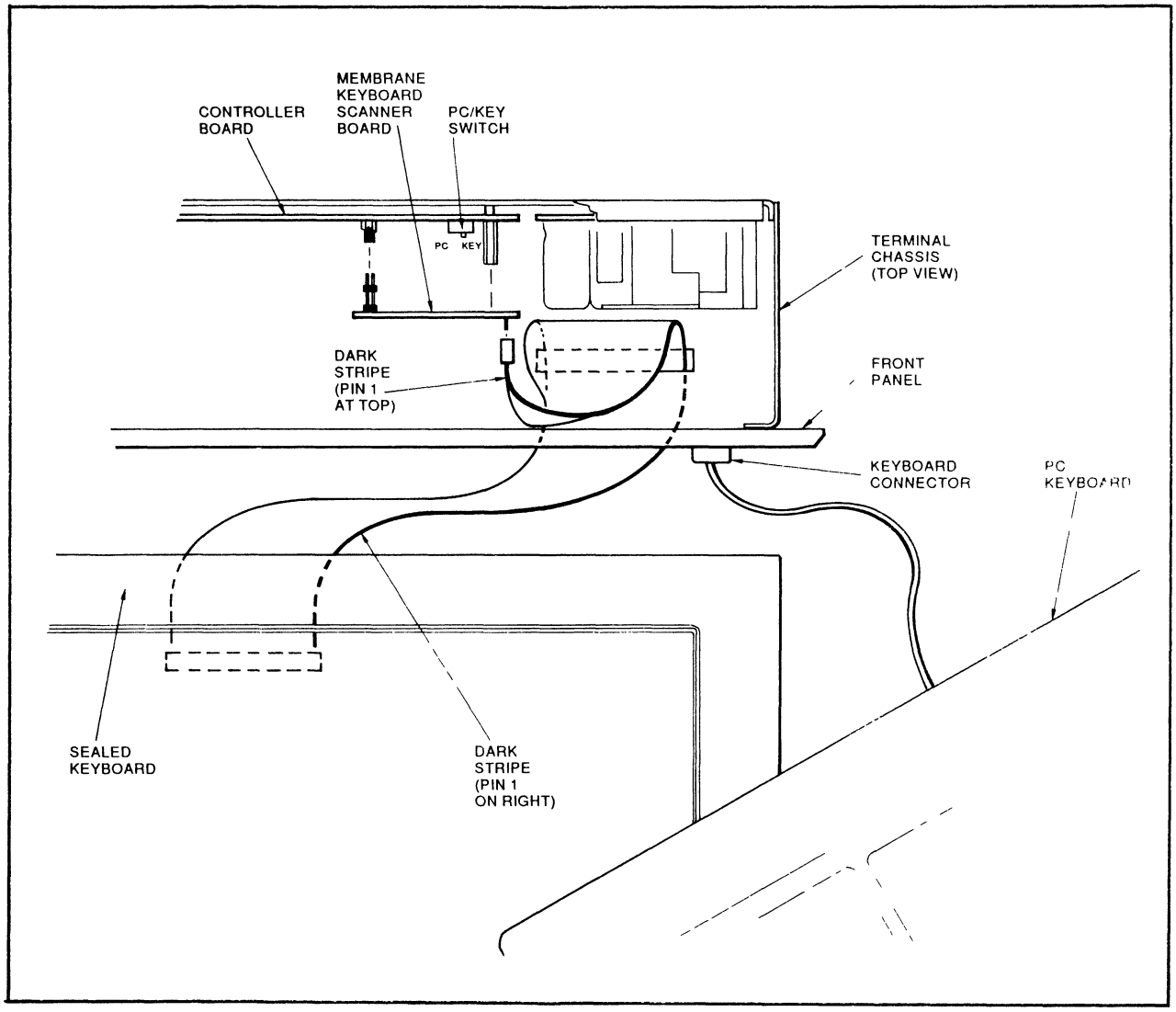


Figure 2-6 Keyboard Switch and Ribbon Connectors

An external full-sized sealed keyboard (4800-K1) is also supported A circuit card plugged into the controller card is the interface between the controller card and the external sealed keyboard

To install the 4800-K1 keyboard, a ribbon cable is connected to the back of the keyboard The cable's other end is inserted through the slot in the bottom of the terminal and connected to the interface circuit card (see Figure 2-6)

2 1 5 Power-Up

The terminal is ready to be powered up (turned on) after it is properly installed

The terminal is turned on by pressing the side of its ON/OFF switch marked "1" The terminal will run through a power-up diagnostics routine, after which time the cursor will appear in the top left corner of the screen The terminal is now in operating mode (see Section 2 2 1)

2 2 MAIN MENU

2 2 1 Operating Mode and Set-Up Mode

The terminal is always in one of two modes operating mode or set-up mode It is in operating mode whenever a menu or prompt is not being displayed on the screen It automatically enters operating mode when it is powered up The terminal must be in operating mode to receive any commands over the serial port To force the terminal into operating mode, press the <Return> key while the Main Menu is displayed (see Figure 2-7)

However, to configure the terminal, the set-up mode must be used

To exit the operating mode, do any of the following

- On the keyboard, press "F10" twice, or
- On the keypad, press "F1" and "-->" simultaneously (see Figure 2-9)

If the password has been enabled (see Section 2 2 2), the user will be prompted to type a 3-letter password Only if the password has been correctly entered will the Main Menu be displayed on the screen

2 2 2 Password and Keypad Menu Lockout

The terminal provides two ways of "tamper-proofing" the terminal's configuration a password and a keypad menu lockout (If an expansion module has been installed, the password and keypad menu lockout can also protect programs stored in the terminal's memory)

Password

Whenever you leave operating mode (see the previous section), if a password has been selected the following prompt will be displayed

Enter password (3 characters) or <RET> or <ENTER> to quit

The password consists of 3 alphanumeric characters If you type the correct 3-character sequence, the Main Menu will be displayed From the Main Menu you can reconfigure the terminal, run diagnostics, set tab stops or a new password, or edit/execute stored screen utilities (if a screen memory option has been added to the terminal) If you type an incorrect password, the terminal goes back into operating mode

The password can be changed or disabled by selecting item 3 from the Main Menu In response to the prompt "Enter new password", you can do one of the following

- To change the password, type any three alphanumeric characters on the keyboard, then press the <Return or Enter> key (the password will not be accepted until <Return or Enter> key is pressed) The password has now been changed If a character was unintentionally pressed, you may use the backspace (before the <Return or Enter> key is pressed) key to erase one or all three characters

- To disable the password, just press the <ESC> key on the keyboard, without typing any other characters (The password can be subsequently re-enabled by reselecting item 3 from the Main Menu)

- To not change the password, just press the <Return or Enter> key without typing any characters

If the password is forgotten, the user can do either of the following

- The remote command RETURN PASSWORD will return the password to the host computer (see Chapter 5)

- An Oil option board can be removed if installed, or installed if currently removed The firmware then will not recognize the password as valid If the user then goes in and stores a password, the original unknown password will be invalidated

Keypad Menu Lockout

There is an option in the Configuration Menu to lock out entry from the keypad and sealed keyboard If the keypad is locked out, the password prompt can only be invoked by pressing "F10" twice on the full-stroke keyboard that plugs into the terminal's front panel This has the effect of preventing the user from entering set-up mode from the keypad

2 2 3 Main Menu Items

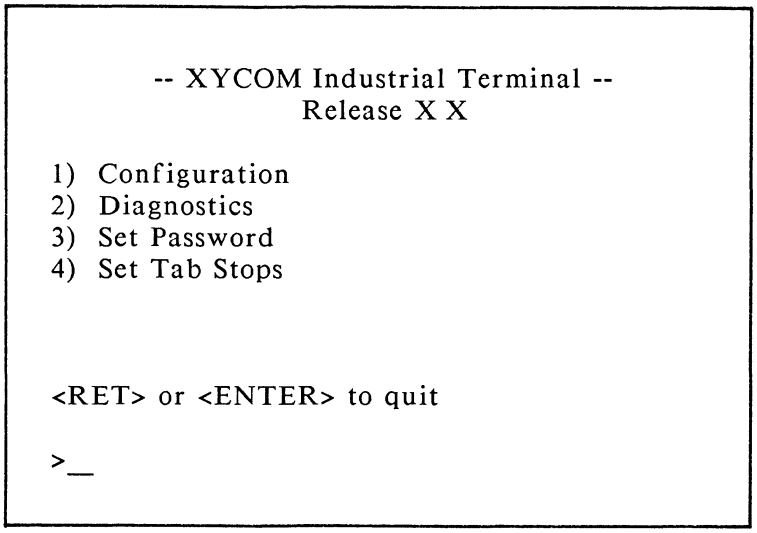


Figure 2-7 Main Menu

The Main Menu options are the following

1) Configuration -- Explained in Section 2 3
2) Diagnostics -- Explained in Chapter 7
3) Set Password -- The current password remains in effect until changed by this option (explained in Section 2 2 2)
4) Set Tab Stops -- Tab stop positions are stored in the terminal's configuration EEPROM and remain in effect even if the terminal is powered up or reset (explained in Section 2 2 4)

2 2 4 Tab Stops

If you press the "4" key when the Main Menu is displayed, the tab stops currently in effect will be displayed on the screen Note that these tab stops are in effect only when the terminal is in ANSI mode, not in Hazeltine 1500 mode In ANSI mode, by pressing the <TAB> key you will send the cursor to the next tab stop

The first row of numbers in the display are the column numbers Below these column numbers (0 to 9 eight times, or a total of 80 columns) are the tab settings An "S" below a number indicates a tab stop at that particular column position, while a blank beneath a number indicates no tab stop at that position Tab settings at any column can be entered by using the cursor keys to move the cursor under the desired column, then typing S Tab settings can be removed by moving the cursor to the desired column and typing a space Up to 80 tabs (one for each column) can be entered

If ANSI remote commands are used to change the tab stops, these changes will not be saved if the terminal is powered-down or reset The terminal will be reinitialized to the settings in the Configuration Menu

2 3 CONFIGURATION MENU

If you press the 1 key on the keyboard or keypad when the Main Menu is displayed, you will invoke the Configuration Menu

6	Baud -- 1=300 2=600 3=1200 4=2400 5=4800 6=9600 7=19 2K	
0	Parity -- 0=Zero 1=One 2=Even 3=Odd	
0	1=Parity Enabled	0=Disabled
1	1=8 Data Bits	0=7 Data Bits
1	1=Full Duplex	0=Half Duplex
1	1=Handshaking Enabled	0=Disabled
0	1=RTS/CTS Handshaking	0=Xon/Xoff Handshaking
0	0=XON/XOFF Receive 1=XON/XOFF Transmit - 2=XON/XOFF Transmit/Receive	
0	1=ANSI Emulation	0=Hazeltine 1500
0	1=Display Control Characters	0=Normal Display
1	1=Enable Auto Line Feed	0=Disable
1	1=Enable Autowrap	0=Disable
0	1=Alternate Keyboard Translation	0=Standard
0	1=Disable Scrolling	0=Enable
0	1=Block Cursor	0=Underline Cursor
0	1=Soft Scroll	0=Pop Scroll
1	1=60 Hz	0=50 Hz
0	1=Lock Menu Entry From Keypad	0=Unlock
1	Number of Status Lines	0,1,2,3,4,5
0	Keypad - 0=4x7(A) 1=4x7(F1)	

Figure 2-8 Configuration Menu

The first column of Figure 2-8 lists the current settings of all the configuration options The available options and their corresponding settings are listed to the right

To change a configuration option, first move the cursor to the row containing the value which is to be changed This is done by pressing the up-arrow and down-arrow keys When the cursor is properly positioned, press a number key to select the desired option After all changes are made, press the "ENTER" or "RETURN" key When the Main Menu is subsequently displayed, select "Return to terminal mode" by pressing the "ENTER" or "RETURN" key The terminal will then be in operating mode using the configuration options you selected

2 3 1 Considerations for Selecting Configuration Options

The first seven configuration options affect the communications characteristics of the terminal These configuration options should be set to accommodate the communications characteristics of the host device

Baud The baud rate of the terminal should be set to match the baud rate of the host device The terminal transmits and receives data at the same baud rate

Parity Should match the host computer's parity

NOTE

> Selection of disable parity is not allowed with seven data bits per character

Data Bits Per Character The number of data bits per character can be set to seven or eight, and should match the number of data bits per character used by the host device Eight data bits per character are required for use of thin-line and block graphics characters (see Sections 5 6 and 5 7)

NOTE

> Selection of seven data bits per character is not allowed with parity disabled If the terminal is configured for 7 bits and no parity, it will use 8 bits and no parity instead

Full/Half Duplex If the connected device is capable of simultaneous two-way communications and is set up for echoing, the terminal should be used in full-duplex mode If echoing is not used or the host is not capable of simultaneous two-way communications, select half-duplex mode

NOTE: When the unit is configured for half-duplex, the RTS line takes on a special function When a character is transmitted from the terminal, RTS will go high and remain high until one of the terminating characters are transmitted

<CR>	Carriage Return ASCII 13 (decimal)
<ETX>	End of Text ASCII 3 (decimal)
<EOT>	End of Transmission ASCII 4 (decimal)

If DSR is high the character is not transmitted until CTS goes high

When the termination character is transmitted, RTS will go low and remain there until the next non-termination character is transmitted

Handshaking Enabled Must be set to 1 to enable either RTS/CTS or XON/XOFF handshaking

NOTE: You can not enable handshaking and half-duplex If handshaking is enabled with half-duplex selected, the terminal will ignore handshaking and disable it when you leave the configuration menu In addition, if full-duplex is selected and handshaking disabled, RTS will always be high in ANSI mode, and after the first key is typed in Hazeltine 1500 mode after entering operational mode

RTS/CTS Handshaking, XON/XOFF

1 = RTS/CTS handshaking
0 = XON/XOFF generation

RTS/CTS Handshaking enabled Handshaking is accomplished through hardware in the following manner

RTS is an output from the terminal It will be asserted (High) when it is OK for an external device to send data to the terminal When RTS is inactive (Low), the sending unit should not attempt to send data This protects the terminal from input buffer overflow

CTS is an input to the terminal If this line is asserted (High) the terminal assumes that it is Okay to transmit data to an external device When CTS is inactive (Low) the terminal will stop transmitting data to an external device This keeps the terminal from overflowing the input buffers on an external device

XON/XOFF Handshaking enabled When this is selected the user should further define how the terminal will respond by selecting one of 3 options associated with XON/XOFF handshaking

NOTE

An XON character is DC1 ASCII 17 Decimal An XOFF character is DC3 ASCII 19 Decimal

When

0 = XON/OFF Receive

is selected, the following occurs

The terminal will <u>not</u> generate XON/XOFF characters If an XOFF character is received by the terminal, it will stop sending data until an XON is received If an XON character is received, it will assume it is okay to send data to an external device

When

1 = XON/XOFF Transmit

is selected, the following occurs

An XOFF will be sent by the terminal when the sending device should stop sending data

An XON will be sent by the terminal when it is okay for the external device to resume sending data after it has once sent an XOFF

XON/XOFF characters received by the terminal are treated like any other character

When

2 = XON/XOFF Transmit/Receive

is selected, the following occurs

An XOFF will be sent by the terminal when the sending device should stop sending data

An XON will be sent by the terminal when it is okay for the external device to resume sending data after it has once sent an XOFF

If an XOFF is received by the terminal, it will stop sending data until an XON is received

If an XON is received by the terminal, it will assume that it is okay to send data to an external device

Care should be taken when using XON/XOFF handshaking If the data stream being transmitted contains the XOFF character, you could inadvertently disable communications

Display Control Codes/Normal Display During normal operation, the terminal executes control codes that it receives such as carriage return, linefeed, etc When the Display Control Codes is selected, the terminal can be made to simply display control codes and not execute them When the terminal displays a control code, it shows a two-letter abbreviation of the ASCII control code (see Table 2-1) in a single character space Displaying control codes is useful when installing and testing communications

Table 2-1 Two-letter Abbreviations of ASCII Control Codes

Hexadecimal Code	ASCII Code	Two-letter Abbreviation
00	NUL	NL
01	SOH	SH
02	STX	SX
03	ETX	EX
04	EOT	ET
05	ENQ	EQ
06	ACK	AK
07	BEL	BL
08	BS	BS
09	HT	HT
0A	LF	LF
0B	VT	VT
0C	FF	FF
0D	CR	CR
0E	SO	SO
0F	SI	SI
10	DLE	DL
11	DC1 (XON)	D1
12	DC2	D2
13	DC3 (XOFF)	D3
14	DC4	D4
15	NAK	NK
16	SYN	SY
17	ETB	EB
18	CAN	CN
19	EM	EM
1A	SUB	SB
1B	ESC	EC
1C	FS	FS
1D	GS	GS
1E	RS	RS
1F	US	US

Block Cursor/Underline Cursor Either an underline or a block cursor can be chosen Both types of cursors are blinking The block cursor is more visible than the underline cursor

Enable Automatic Linefeed/Disable If automatic linefeed is enabled, the cursor will automatically perform a linefeed after it receives and executes a carriage return Linefeeds are ignored If disabled, only a carriage return will be executed when a carriage return is received (linefeeds are executed as linefeeds)

Alternate Keyboard Translation/Standard The term "keyboard translation" refers to the character(s) transmitted by the terminal whenever a key on the keyboard is pressed The configuration menu offers two keyboard translation options

0 = Standard
1 = Alternate

The only difference between the two translation options is in the character sequences transmitted when the following keys are pressed on the keyboard

CNTL-Q, CNTL-R, CNTL-S, CNTL-T
the four arrow keys
F1, F2, F3, F4

Table 3-5 lists the characters generated by all the above keys

All other keys are interpreted identically (see Tables 3-1, 3-2, 3-3 and 3-4)

The alternate translation option is useful with programs (such as CP/M) which expect CNTL-Q, CNTL-R, CNTL-S, or CNTL-T to produce the corresponding control codes

ANSI Emulation/Hazeltine 1500 The terminal can emulate either a Hazeltine 1500 terminal or an ANSI x3 64 terminal Hazeltine 1500 and ANSI emulation differ in the character sequences which must be transmitted to the terminal to execute a remote command For example, to perform the remote command Cursor On, a terminal configured as a Hazeltine 1500 must be sent the character sequence "7EH 02H" (or the ASCII characters ~<STX>) However, if configured as an ANSI terminal, the same command requires the character sequence "<ESC> [= 1 h" (ASCII) Chapter 5 lists both the Hazeltine 1500 and ANSI character sequences which must be transmitted to the terminal to perform any available remote command

Another difference is the character(s) transmitted when one of the cursor control keys is pressed See Table 3-5 for a list of the characters generated by the cursor control keys

ANSI emulation provides support for most DEC VT100/220 remote commands The VT100/220 commands which are supported are listed in Chapter 5, while VT100/220 commands not supported are listed in Appendix B

Enable Autowrap/Disable If autowrap is enabled, lines more than 80 characters long will wrap around to the next line If disabled, any character issued after column 80 will be printed in column 80 Autowrap is automatically enabled if Hazeltine 1500 emulation is selected

Disable Scrolling/Enable If scrolling is disabled, moving the cursor below the last line in the screen will cause the cursor to wrap to the top of the screen

Lock Menu Entry From Keypad/Unlock If menu entry is locked from the keypad and sealed keyboard, only pressing the F10 key twice on the full-stroke keyboard will exit operating mode into set-up mode If menu entry is unlocked, the menus can also be entered from the keypad and sealed keyboard (see Section 2 2 1)

Soft Scroll/Pop Scroll Soft Scroll selects a smooth, slower scroll Pop Scroll selects a line-at-a-time rapid scroll

60 Hz/50 Hz This option should be set to match the frequency of the AC power source (usually 60Hz in the U S A)

Number of Status Lines (0-5) If this menu selection is chosen, the user will have from 0 to 5 non-scrolling status lines to be maintained at the bottom of the display The full area of the screen, including the 25th line, is addressable

4x7 Keypad All versions of the keypad have 28 keys total The 4870 uses the 4x7 format layout shown in Figure 2-9

NOTE

Certain options in the configuration menu are not allowed You can select these options but when you return to the Configuration Menu they will be changed These options are

- Autowrap cannot be disabled in Hazeltine mode
- Handshaking cannot be enabled when in half-duplex

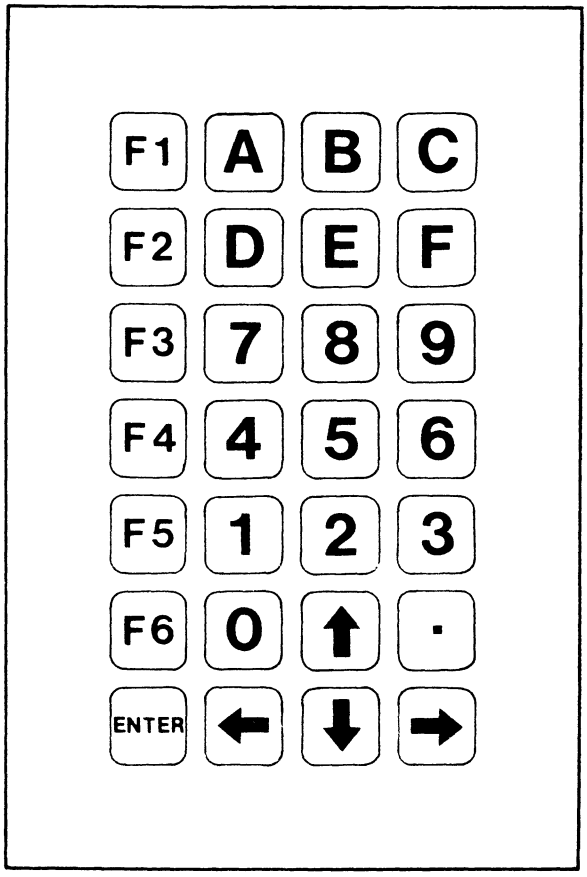


Figure 2-9 Keypad for 4870

Chapter 3

KEYPAD AND KEYBOARD

3 1 INTRODUCTION

The 4870 industrial terminal provides three ways for an operator to enter data and commands through the built-in sealed membrane keypad, through the optional sealed keyboard, or through the optional full-stroke keyboard Generally, only the sealed membrane keypad will be used when the terminal is serving as a plant-floor operator interface A full-stroke full-size keyboard is usually used for programming and supervisory operations

The keypad and keyboards generate ASCII codes when a key or combination of keys is pressed The terminal handles the codes in different ways depending on whether it's being used in full or half-duplex mode

3 2 KEYPAD AND CODES

Each terminal has a built-in sealed membrane keypad The 4870 has a 4-column, 7-row layout Refer to Figure 2-9 for keypad layout

Whenever a keypad key is pressed, the terminal will sound an audible tone to provide operator feedback The tone is not generated when the full-stroke keyboard keys are pressed

3 2 1 Application Mode

The ANSI and Hazeltine remote commands Enable Application Mode (the ANSI two-character sequence <ESC>=, and the Hazeltine sequence 7EH 2EH) and Disable Application Mode (the ANSI two-character sequence <ESC>>, and the Hazeltine sequence 7EH 2FH) determine the codes which are transmitted whenever a keypad key is pressed (see Table 3-2) At power-up or reset (and when going from set-up mode to operating mode), the keypad is automatically put in normal mode In normal mode, the keypad keys transmit codes listed in Table 3-1

NOTE

> Normal and application modes do not affect characters transmitted when a keyboard key is pressed These modes only affect keypad keys

Table 3-1 Membrane Keypad ASCII Codes

Key	Hexadecimal Code	ASCII Code	Notes
0	30	0	
1	31	1	
2	32	2	
3	33	3	
4	34	4	
5	35	5	
6	36	6	
7	37	7	
8	38	8	
9	39	9	
A	41	A	
B	42	B	
C	43	C	
D	44	D	
E	45	E	
F	46	F	
F1	47	G	
F2	48	H	
F3	49	I	
F4	4A	J	
F5	4B	K	
F6	4C	L	
	2E		
ENTER	0D	<CR>	
	Hazeltine	ANSI	
↑	7E 0C	<ESC>[A	In Hazeltine half-duplex, this key sends no code, but does move cursor on screen
↓	0A	<ESC>[B	
←	08	<ESC>[D	
→	10	<ESC>[C	In Hazeltine half-duplex, this key sends no code, but does move cursor on screen

After the remote command Enable Application Mode is executed, the keypad is put in application mode In this mode, the host computer can distinguish between keys pressed on the keypad from those pressed on the keyboard The terminal remains in application mode until a Disable Application Mode command is executed, the terminal is reset or powered-up again, or the user goes from set-up mode to operating mode In application mode, the keypad keys transmit different codes for Hazeltine 1500 or ANSI emulation If the keypad is in application mode and the terminal is configured for half-duplex, the codes produced by the keypad keys are not echoed on the screen They are only transmitted out the serial port Table 3-2 lists the codes transmitted under application mode

Table 3-2 Membrane Keypad Codes (Application Mode)
(4870)

Key	Hexadecimal Code(1) (Hazeltine 1500 Emulation)	ASCII Code(2) (ANSI Emulation)
0	B0	<ESC>Op
1	B1	<ESC>Oq
2	B2	<ESC>Or
3	B3	<ESC>Os
4	B4	<ESC>Ot
5	B5	<ESC>Ou
6	B6	<ESC>Ov
7	B7	<ESC>Ow
8	B8	<ESC>Ox
9	B9	<ESC>Oy
A	C1	<ESC>Oa
B	C2	<ESC>Ob
C	C3	<ESC>Oc
D	C4	<ESC>Od
E	C5	<ESC>Oe
F	C6	<ESC>Of
F1	C7	<ESC>Og
F2	C8	<ESC>Oh
F3	C9	<ESC>Oi
F4	CA	<ESC>Oj
F5	CB	<ESC>Ok
F6	CC	<ESC>Ol
	AE	<ESC>On
↑	91	<ESC>OA
↓	94	<ESC>OB
←	92	<ESC>OD
→	93	<ESC>OC
ENTER	8D	<ESC>OM

NOTES

(1) Same as Table 3-1, except that bit 7 is set to 1, and the arrow codes differ
(2) Same as the codes returned by a VT-100 keypad in application mode

3 2 2 Programming the Keypad Keys (4800-E1 Option Only)

If the terminal has option 4800-E1 (screen memory) installed, keypad keys can also be programmed to return a user-defined sequence whenever pressed Whenever a keypad key is pressed, the contents of a specified screen program will be transmitted See the 4800-E1 manual

3 3 KEYBOARD AND CODES

Two detachable full-size keyboards are optionally available for the industrial terminals One is a standard full-stroke keyboard, the other is a sealed keyboard Both return identical codes

NOTE

> When Menu Entry Lockout is enabled, the sealed keyboard F10 key will not respond

The ASCII codes generated by both full-size keyboards are listed in Tables 3-3, 3-4 and 3-5

Table 3-3 Codes for Keyboard Alphanumeric Keys
(Full and Half-duplex)

	no CTRL, no SHIFT		no CTRL, SHIFT		CTRL, no SHIFT		CTRL, SHIFT	
Key	Hex	ASCII	Hex	ASCII	Hex	ASCII	Hex	ASCII
A	61	a	41	A	01	<SOH>	01	<SOH>
B	62	b	42	B	02	<STX>	02	<STX>
C	63	c	43	C	03	<ETX>	03	<ETX>
D	64	d	44	D	04	<EOT>	04	<EOT>
E	65	e	45	E	05	<ENQ>	05	<ENQ>
F	66	f	46	F	06	<ACK>	06	<ACK>
G	67	g	47	G	07	<BEL>	07	<BEL>
H	68	h	48	H	08	<BS>	08	<BS>
I	69	i	49	I	09	<HT>	09	<HT>
J	6A	j	4A	J	0A	<LF>	0A	<LF>
K	6B	k	4B	K	0B	<VT>	0B	<VT>
L	6C	l	4C	L	0C	<FF>	0C	<FF>
M	6D	m	4D	M	0D	<CR>	0D	<CR>
N	6E	n	4E	N	0E	<SO>	0E	<SO>
O	6F	o	4F	O	0F	<SI>	0F	<SI>
P	70	p	50	P	10	<DLE>	10	<DLE>
Q	71	q	51	Q				
R	72	r	52	R				
S	73	s	53	S	See Table 3-5			
T	74	t	54	T				

Table 3-3 (continued)

	no CTRL, no SHIFT		no CTRL, SHIFT		CTRL, no SHIFT		CTRL, SHIFT	
Key	Hex	ASCII	Hex	ASCII	Hex	ASCII	Hex	ASCII
U	75	u	55	U	15	<NAK>	15	<NAK>
V	76	v	56	V	16	<SYN>	16	<SYN>
W	77	w	57	W	17	<ETB>	17	<ETB>
X	78	x	58	X	18	<CAN>	18	<CAN>
Y	79	y	59	Y	19	<EM>	19	<EM>
Z	7A	z	5A	Z	1A	<SUB>	1A	<SUB>
					PC/Mem	Key	PC/Mem	Key
1	31	1	21	!	31 <DC1> 11	1	21 <SH> 01	!
2	32	2	40	@	32 <DC2> 12	2	00 <NULL> 00	<NULL
3	33	3	23	#	33 <DC3> 13	3	23 <ETX> 03	#
4	34	4	24	$	34 <DC4> 14	4	24 <EOT> 04	$
5	35	5	25	%	35 <NAK> 15	5	25 <ENQ> 05	%
6	36	6	5E	^	36 <SYN> 16	6	1E <RS> 1E	<RS>
7	37	7	26	&	37 <ETB> 17	7	26 <ALK> 06	&
8	38	8	2A	*	38 <CAN> 18	8	2A <LF> 0A	*
9	39	9	28	(	39 <EM> 19	9	28 <BS> 08	(
0	30	0	29	)	30 <DLE> 10	0	29 <HT> 09	)
Backspace	08	<BS>	08	<BS>	08 <BS> 08	<BS>	08 <BS> 08	<BS>
ESC	1B	<ESC>	1B	ESC	1B <ESC> 1B	<ESC>	1B <ESC> 1B	<ESC>
SPACE	20		20		20 <NULL> 00		20 <NULL> 00	
'	27	'	22	"	27 <BEL> 07	'	22 <STX> 02	"

Table 3-3 (continued)

Key	no CTRL, no SHIFT		no CTRL, SHIFT		CTRL, no SHIFT		CTRL, SHIFT	
	Hex	ASCII	Hex	ASCII	Hex	ASCII	Hex	ASCII
					PC/Mem	Key	PC/Mem	Key
PRT SCN	2A	*	AA	*	2A <LF> 0A	*	AA <LF> 0A	*
,	2C	,	3C	<	2C <FF> 0C	,	3C <FS> 1C	<
-	2D	-	5F	_	1F <CR> 0D	<US>	1F <US> 1F	<US>
	2E		3E	>	2E <SO> 0E		3E <RS> 1E	>
/	2F	/	3F	?	2F <SI> 0F	/	3F <US> 1F	?
,	3B	,	3A		3B <ESC> 1B	,	3A <SB> 1A	
=	3D	=	2B	+	3D <GS> 1D	=	2B <VT> 0B	+
[	5B	[	7B	{	1B <ESC> 1B	<ESC>	1B <ESC> 1B	<ESC>
\	5C	\	7C	|	1C <FS> 1C	<FS>	1C <FS> 1C	<FS>
]	5D	]	7D	}	1D <GS> 1D	<GS>	1D <GS> 1D	<GS>
‘	60	‘	7E	~	60 <NULL> 00	‘	7E <RS> 1E	~
<DEL>	7F	<DEL>	2E		7F <US> 1F	<DEL>	2E <SO> 0E	

Where

PC/Mem = PC/Membrane
< > = ASCII Character
nn = Hex character

Table 3-4 Codes for Keyboard Control Keys

Key	Half-duplex Hex	Half-duplex ASCII	Full-duplex Hex	Full-duplex ASCII	Notes
TAB	09	<HT>	09	<HT>	
BACK SPACE	08	<BS>	08	<BS>	
DEL	See Table 3-3				
RETURN	0D	<CR>	OD	<CR>	
ENTER	0D	<CR>	0D	<CR>	
<left arrow>					
<right arrow>	See Table 3-5				
<up arrow>					
<down arrow>					
BREAK	00	<NUL>	00	<NUL>	(1)
ESC	1B	<ESC>	1B	<ESC>	
HOME			7E 12	~<DC2>	Haz emulation (2)(4)
			1B 5B 48	<ESC>[H	ANSI emulation (4)
F1					
F2	See Table 3-5				
F3					
F4					
F5 (CLEAR)			7E 1C	~<FS>	Haz emulation (2)(4)
			1B 4F 50	<ESC>OP	ANSI emulation (4)
F6 (CLEAR FOREGROUND)			7E 1D	~<GS>	Haz emulation (2)(4)
			1B 4F 51	<ESC>OQ	ANSI emulation (4)
F7 (CLEAR TO END OF LINE)			7E 0F	~<SI>	Haz emulation (2)(4)
			1B 4F 52	<ESC>OR	ANSI emulation (4)
F8 (CLEAR TO END OF SCREEN			7E 18	~<CAN>	Haz emulation (2)(4)
WITH FOREGROUND SPACES)			1B 4F 53	<ESC>OS	ANSI emulation (4)
F9	88		88		
F10					(3)

NOTES

(1) The communications line is held low (0) for 200-250 milliseconds
(2) This is not transmitted when in Hazeltine half-duplex mode
(3) Does not transmit a character
(4) When "4x7(A)" keypad is selected in ANSI mode, keys will respond in Hazeltine mode ANSI response not available on "4x7(A)"

Table 3-5 Cursor Control and "F" Keys on Keyboard
(Full-duplex)

Key	Standard(1) Hazeltine 1500		Alternate(1) Hazeltine 1500		Standard(1) ANSI		Alternate(1) ANSI	
	Hex	ASCII	Hex	ASCII	Hex	ASCII	Hex	ASCII
CNTL-Q	7E 0C	~<FF>	11	<DC1>	1B 5B 41	<ESC>[A	11	<DC1>
CNTL-R	08	<BS>	12	<DC2>	1B 5B 44	<ESC>[D	12	<DC2>
CNTL-S	10	<DLE>	13	<DC3>	1B 5B 43	<ESC>[C	13	<DC3>
CNTL-T	0A	<LF>	14	<DC4>	1B 5B 42	<ESC>[B	14	<DC4>
up arrow	7E 0C	~<FF>	11	<DC1>	1B 5B 41	<ESC>[A	11	<DC1>
left arrow	08	<BS>	12	<DC2>	1B 5B 44	<ESC>[D	12	<DC2>
right arrow	10	<DLE>	13	<DC3>	1B 5B 43	<ESC>[C	13	<DC3>
down arrow	0A	<LF>	14	<DC4>	1B 5B 42	<ESC>[B	14	<DC4>
F1 (2)	11	<DC1>	7E 0C	~<FF>	11	<DC1>	1B 5B 41	<ESC>[A
F2 (2)	12	<DC2>	0A	<LF>	12	<DC2>	1B 5B 42	<ESC>[B
F3 (2)	13	<DC3>	08	<BS>	13	<DC3>	1B 5B 44	<ESC>[D
F4 (2)	14	<DC4>	10	<DLE>	14	<DC4>	1B 5B 43	<ESC>[C

NOTE

(1) The difference between standard and alternate is discussed in Chapter 2, Section 2 2 5, under "Keyboard Translation"

(2) Do not confuse the keyboard keys with F1-F6 keys on the keypad

Table 3-6 Numeric Pad (with NUM LOCK Off)(1)

Key	no SHIFT Hex	no SHIFT ASCII	SHIFT Hex	SHIFT ASCII
0	B0		30	0
1	B1		31	1
2	see down arrow (Table 3-5)		32	2
3	B3		33	3
4	see left arrow (Table 3-5)		34	4
5	35	5	35	5
6	see right arrow (Table 3-5)		36	6
7(2)	see home (Table 3-4)		37	7
8	see up arrow (Table 3-5)		38	8
9	B9		39	9

NOTE

(1) With NUM LOCK on, ASCII numbers from 0 through 9 will be generated, and the SHIFT will have no effect CTRL has no effect on the numeric keypad keys

(2) In the "NO SHIFT" mode, Key 7 is "HOME"

Chapter 4

VIDEO DISPLAY

4 1 VIDEO DISPLAY FORMAT

Screen size ——— 9" diagonal

Screen color ——— Amber

Screen capacity —— 25 rows x 80 columns (standard characters)
25 rows x 40 columns (double-wide characters)
12 rows x 80 columns (double-high characters)
12 rows x 40 columns (double-wide/double-high characters)
6 rows x 16 columns (quad-size characters)

Cell size ———— 8 pixels wide by 8 pixels high

Character size —— 1 cell (regular characters)
5 cells wide by 4 cells high (quad-size characters)
4 cells wide by 4 cells high (process-control characters)

Character set ——— See Appendix C for graphic characters

Character ————	blink	quad-size
attributes	underline	double-size
	double-wide	reverse video
	double-high	

Remote commands — Variety of commands to draw boxes, and vertical or horizontal lines, and high-resolution bars (see Section 5 4)

Cursor ———— blinking underline, blinking block, or none

4 2 CURSOR ADDRESSING

The 4870 terminal provides two cursor-addressing commands Cursor To and Return Cursor Position One of these -- Cursor To X,Y -- allows the cursor to be positioned anywhere on the video display The other -- Return Cursor Position -- allows the current position of the cursor to be read

The video display has a coordinate system for cursor positioning A diagram of the coordinate system listing row and column coordinates for each possible cursor position is given in Figures 5-1 and 5-2 Row and column coordinates begin with 0 in Hazeltine mode, with 1 in ANSI mode When the cursor address is read, the system will return its column and row coordinate Those coordinates are also used to move the cursor

NOTE

> The coordinates for Hazeltine and ANSI emulation are in reverse order In Hazeltine, the column coordinate precedes the row coordinate (x, y), while in ANSI the row coordinate precedes the column (y, x)
>
> Note also that column and row coordinates are different for Hazeltine and ANSI emulation (see Figures 5-1 and 5-2)

The character sequence required to execute the Cursor To X,Y and Return Cursor Position commands depends upon whether the terminal is configured for Hazeltine or ANSI emulation Chapter 5 contains the all of the commands

4 3 STATUS AREA

The last 0-5 rows of the video display can be used as the status area The status area is not affected by the action of the normal display area (i e , it does not scroll and is not cleared when the screen is cleared) In order to write to the status area, a Cursor to X,Y remote command must be sent to the terminal to move the cursor to the status area It is also true that while positioned on the status area, remote commands will not affect the normal display area Therefore, a clear screen command executed while in the status area will only clear the status area

4 4 SCROLLING

The video display will scroll up whenever any of the following conditions exist

- the cursor is in the last character position of the bottom line of the normal display area when scrolling is enabled, autowrap is enabled, and a displayable ASCII code is received
- the cursor is in the bottom line, auto linefeed is enabled, and a carriage return <CR> code is received
- the cursor is in the bottom line, auto linefeed is disabled, and a linefeed code is received
- any "move cursor with scrolling" commands are received

When the display scrolls up, the top line of the display is removed, all lines on the display except the status line shift up one line, a blank line is added immediately above the status line, and the cursor is moved to the new line This new line consists of background spaces

Note that if scrolling is turned off (either through the Configuration Menu or a remote command), the video display will not scroll

4 5 ATTRIBUTES

The appearance of characters on the display can be enhanced by assigning attributes to characters In addition the attributes can be used to select alternate character sets or size of characters Attributes that can be assigned to characters are

blinking characters
underline
double-wide, double-high, double-sized, or quad-sized characters
reverse video
utility graphics
process graphics symbols

Attributes are assigned by sending an attribute command to the terminal immediately before the character string that is to be displayed with that attribute All characters subsequently received by the terminal will be displayed with that attribute until the assigned attribute is changed by sending a different attribute command to the terminal Chapter 5 contains all command information

4 6 CHARACTER SIZE CONSIDERATIONS

The terminal is capable of displaying five sizes of characters regular-size, double-wide, double-high, double-wide and double-high, and quad-size Different size characters can be shown on the video display simultaneously

The relative sizes of the field for the different sizes of characters are shown in Figure 4-1 Note that larger characters occupy fields that are multiples of the regular-size character field

Care must be exercised in positioning the cursor when using the larger characters This is because, in general, the cursor moves a single regular-size character field at a time The exception to this rule is when a character is being written to the video display In this case the cursor will advance the proper number of regular-size character fields automatically after the character is displayed, so that it is ready to accept another character of the same size In addition, the cursor will do a carriage return followed by the proper number of linefeeds to start a new line if a character is received when the cursor is in the last column of a line The cursor is also sensitive to character size when a linefeed, backspace, or carriage return is received When cursor movement other than a linefeed, backspace, or carriage return is attempted within a large character field, the cursor may disappear Cursor movement is explained in detail with the help of diagrams on the following pages

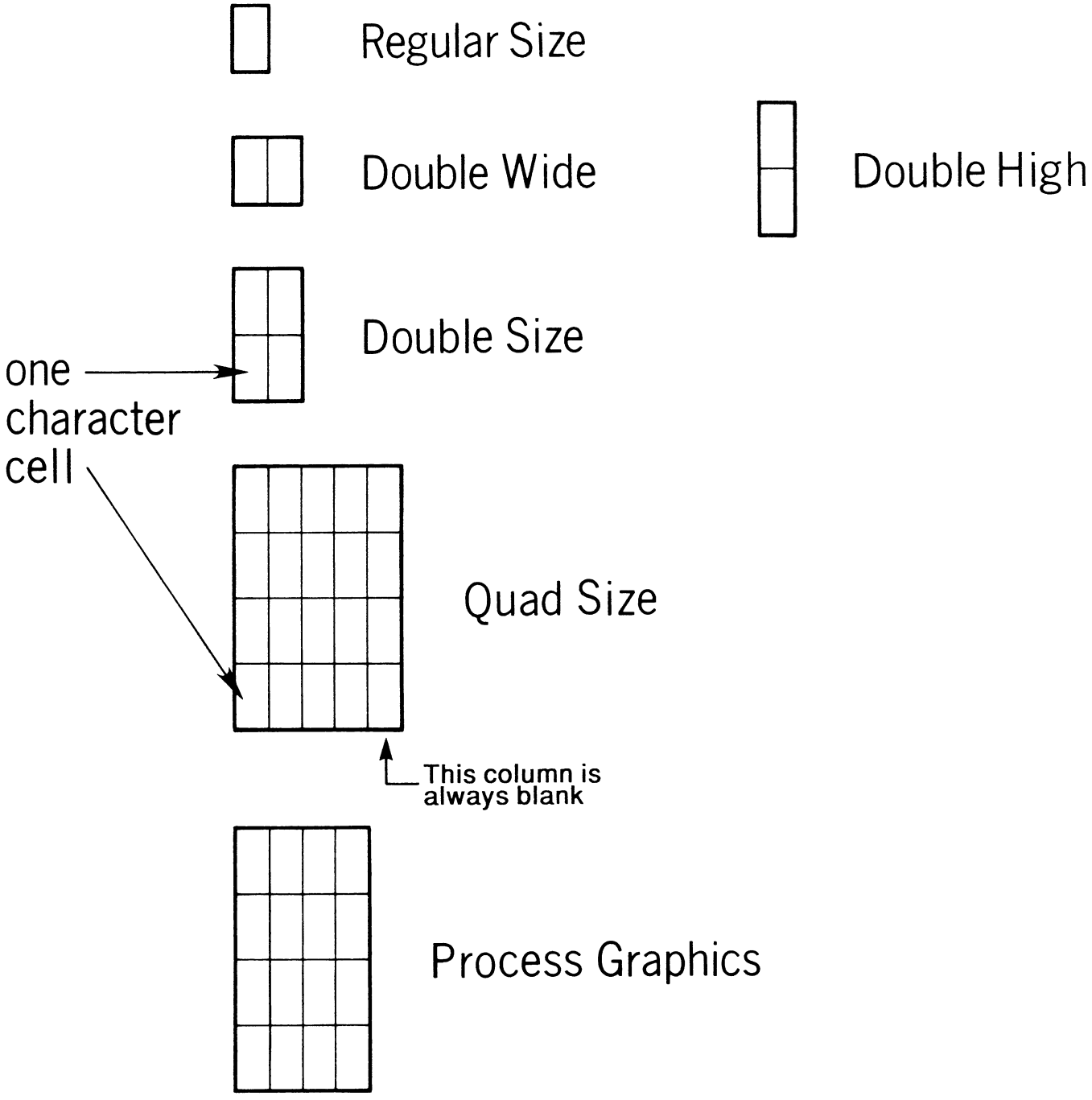
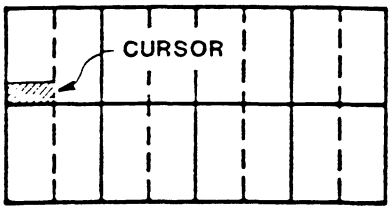
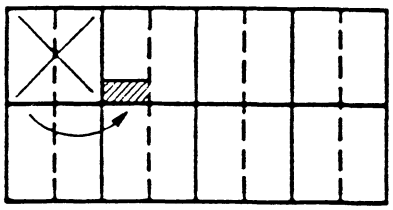


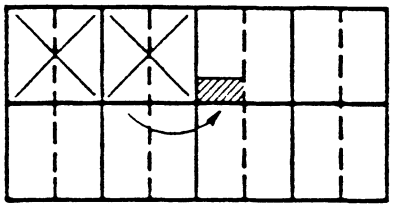
Figure 4-1 Relative Sizes of Character Fields



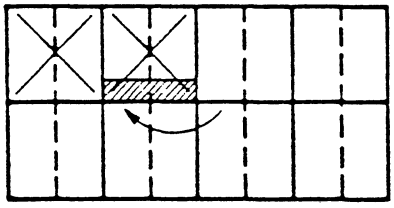
a) Character Size is Double-Wide with underline cursor positioned as shown Note that cursor underlines only half of the double-wide character field



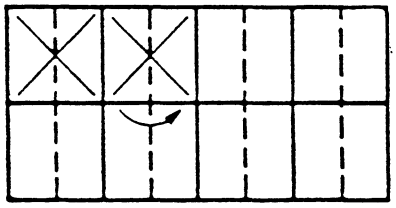
b) The character X is received and displayed and the cursor automatically advances two regular-size character fields to the next double-wide character field



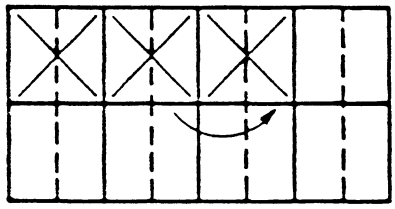
c) Another character X is received and displayed and the cursor advances again



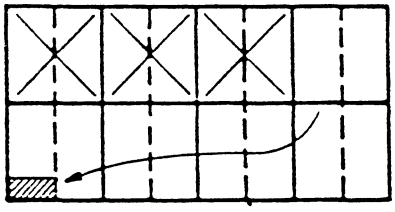
d) A single backspace command has been received and executed so the cursor is "pointing" to the left-hand regular-size character field of the second X, but now the cursor appears double-wide beneath the second X



e) If a cursor right command is received and executed, the cursor moves to the position indicated and disappears



f) If another X were received with the cursor located as in (e), the X would be displayed and the cursor would still be invisible



g) If a carriage return and a linefeed or only a carriage return with automatic linefeed enabled is received, the cursor moves to the start of the first double-wide character field on the next line

See Diagrams, Figure 4-2

a) Character Size is Double-High with underline cursor positioned as shown Note that the cursor is in the middle of the first double-high character field

b) The character X is received and displayed and the cursor automatically advances one regular-size character field to the middle of the next double-high character field

c) Another character X is received and displayed and the cursor advances again

d) Here, a single backspace command has been received and executed and the cursor moves left one regular-size character field to the middle of the second X

e) If a cursor right command is received and executed, the cursor moves back to the position it was at in (c)

f) Another character X is received and displayed and the cursor advances

g) If a carriage return and a linefeed, or only a carriage return with automatic linefeed enabled is received, the cursor moves to the next line and is in the proper position to receive another character

h) Another character X is received and displayed, and the cursor advances

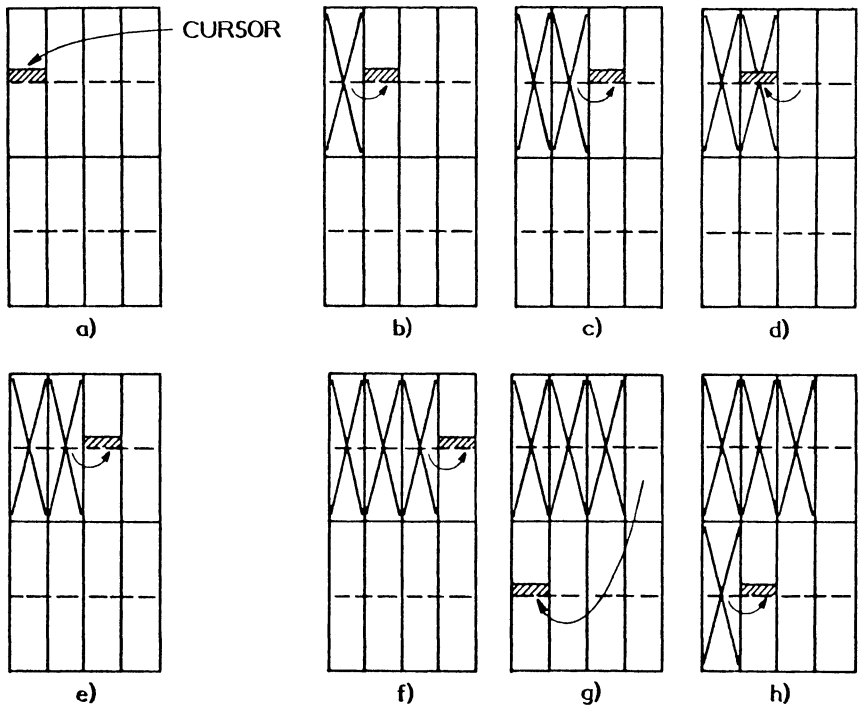


Figure 4-2 DOUBLE HIGH

See Diagrams, Figure 4-3

a) Character size is double-high/double-wide with underline cursor positioned as shown The cursor is in the middle of the double-high/double-wide character field and underlines only the first regular-size character field

b) The character X is received and displayed and the cursor automatically advances two regular-size character fields to the middle of the next double--high/double-wide character field

c) Another character X is received and displayed and the cursor advances again

d) Single backspace command has been received and executed so the cursor is "pointing" to the upper left-hand regular-size character field of the second X

e) If a cursor right command is received and executed and the cursor moves to the position indicated at (e) and disappears

f) If another X were received with the cursor located as in (e), the X would be displayed and the cursor would still be invisible

g) If a carriage return and a linefeed or only a carriage return with automatic linefeed enabled is received, the cursor moves to the proper position to write the next row of double-high/double-wide characters In this position, the cursor underlines only the upper left-hand regular-size character field

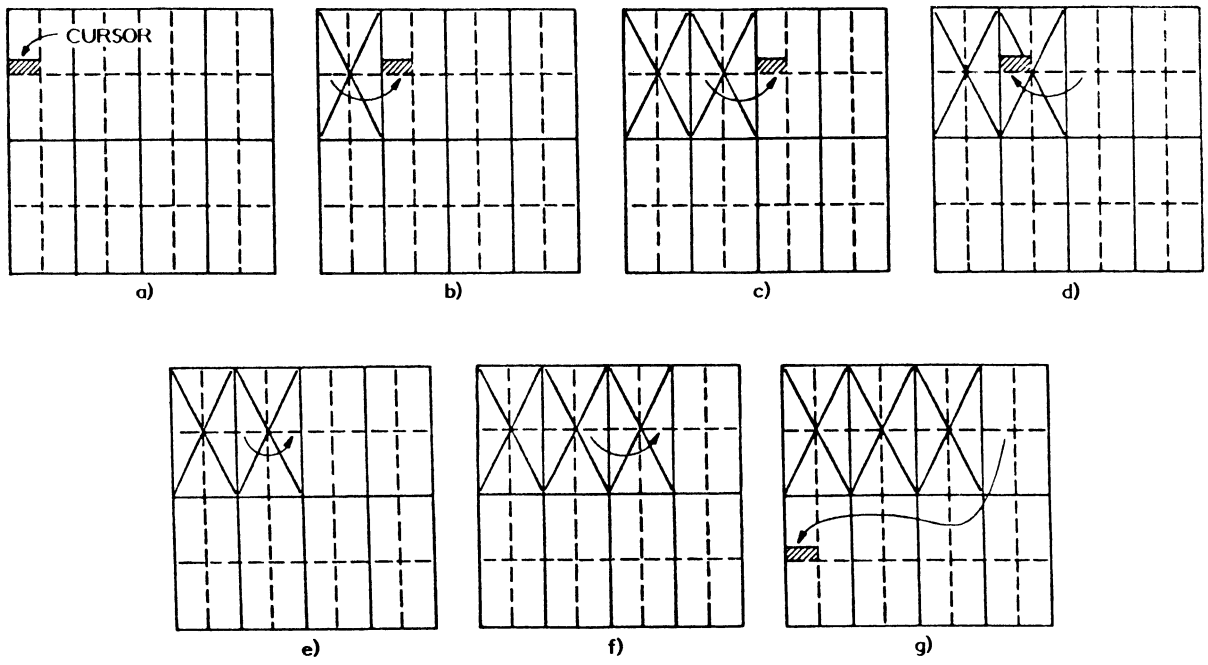


Figure 4-3 DOUBLE HIGH/DOUBLE WIDE Character Cursor Movement

See Diagrams, Figure 4-4

a) Character size is Quad-Size with underline cursor starting in position 1

The character X is received and displayed and the cursor automatically advances to the start of the next quad-size character at 2 To move the cursor back to position 1 requires one backspace command To move the cursor from 1 to 2 without writing a character requires five cursor right commands

When moving within a quad-size character field, the cursor is always visible and remains a regular-size character

b) To move the cursor from position 2 to the start of the first quad-size character field at 3, type a carriage return and linefeed, or only a carriage return with automatic linefeed enabled will move the cursor to 3

If another displayable character code were received with the cursor within a displayed quad-size character, the new character would overwrite all or a portion of the existing character depending on the position of the cursor

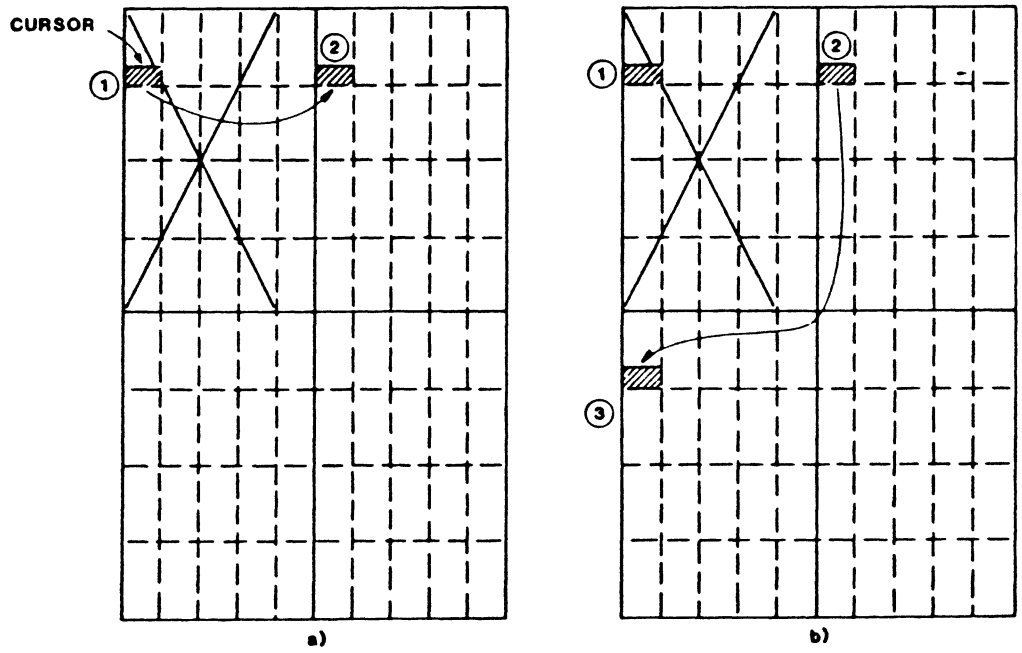


Figure 4-4 QUAD SIZE Character Cursor Movement

4 7 THIN-LINE GRAPHICS

The terminal can display 16 different thin-line graphics characters (regular size and double wide only) These characters can be used, for example, to display diagrams on the video display To display thin-line graphics characters, character set bits 0-2 of attribute byte No 2 must be set to 000 (specifying regular characters)

The diagrams in Appendix C, character codes 128-143 decimal (80-8F hexadecimal), show the appearance of the 16 thin-line graphics characters

NOTE

> The terminal must be configured for 8 bits/character in order to display these via codes from the serial port

4 8 BLOCK GRAPHICS

The terminal can display 16 different regular-size and double wide block graphics characters These characters can be used in combination, for example, to display diagrams and characters that are larger than quad-size Appendix C contains character codes 144-159 decimal (90-9F hexadecimal), and shows all the block graphics characters

Each block graphics character is made up of pixels The different block graphics characters are made up by turning on different combinations of these pixels

NOTE

> The terminal must be configured for 8 bits per character in order for these characters to be displayed via codes from the serial port

4 9 SPECIAL BAR GRAPHICS CHARACTERS

Special graphic characters are provided to draw solid character cells of specified heights and widths These special bar graphics characters are regular or double size (bits 0-2 of attribute byte No 2 must be set to 000) There are four types of special bar graphics characters

- Vertical bar up characters, which draw bars of varying heights, all beginning at the bottom of the character cell See Appendix C, characters codes 209 - 216 decimal (D1 - D8 hexadecimal)

- Vertical bar down characters, which begin at the top of the character cell and extend downwards See Appendix C, character codes 225 - 232 decimal (E1 - E8 hexadecimal)

- Horizontal bar right characters, which begin at the left edge of the character cell and extend to the right See Appendix C, character codes 240 - 247 decimal (F0 - F7 hexadecimal)

- Horizontal bar left characters, which begin at the right edge of the character cell and extend to the left See Appendix C, character codes 248 - 255 decimal (F8 - FF hexadecimal)

4 10 PROCESS GRAPHIC SYMBOLS

If the Character Set bits 0-2 of attribute byte No 2 are zero when a displayable character is typed, the character printed on the key is displayed on the terminal screen (These bits are automatically set to the default value 0 whenever the terminal is powered up or reset) However, process graphic symbols can be selected by setting the Character Set bits to the following value

Attribute Byte No 2			
Bit 2	Bit 1	Bit 0	Attribute
0	1	1	process graphic symbols

If process graphic symbols are selected, the character transmitted by the terminal will not change (see Chapter 3) However, certain characters codes sent to the display will cause graphic symbols to be displayed For example, in process graphics mode, typing an uppercase "M" will still cause the character 4DH (hex value of "M") to be transmitted, but if an "M" is received, a small box instead of "M" will be displayed

Table 4-1 lists all the process graphic symbols and the characters which will generate each one

Appendix C shows what these symbols look like

Table 4-1 Process Graphic Symbols

Hex Value	ASCII Character	Process Control Symbol
20H		4x4 space
21H	!	motor in 4x3 cell
22H	"	not used
23H	#	left tank top in 4x1 cell
24H	$	right tank top in 4x1 cell
25H	%	small diamond in 4x2 cell
26H	&	left tank bottom in 4x1 cell
27H	'	right tank bottom in 4x1 cell
28H	(	left arrow in 4x2 cell

Table 4-1 Process Graphic Symbols (CONT'D)

Hex Value	ASCII Character	Process Control Symbol
29H	)	right arrow in 4x2 cell
2AH	*	small box in 4x2 cell
2BH	+	up valve in 4x2 cell
2CH	,	right/left facing valve in 4x2 cell
2DH	-	pump/compressor in 4x2 cell
2EH		up arrow in 4x2 cell
2FH	/	down arrow in 4x2 cell
30H	0	small circle in 4x2 cell
31H	1	circuit breaker type 1 in 2x4 cell
32H	2	fuse in 2x4 cell
33H	3	disconnect in 3x4 cell
34H	4	pump/blower in 4x2 cell
35H	5	circuit breaker type 2 in 4x2 cell
36H	6	left turbine in 3x2 cell
37H	7	right turbine in 3x2 cell
38H	8	left medium box in 4x2 cell
39H	9	right medium box in 4x2 cell
3AH	,	left medium circle in 4x3 cell
3BH		right medium circle in 4x3 cell
3CH	<	mini circle in 2x1 cell
3DH	=	mini left arrow in 2x1 cell
3EH	>	mini right arrow in 2x1 cell
3FH	?	mini up arrow in 2x1 cell
40H	@	mini down arrow in 2x1 cell
41H	A	motor
42H	B	large circle (left)
43H	C	large circle (right)
44H	D	tank top (left)
45H	E	tank top (right)
46H	F	small diamond
47H	G	large diamond (left)
48H	H	large diamond (right)
49H	I	tank bottom (left)
4AH	J	tank bottom (right)
4BH	K	left arrow
4CH	L	right arrow
4DH	M	small box
4EH	N	up valve
4FH	O	right/left facing valve
50H	P	pump/compressor
51H	Q	up arrow
52H	R	down arrow
53H	S	small circle
54H	T	transformer
55H	U	circuit breaker (type 1)

Table 4-1 Process Graphic Symbols (CONT'D)

Hex Value	ASCII Character	Process Control Symbol
56H	V	fuse
57H	W	disconnect
58H	X	pump/blower
59H	Y	circuit breaker (type 2)
5AH	Z	turbine (left)
5BH	[	turbine (right)
5CH	\	large box (left)
5DH	]	large box (right)
5EH	^	medium box (left)
5FH	_ (underscore)	medium box (right)
60H	` (grave)	medium circle (left)
61H	a	medium circle (right)
62H	b	top left quarter of large circle in 4x2 cell
63H	c	top right quarter of large circle in 4x2 cell
64H	d	bottom left quarter of large circle in 4x2 cell
65H	e	bottom right quarter of large circle in 4x2 cell
66H	f	top left quarter of small circle in 2x1 cell
67H	g	top right quarter of small circle in 2x1 cell
68H	h	bottom left quarter of small circle in 2x1 cell
69H	i	bottom right quarter of small circle in 2x1 cell
6AH	j	small tank top in 4x1 cell
6BH	k	small tank bottom in 4x1 cell
6CH	l	mini tank top in 2x1 cell
6DH	m	mini tank bottom in 2x1 cell
6EH	n	mini diamond in 2x1 cell
6FH	o	mini box in 2x1 cell
70H	p	mini right valve in 2x1 cell
71H	q	mini up valve in 2x1 cell
72H	r	mini motor in 2x2 cell
73H	s	mini pump/blower in 2x1 cell
74H	t	mini transformer in 2x2 cell
75H	u	mini circuit breaker type 1 in 1x2 cell
76H	v	mini fuse in 1x2 cell
77H	w	mini disconnect in 1x2 cell
78H	x	mini blower/compressor in 2x1 cell
79H	y	mini circuit breaker type 2 in 2x1 cell
7AH	z	mini left turbine in 1x1 cell
7BH	{	mini right turbine in 1x1 cell

If the terminal is in process graphic mode and a character not in the above table is typed or received, nothing will be displayed

The process graphic symbols are shown in Appendix C

4 11 UTILITY GRAPHICS

If the character set bits (bits 0-2 of attribute byte No 2) are set to the value 111, the terminal will be in utility graphics mode In this mode, receiving certain alphabetic characters will cause pieces of process control symbols to be displayed The terminal uses these pieces to construct the process graphics symbols You may be able to use these pieces to construct your own graphics, or to connect process graphic characters

Note that this mode affects only the character/symbol displayed when certain character codes are sent to the display It does not change the character transmitted by the terminal when the key is pressed (see Chapter 4)

Table 4-2 describes the Utility Graphics available

Table 4-2 Utility Graphics

Character Codes	Graphics Description
32-79 (20-4FH)	Process Graphics Pieces
80-87 (50-57H)	Process Graphic Connectors (Thin)
88-95 (58-5FH)	Process Graphic Connectors (Thick)
96-111 (60-6FH)	Thick Line Graphics
112-175 (70-AFH)	Process Graphic Pieces
176-187 (B0-BBH)	Miscellaneous Connectors

Appendix C shows what the characters look like

4 12 GRAPHIC SHADING CHARACTERS

The shading characters can be used to create varying shades of gray (or texture) as used in bar chart shading

Appendix C, character codes 221, 222 and 237, 238 decimal (DD, DE and ED, EE hexadecimal), shows the shading graphic symbols

Chapter 5

REMOTE COMMANDS

5 1 INTRODUCTION

Remote commands allow the terminal to be controlled by the host device Remote commands require lead-in character(s) to be received by the terminal immediately before the command code is received In Hazeltine 1500 emulation, the lead-in character is ~ (7EH), called a tilde In ANSI emulation, the lead-in character is ESC (1BH), or the two-character sequence ESC [(1BH 5BH) The lead-in code does not affect the display when received by the terminal

If the code following the lead-in character is not a valid command code requiring a lead-in character, both the lead-in character and the code that follows it will be ignored by the terminal

Configuration changes performed with remote commands are not saved when the terminal is turned off or reset

NOTE

> If a remote command has been issued to change the current configuration, the new configuration is lost on power-down or reset To save the new configuration, it is only necessary to enter the Configuration Menu, then exit it (without having to change the Configuration Menu) This will save the new configuration in EEPROM, and consequently upon power-up or reset, the new configuration will be brought up

See Table 5-1 for a list of remote commands your Terminal can receive from a host device

5 2 HAZELTINE 1500 EMULATION

For a detailed description of some commands, see Section 5 4 Cursor addressing commands are briefly described in Section 4 2

In Table 5-1, parameters such as <attr-1> or <xstart> are single bytes in the range 00H through FFH

Table 5-1 Remote Commands
(Hazeltine 1500 Emulation)

Commands	ASCII	Hex
Control Characters		
Bell	<BEL>	07
Backspace	<BS>	08
Cursor to Next Foreground Field	<HT>	09
Linefeed	<LF>	0A
Carriage Return	<CR>	0D
Configuration Commands		
Enable Application Mode	~	7E 2E
Disable Application Mode	~ /	7E 2F
Cursor Off	~<S0H>	7E 01
Cursor On	~<STX>	7E 02
Scrolling Off	~<BEL>	7E 07
Scrolling On	~<BS>	7E 08
Unlock Keyboard	~<ACK>	7E 06
Lock Keyboard	~<NAK>	7E 15
Attribute Commands		
Set/Reset Attributes	~6<attribute#>*	7E 36 <attribute #>
Change Char Attributes	~<ETX> <attr-1> <attr-2>	7E 03 <attr-1> <attr-2>
Cursor Movement Commands		
Cursor Right (no scroll)	<DLE>	10
Return Cursor Position	~<ENQ>	7E 05
Cursor Down (no scroll)	~<VT>	7E 0B
Cursor Up	~<FF>	7E 0C
Cursor to X,Y	~<DC1> X Y	7E 11 X Y
Home Cursor	~<DC2>	7E 12

Table 5-1 Remote Commands (continued)
(Hazeltine 1500 Emulation)

Commands	ASCII	Hex
Clear Commands		
Clear to EOL with Background Spaces	~<SI>	7E 0F
Clear to EOS with Background Spaces	~<ETB>	7E 17
Clear to EOS with Foreground Spaces	~<CAN>	7E 18
Clear Foreground	~<GS>	7E 1D
Clear Screen	~<FS>	7E 1C
Background Field Follows	~<EM>	7E 19
Foreground Field Follows	~<US>	7E 1F
Delete Commands		
Delete Line	~<DC3>	7E 13
Insert Line	~<SUB>	7E 1A
Draw Commands		
Draw Box	~<HT> <char> <xstart> <ystart> <xend> <yend>	7E 09 <char> <xstart> <ystart> <xend> <yend>
Draw Vertical Line (upward)	~<LF> <char> <xstart> <ystart> <length>	7E 0A <char> <xstart> <ystart> <length>
Draw Horizontal Line (left to right)	~<CR> <char> <xstart> <ystart> <length>	7E 0D <char> <xstart> <ystart> <length>
Draw Bar Chart	~<S0> <xstart> <ystart> <length1> <length2>	7E 0E <xstart> <ystart> <length1> <length2>
Draw Bar Chart Down	~<space> <xstart> <ystart> <length1> <length2>	7E 20 <xstart> <ystart> <length1> <length2>
Draw Bar Chart Right	~! <xstart> <ystart> <length1> <length2>	7E 21 <xstart> <ystart> <length1> <length2>
Draw Bar Chart Left	~" <xstart> <ystart> <length1> <length2>	7E 22 <xstart> <ystart> <length1> <length2>
Additional Commands		
Pause	~ # <time>	7E 23 <time>
Return Password	~ %	7E 25
Plot Point	~0XY	7E 30 X Y
Unplot Point	~1XY	7E 31 X Y
* See Section 5 4 32		

5 3 ANSI EMULATION

In ANSI mode, the parameters are one or more ASCII characters Most parameters are numbers, with characters in the range 30H (the character "0") through 39H (the character "9")

NOTE

> In ANSI emulation, if the decimal value of a numeric parameter is greater than 9, two characters are necessary For example, the decimal number 10 is represented as "1" followed by "0" (hex value 31 30) Likewise, if the decimal value is greater than 99, three characters are necessary
>
> ANSI values must be between 0-255

In ANSI mode, parameters are separated by a semicolon, and all characters except ESC are displayable ASCII characters

Appendix F provides a table which converts between ASCII, hexadecimal, and decimal values

Table 5-2 Remote Commands (ANSI Emulation)

Control Characters

00 - ignored
05 - answer-back request
07 - ring bell
08 - move cursor left 1 position
09 - go to next tab stop
0A - linefeed or new line
0B - same as 0A
0C - same as 0A
0D - move cursor to left margin of current line (carriage return)
18 - cancel current ESC sequence
1A - same as 18
1B - ESC

Configuration Commands (See Notes 2 and 3)

ESC [? 7 h - enable autowrap
ESC [? 25 h - cursor on
ESC [? 7 l - disable autowrap
ESC [? 25 l - cursor off
ESC [? 4 h - smooth scroll
ESC [? 4 l - pop scroll

ESC [2 h - lock keyboard
ESC [2 l - unlock keyboard
ESC [20 h - enable auto line-feed
ESC [20 l - disable auto line-feed

ESC [= 1 h - cursor on
ESC [= 2 h - scrolling on
ESC [= 3 h - treat tab as ANSI tab
ESC [= 1 l - cursor off
ESC [= 2 l - scrolling off
ESC [= 3 l - treat tab as Hazeltine tab

Attribute Commands (See Note 1)

ESC [m - attributes off
ESC [0 m - attributes off
ESC [4 m - underline
ESC [5 m - blink
ESC [7 m - reverse video on
ESC [24 m - underline disable
ESC [25 m - blink disable
ESC [27 m - reverse video off

Table 5-2 Remote Commands (continued)
(ANSI Emulation)

Attribute Commands (continued) (See Note 1)

ESC [50 m - select regular character set
ESC [51 m - select double-high characters
ESC [52 m - select quad-sized characters
ESC [53 m - select process control symbols
ESC [54 m - select double-wide characters
ESC [55 m - select double-size characters
ESC [56 m - select quad-sized characters
ESC [57 m - select utility graphics
ESC [1 ,attr1,attr2 p - change character attributes

Cursor Movement Commands

ESC [pn A - cursor up pn lines
ESC [pn B - cursor down pn lines
ESC [pn C - cursor right pn characters
ESC [pn D - cursor left pn characters
ESC [y,x H - cursor to position x,y
ESC [H - cursor home (1,1)
ESC [y,x f - cursor to position x,y
ESC [f - cursor home (1,1)
ESC D - cursor down with scroll
ESC M - cursor up with scroll
ESC E - cursor to beginning of next line with scroll
ESC 7 - save cursor and attributes
ESC 8 - restore cursor and attributes

Tab Stop Commands (See Note 4)

ESC H - set tab stop at current column
ESC [g - clear tab stop at current column
ESC [0 g - clear tab stop at current column
ESC [3 g - clear all tab stops

Clear Commands

ESC [pn X - clear pn characters on current line with background spaces
ESC [K - clear to end of line with background spaces
ESC [? K - clear to end of line with background spaces
ESC [0 K - clear to end of line with background spaces
ESC [? 0 K - clear to end of line with background spaces
ESC [1 K - clear to beginning of line with background spaces
ESC [? 1 K - clear to beginning of line with background spaces
ESC [2 K - clear entire line with background spaces
ESC [? 2 K - clear entire line with background spaces
ESC [J - clear to end of screen with background spaces

Table 5-2 Remote Commands (continued)
(ANSI Emulation)

Clear Commands (continued)

```
ESC [ ? J       - clear to end of screen with background spaces
ESC [ 0 J       - clear to end of screen with background spaces
ESC [ ? 0 J     - clear to end of screen with background spaces
ESC [ 1 J       - clear to beginning of screen with background spaces
ESC [ ? 1 J     - clear to beginning of screen with background spaces
ESC [ 2 J       - clear entire screen with background spaces
ESC [ ? 2 J     - clear entire screen with background spaces
ESC [ 8 p       - clear to end-of-screen with foreground spaces
ESC [ 9 p       - background follows
ESC [ 10 p      - clear foreground
ESC [ 11 p      - foreground follows
```

Insert/Delete Commands

```
ESC [ pn L      - insert pn blank line(s) at current cursor position
ESC [ pn M      - delete pn line(s) from cursor position
ESC [ pn @      - insert pn space(s) in line at cursor position
ESC [ pn P      - delete pn character(s) from line at cursor position
```

Report Commands

```
05 <ENQ>        - answer-back
                  device returns message - XYCOM TERMINAL followed by 4 spaces
ESC [ 5 n       - device status report
                  device ok returns - ESC [ 0 n
                  device not ok returns - ESC [ 3 n
ESC [ 6 n       - report cursor x,y position
                  returns - ESC [ y,xR
ESC [ c         - return options
ESC [ 0 c       - return options
                  returns - ESC [ ? 1,0c
```

Additional Commands

```
ESC c               - reset to initial state
ESC =               - select application mode for keypad keys
ESC >               - select normal mode for keypad keys
ESC b               - unlock keyboard
ESC '               - lock keyboard
ESC [ 18,time p     - pause
ESC [ 20 p          - return password
```

Table 5-2 Remote Commands (continued)
(ANSI Emulation)

Draw Commands

ESC [2 ,char,ystrt,xstrt,yend,xend p	- draw box
ESC [3 ,char,ystrt,xstrt,length p	- draw vertical line
ESC [4 ,char,ystrt,xstrt,length p	- draw horizontal line
ESC [5 ,ystrt,xstrt,len1,len2 p	- draw bar chart up
ESC [25,ycor,xcor p	- plot point
ESC [26,ycor,xcor p	- unplot point
ESC [15,ystrt,xstrt,len1,len2 p	- draw bar chart down
ESC [16,ystrt,xstrt,len1,len2 p	- draw bar chart right
ESC [17,ystrt,xstrt,len1,len2 p	- draw bar chart left

NOTES

(1) Multiple attributes can be selected in a single attribute command
ESC [50,40,31m

(2) Multiple configurations can be specified in a single configuration command Example
ESC [= 1,2,3 h
ESC [? 7,25 h
ESC [2,20 h

(3) Configuration options that can be set by both the remote commands and the Configuration Menu are not saved on power-down unless the Configuration Menu is entered and exited

(4) Tab stops set/reset with remote commands are not saved on power-down unless the "Set Tab Stop" menu is entered and exited

5 3 1 VT100/220 Support

When the terminal is configured for ANSI mode, it emulates the DEC VT100 and VT220 terminals Some VT100/220 commands are not handled by the terminal On the other hand, some commands not supported by the VT100/220 are available on the terminal

The VT100/220 functions not emulated are listed below

- 132 column mode is not supported
- not all special function keys are supported
- transmit and receive baud rates are not independent and there are fewer available baud rates
- no split screen capability
- different set-up procedure for configuration
- no user controllable LEDs
- no margin bell, key click
- optional DEC character sets and graphics are not supported
- VT52 mode is not supported
- can not invoke confidence tests remotely
- line attributes (double-high, double-wide) supported differently
- application mode supported only on the keypad, not on the keyboard
- insert mode not supported

When codes for these functions are received, they are ignored

Appendix B lists the VT100/220 codes not supported by the terminal

5 4 THE AVAILABLE REMOTE COMMANDS

Most of the remote commands listed in Table 5-1 are self-explanatory However, some of the commands require further information, which is presented below, and some of the commands will affect the terminal's configuration options which are discussed in Chapter 2

NOTE

> All commands may be entered either in hex or in ASCII, both for Hazeltine 1500 and ANSI emulation However, hex is typically used in Hazeltine emulation, and ASCII is usual for ANSI emulation Therefore, the remote commands for Hazeltine emulation are presented in hex in this chapter, while the remote commands for ANSI emulation are presented in ASCII

Foreground and Background Fields

The following commands are related to the foreground and background fields on the terminal screen

Cursor to Next Foreground Field
Clear to EOL with Background Spaces
Clear to EOS with Background Spaces
Clear to EOS with Foreground Spaces
Background Field Follows
Clear Foreground
Foreground Field Follows
Clear Screen
Clear Line
Clear to the Beginning of a Line
Clear to the Beginning of a Screen
Clear Characters on a Line

The terminal allows you to define foreground and background fields

Data in a foreground field is displayed in reverse video, data in a background field is displayed in normal video All data is displayed as either a foreground or background field, depending upon how the field has been most recently defined The default value is background field, so unless a field has been specifically defined as a foreground field, it will be background

Foreground and background fields may be useful in distinguishing areas on the screen, for example column title fields, from fields in which data should be entered by the user

5 4 1 **Cursor to Next Foreground Field**

Function — Moves the cursor to the first character in the next foreground field In ANSI emulation, the TAB character (09H) defaults to "Cursor To Next Tab Stop" The ANSI user must issue ESC[=3l to use tabs to move the cursor to the next foreground field

Hazeltine emulation	09H	
ANSI emulation	<ESC> [= 3 l	(treat tab as Hazeltine tab (and the last character is a lower case "L"))
	<HT>	

where ESC = 1BH

NOTE

> The command to treat a TAB as a Hazeltine TAB only needs to be issued once All subsequent tabs will seek foreground fields In order to use ANSI TAB stops the remote command <ESC>[= 3h will need to be sent (treat TABs as ANSI TABs Remote Command)

5 4 2 Clear to EOL with Background Spaces

Function — All characters from the current cursor position to the end of the line are cleared to spaces In addition, all character positions from the current cursor position to end of line are defined as a background field

Hazeltine emulation — 7EH 0FH
ANSI emulation — <ESC> [K

where ESC = 1BH

5 4 3 Clear to EOS with Background Spaces

Function — All characters from the current cursor position to the end of the scrolled screen are cleared to spaces In addition, all character positions from the current cursor position to end of the screen are defined as a background field If in the Status Area, it clears to the end of the screen

Hazeltine emulation — 7EH 17H
ANSI emulation — <ESC> [J

where ESC = 1BH

5 4 4 Clear to EOS with Foreground Spaces

Function — All characters from the current cursor position to the end of the scrolled screen are cleared to spaces In addition, all character positions from the current cursor position to the end of screen are defined as a foreground field If in the Status Area, it clears to the end of the screen

Hazeltine emulation — 7EH 18H
ANSI emulation — <ESC> [8 p

where ESC = 1BH

5 4 5 **Background Field Follows**

Function — All subsequent data is displayed as a background field, until a Foreground Field Follows command is executed

Hazeltine emulation — 7EH 19H
ANSI emulation — <ESC> [9 p

where ESC = 1BH

5 4 6 **Clear Foreground**

Function — All foreground fields on the entire screen are replaced by foreground spaces, and the cursor is moved to the first position of the first foreground field

Hazeltine emulation — 7EH 1DH
ANSI emulation — <ESC> [10 p

where ESC = 1BH

5 4 7 **Foreground Field Follows**

Function — All subsequent data is displayed as a foreground field, until a Background Field Follows command is executed

Hazeltine emulation — 7EH 1FH
ANSI emulation — <ESC> [11 p

where ESC = 1BH

5 4 8 **Clear Screen** (with Background Spaces)

Function — All characters and data are cleared from the display screen

Hazeltine emulation — 7EH 1C
ANSI emulation — <ESC> [2 J

where ESC = 1BH

5 4 9 **Clear Line** (with Background Spaces)

Function All characters are cleared from the current line the cursor is on

Hazeltine emulation N/A
ANSI emulation <ESC> [K

where ESC = 1BH

5 4 10 **Clear to Beginning of the Line** (with Background Spaces)

Function All characters are cleared from the current cursor position to the beginning of the current line

Hazeltine emulation N/A
ANSI emulation <ESC> [1 K

where ESC = 1BH

5 4 11 **Clear to the Beginning of the Screen** (with Background Spaces)

Function All characters are cleared from the current cursor position to the beginning of the screen

Hazeltine emulation N/A
ANSI emulation <ESC> [1 J

where ESC = 1BH

5 4 12 **Clear Characters on a Line** (with Background Spaces)

Function All specified number of characters are cleared on the current line

Hazeltine emulation N/A
ANSI emulation <ESC> [pn X

where ESC = 1BH
pn = the number of characters to be cleared

The following example first defines some background fields, leaving the foreground fields blank Then it homes the cursor and proceeds to fill the previously defined foreground fields with data

ANSI

Command	Comments
"<ESC> [2J"	,Clear Screen
"Weld Station "	,Message on Screen
"<ESC> [11p"	,Foreground Field Follows
"<SPACE>"	,Blank character, required to establish the rest ,of the line as foreground field
"<CR>"	,Carriage Return
"<LF>"	,Linefeed
"<ESC>[9p"	,Background Field Follows
"STATUS "	,Message on Screen
"<ESC>[11p"	,Foreground Field Follows
"<SPACE>"	,Blank character
"<CR>"	,Carriage Return
"<LF>"	,Linefeed
"<ESC>[8p"	,Clear to end of Screen with Foreground Spaces
"<ESC>[3 1"	,Treat TABs as Hazeltine TABs
"<TAB>"	,Cursor to Next Foreground Field
"Carriage Assembly - Left"	,Message on Screen
"<TAB>	,Cursor to Next Foreground Field
"Not Operational"	,Message on Screen
"<TAB>"	,Cursor to Next Foreground Field
"Over Current Detected"	,Message on Screen
"<ESC> = 3H"	

Hazeltine 1500

Command	Comments
"7EH 1CH"	, Clear Screen
"WELD STATION "	, Message on Screen
"7EH 1FH"	, Foreground Field Follows
"<SPACE>"	, Blank character, required to establish rest of , line as foreground field
"<CR>, <LF>"	, Carriage Return, Linefeed
"7EH 19H"	, Background Field Follows
"STATUS "	, Message on Screen
"7EH 1FH"	, Foreground Field Follows
"<SPACE>"	, Blank character
"<CR>, <LF>"	, Carriage Return, Linefeed
"7EH 18H"	, Clear to EOS with Foreground Spaces
"7EH 12H"	, Home Cursor
"09H"	, Cursor to Next Foreground Field
"Carriage Assembly - Left"	, Message on Screen
"09H"	, Cursor to Next Foreground Field
"Not Operational"	, Message on Screen
"09H"	, Cursor to Next Foreground Field
"Overcurrent Detected"	, Message on Screen

5 4 13 **Draw Box**

Function Draws a box The coordinates of the upper left and lower right corners are included in the character sequence

Hazeltine emulation 7EH 09H <char> <xstart> <ystart> <xend> <yend>
ANSI emulation <ESC>[2,<char>,<ystart>,<xstart>,<yend>,<xend>p

where char -- Hazeltine emulation
01H = thick-line box
02H = thin-line box
03H = thin-line box using utility graphics characters
04H = thick-line box using utility graphics characters
Any displayable ASCII character = box composed of that character
-- ANSI emulation
1(31H) = thick-line box
2(32H) = thin-line box
3(33H) = thin-line box using utility graphics characters
4(34H) = thick-line box using utility graphics characters
Sequence of two ASCII decimal characters = box composed of the ASCII equivalent of the decimal value For example, to draw a box composed of the character "A"(65) the following two characters are required 6(36H) and 5(35H)

xstart = x coordinate of upper left corner of box
ystart = y coordinate of upper left corner of box
xend = x coordinate of lower right corner of box
yend = y coordinate of lower right corner of box

Note This command will not cause automatic scrolling if a box the size of the screen is drawn

5 4 14 Draw Vertical Line in Upward Direction

Function — Draws an upward vertical line, beginning at the coordinate included in the command sequence, toward the screen's top edge

Hazeltine emulation 7EH 0AH <char> <xstart> <ystart> <length>
ANSI emulation <ESC>[3,<char>,<ystart>,<xstart>,<length>p

where char -- Hazeltine emulation
01H = thick line
02H = thin line
03H = thin right of cell connector (utility graphic 51H)
04H = thin left of cell connector (utility graphic 53H)
05H = thick right of cell connector (utility graphic 59H)
06H = thick left of cell connector (utility graphic 5BH)
Any displayable ASCII character = line composed of that character

-- ANSI emulation
1(31H) = thick line
2(32H) = thin line
3(33H) = thin right of cell connector (utility graphic 51H)
4(34H) = thin left of cell connector (utility graphic 53H)
5(35H) = thick right of cell connector (utility graphic 59H)
6(36H) = thick left of cell connector (utility graphic 5BH)

Sequence of two ASCII decimal characters = line composed of the ASCII equivalent of the decimal value For example, to draw a line composed of the character "A"(65) the following two characters are required 6(36H) and 5(35H)

xstart = x coordinate of start of line
ystart = y coordinate of start of line
length = length of line (in units of character cells)

5 4 15 **Draw Horizontal Line from Left to Right**

Function — Draws a horizontal line (from left to right) starting at the coordinate in the character sequence, toward the right edge of the screen

Hazeltine emulation 7EH 0DH <char> <xstart> <ystart> <length>
ANSI emulation <ESC>[4,<char>,<ystart>,<xstart>,<length>p

where char -- Hazeltine emulation
01H = thick line
02H = thin line
03H = thin top of cell connector (utility graphic 50H)
04H = thin bottom of cell connector (utility graphic 52H)
05H = thick top of cell connector (utility graphic 58H
06H = thick bottom of cell connector (utility graphic 5AH)
Any displayable ASCII character = line composed of that character

-- ANSI emulation
1(31H) = thick line
2(32H) = thin line
3(33H) = thin top of cell connector (utility graphic 50H)
4(34H) = thin bottom of cell connector (utility graphic 52H)
5(35H) = thick top of cell connector (utility graphic 58H)
6(36H) = thick bottom of cell connector (utility graphic 5AH)

Sequence of two ASCII decimal characters = line composed of the ASCII equivalent of the decimal value For example, to draw a line composed of the character "A"(65) the following two characters are required 6(36H) and 5(35H)

xstart = x coordinate of start of line
ystart = y coordinate of start of line
length = length of line (in units of character cells)

5 4 16 **Draw Bar Up**

Function Draws a high-resolution vertical bar one character wide The coordinate of the bottom character cell of the bar and its height are included in the character sequence This command includes a character specifying the height of the bar to be erased before the new bar is drawn, so that bars can be updated dynamically

Hazeltine emulation 7EH 0EH <xstart> <ystart> <length1> <length2>
ANSI emulation <ESC>[5,<ystart>,<xstart>,<length1>,<length2>p

where xstart = x coordinate of start of bar
ystart = y coordinate of start of bar
length1 = height of column (in units of 1/8 of a character cell)
The height must be in the range 0 through 200 8 is equivalent to the height of one character, 168 is equal to the height of 21 characters
length2 = Before the new vertical bar is drawn, a blank bar of length2 is drawn This erases the previous bar If length2 is zero, no blank line will be drawn

5 4 17 **Draw Bar Down**

Function Same as Draw Bar Up, except that bar is drawn downward, and <xstart> and <ystart> specify the top character cell of the bar

Hazeltine emulation 7EH 20H <xstart> <ystart> <length1> <length2>
ANSI emulation <ESC>[15,<ystart>,<xstart>,<length1>,<length2>p

where xstart = x coordinate of start of bar
ystart = y coordinate of start of bar
length1 = height of column (in units of 1/8 of a character cell)
The height must be in the range 0 through 200 8 is equivalent to the height of one character, 168 is equal to the height of 21 characters
length2 = Before the new vertical bar is drawn, a blank bar of length2 is drawn This erases the previous bar If length2 is zero, no blank line will be drawn

5 4 18 **Draw Bar Right**

Function Same as Draw Bar Up, except that bar is drawn rightward, and <xstart> and <ystart> specify the left end character cell of the bar

Hazeltine emulation 7EH 21H <xstart> <ystart> <length1> <length2>
ANSI emulation <ESC>[16,<ystart>,<xstart>,<length1>,<length2>p

where xstart = x coordinate of start of bar
ystart = y coordinate of start of bar
length1 = width (in units of 1/8 of a character cell)
The width must be in the range 0 through 255 8 is equivalent to the width of one character, 248 is equal to the width of 31 characters
length2 = Before the new horizontal bar is drawn, a blank bar of length2 is drawn This erases the previous bar If length2 is zero, no blank line will be drawn

NOTE: For the Draw Bar Right and Draw Bar Left commands, a length of 248 will only be 31 characters wide (there are 80 characters in a line) Multiple bars must be used to span more than 31 characters cells

5 4 19 **Draw Bar Left**

Function Same as Draw Bar Right, except that bar is drawn leftward, and <xstart> and <ystart> specify the right end character cell of the bar

Hazeltine emulation 7EH 22H <xstart> <ystart> <length1> <length2>
ANSI emulation <ESC>[17,<ystart>,<xstart>,<length1>,<length2>p

where xstart = x coordinate of start of bar
ystart = y coordinate of start of bar
length1 = width of column (in units of 1/8 of a character cell)
The width must be in the range 0 through 255 8 is equivalent to the width of one character, 248 is equal to the width of 31 characters
length2 = Before the new horizontal bar is drawn, a blank bar of length2 is drawn This erases the previous bar If length2 is zero, no blank line will be drawn

5 4 20 **Pause**

Function Causes the terminal to pause for a specified period before retrieving and displaying the next character or command from the serial port (or screen program, if the 4800-E1 option is installed)

Hazeltine emulation 7EH 23H<time>
ANSI emulation <ESC>[18,<time>p

where ESC = 1BH
time = duration of pause (in tenths of a second)

5 4 21 **Enable Application Mode**

Function Puts the keypad into application mode (see Section 3 1 1) The keypad remains in application mode until the terminal is reset, powered-up again, or goes from set-up mode to operating mode

Hazeltine emulation 7EH 2EH
ANSI emulation <ESC> =

where ESC = 1BH

5 4 22 **Disable Application Mode**

Function Returns the keypad from application mode to normal mode (see Section 3 1 1)

Hazeltine emulation 7EH 2FH
ANSI emulation <ESC> >

where ESC = 1BH

5 4 23 **Cursor Off** (See Note 1)

Function Makes the cursor invisible

Hazeltine emulation 7EH 01H
ANSI emulation <ESC>[=1l

(these last two characters are the digit "1" followed by a lower case "L")

where ESC = 1BH

5 4 24 **Cursor On**

Function Makes the cursor visible

Hazeltine emulation 7EH 02H
ANSI emulation <ESC>[=1h

where ESC = 1BH

5 4 25 **Scrolling Off**

Function Disables screen scrolling Any keystroke or serial input that would normally cause the screen to scroll will instead cause the cursor to go to the top of the screen

Hazeltine emulation 7EH 07H
ANSI emulation <ESC>[=2l

(the last character is a lower case "L")

where ESC = 1BH

5 4 26 **Scrolling On**

Function Enables screen scrolling Used to re-enable scrolling after the Scrolling Off command has been used to disable scrolling

Hazeltine emulation 7EH 08H
ANSI emulation <ESC>[=2h

where ESC = 1BH

NOTE

Scrolling can also be turned on or off in the Configuration Menu The value set with the Scrolling On or Scrolling Off remote commands is not saved at power-up or reset Instead, the setting in the Configuration Menu is used to set scrolling on/off

---

NOTE:

(1) Cursor is always on when terminal enters operating mode (from set-up or at power-up)

5 4 27 **Insert Line**

Function Inserts a line (or lines) immediately before the current line, and moves the cursor to the beginning of the inserted line

Hazeltine emulation 7EH 1AH
ANSI emulation <ESC>[pn L

where pn is the number of blank lines to insert
ESC = 1BH

5 4 28 **Delete Line**

Function Deletes the line on which the cursor is positioned

Hazeltine emulation 7EH 13H
ANSI emulation <ESC>[pn M

where pn is the number of lines to delete
ESC = 1BH

5 4 29 **Plot Point**

Function Turns on one pixel Each character cell consists of eight pixels

Hazeltine emulation 7EH 30H <x><y>
ANSI emulation <ESC>[25,<y>,<x>p

where ESC = 1BH
x is the horizontal coordinate (0-159)
y is the vertical coordinate (0-71)
(Note that the lower left-hand corner has coordinates 0,0)

5 4 30 **Unplot Point**

Function Turns off one pixel If the specified pixel is not on, this command has no effect

Hazeltine emulation 7EH 31H <x><y>
ANSI emulation <ESC>[26,<y>,<x>p

where ESC = 1BH
x is the horizontal coordinate (0-159)
y is the vertical coordinate (0-71)

(Note that the lower left-hand corner has coordinates 0,0)

5 4 31 Return Password

Function Returns the current password (three characters), followed by a carriage return If the password is disabled (from the Password Menu -- see Chapter 2, Section 2 2 2), only a carriage return is transmitted

Hazeltine emulation 7EH 25H
ANSI emulation <ESC>[20p

where ESC = 1BH

5 4 32 Set/Reset Attribute Command

Function Sets/resets the terminal's attributes

Hazeltine emulation 7EH 36H xx
ANSI emulation <ESC>[<dd> <m>

where ESC = 1BH
xx = the attribute set/reset hazeltine code
dd = the attribute set/reset ANSI code

dd	xx
0	(00H) - attributes off
4	(04H) - underscore on
5	(05H) - blink on
7	(07H) - reverse video on
24	(18H) - underscore off
25	(19H) - blink off
27	(1BH) - reverse video off
50	(32H) - select regular characters
51	(33H) - select double-high characters
52	(34H) - select quad-size characters
53	(35H) - select process graphics characters
54	(36H) - select double-wide characters
55	(37H) - select double-size characters
56	(38H) - select quad-size characters
57	(39H) - select utility graphics

5 4 33 **Return Cursor Position**

Function To read the cursor position, transmit a Return Cursor Position command to the terminal

Hazeltine emulation	7EH 05H
ANSI emulation	<ESC>[6 n

Hazeltine

The terminal will then transmit the response

<column coordinate> <row coordinate> CR

where <column coordinate> will be a hex value between 20H-4FH and 60H-7FH, while <row coordinate> will be a hex value between 60H and 78H CR is the ASCII character corresponding to 0D (hex)

Figure 5-2 lists the row and column coordinates under Hazeltine emulation

ANSI

The terminal will then transmit the response

ESC [<row>,<column>R (ASCII character)

where the row and column are ASCII decimal values (hex values between 30H and 39H) For example, if the cursor is currently in row 12, column 6, the Return Cursor Position command will return the following sequence of ASCII characters

ESC [12, 06 R (ASCII character)

Figure 5-1 lists the row and column coordinates under ANSI emulation

5 4 34 **Cursor to X,Y**

Function To move the cursor to column x, row y on the screen

Hazeltine emulation 7EH 11H <x> <y>
ANSI emulation <ESC> [y,x H

where ESC = 1BH
y = row
x = column

Hazeltine

To move the cursor to column x, row y, transmit a Cursor to X,Y command to the terminal

7EH 11H xx yy

where xx and yy are the hexadecimal equivalents of the decimal values x,y (e g , position 19=13 hex), and <char x> and <char y> are the ASCII characters corresponding to the hex values xx and yy (e g , ASCII DC3 corresponds to 13 hex)

Figure 5-2 lists the row and column coordinates under Hazeltine emulation

ANSI

To move the cursor to row y, column x, transmit a Cursor to X,Y command to the terminal

ESC [y,x H

Note that the decimal coordinate grater than 9 must be expressed as <u>two</u> decimal ASCII characters For example, decimal coordinate 10 is expressed as "1" followed by "0" (31H 30H)

Figure 5-1 lists the row and column coordinates under ANSI emulation

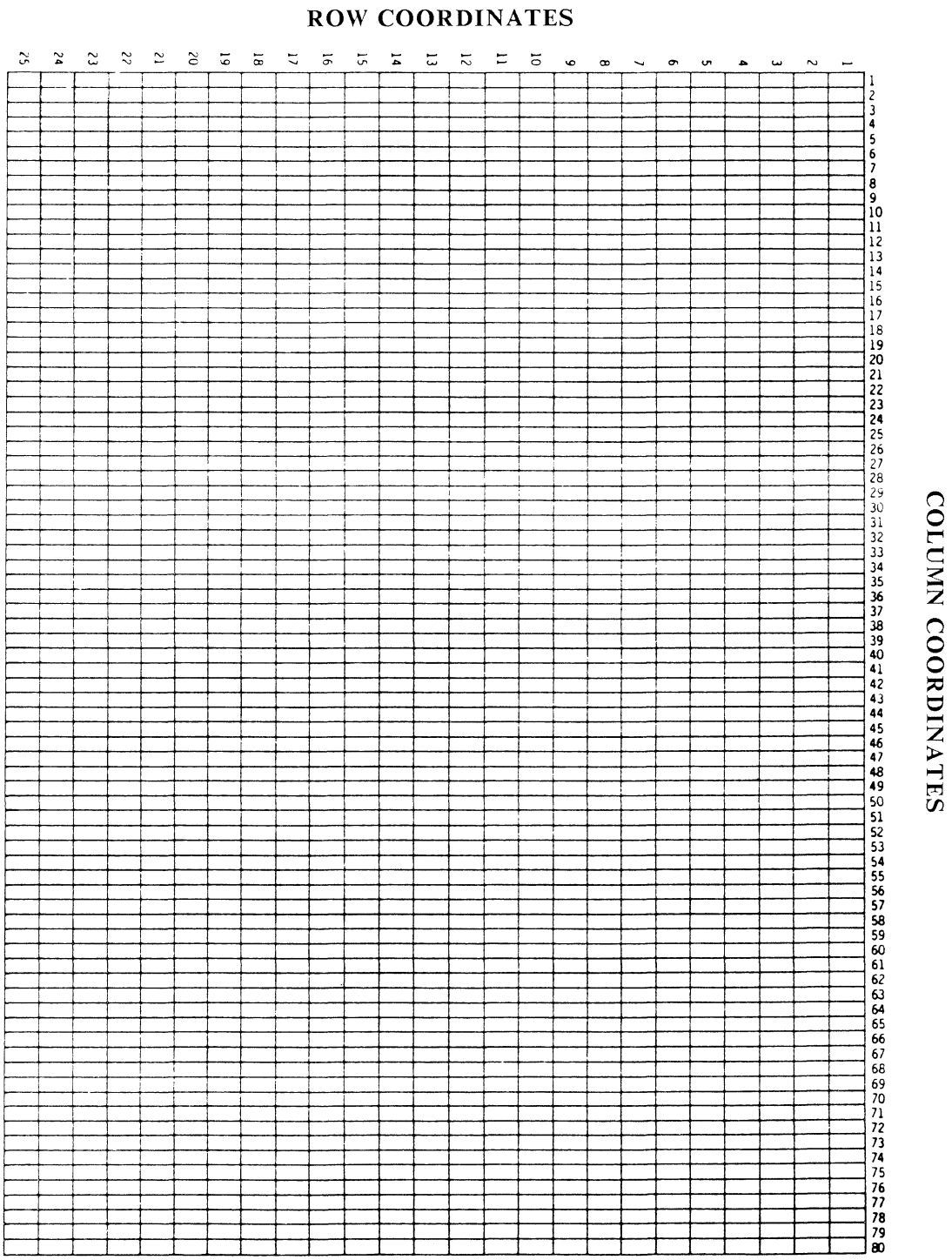


Figure 5-1 Video Display Coordinate System (ANSI Emulation)

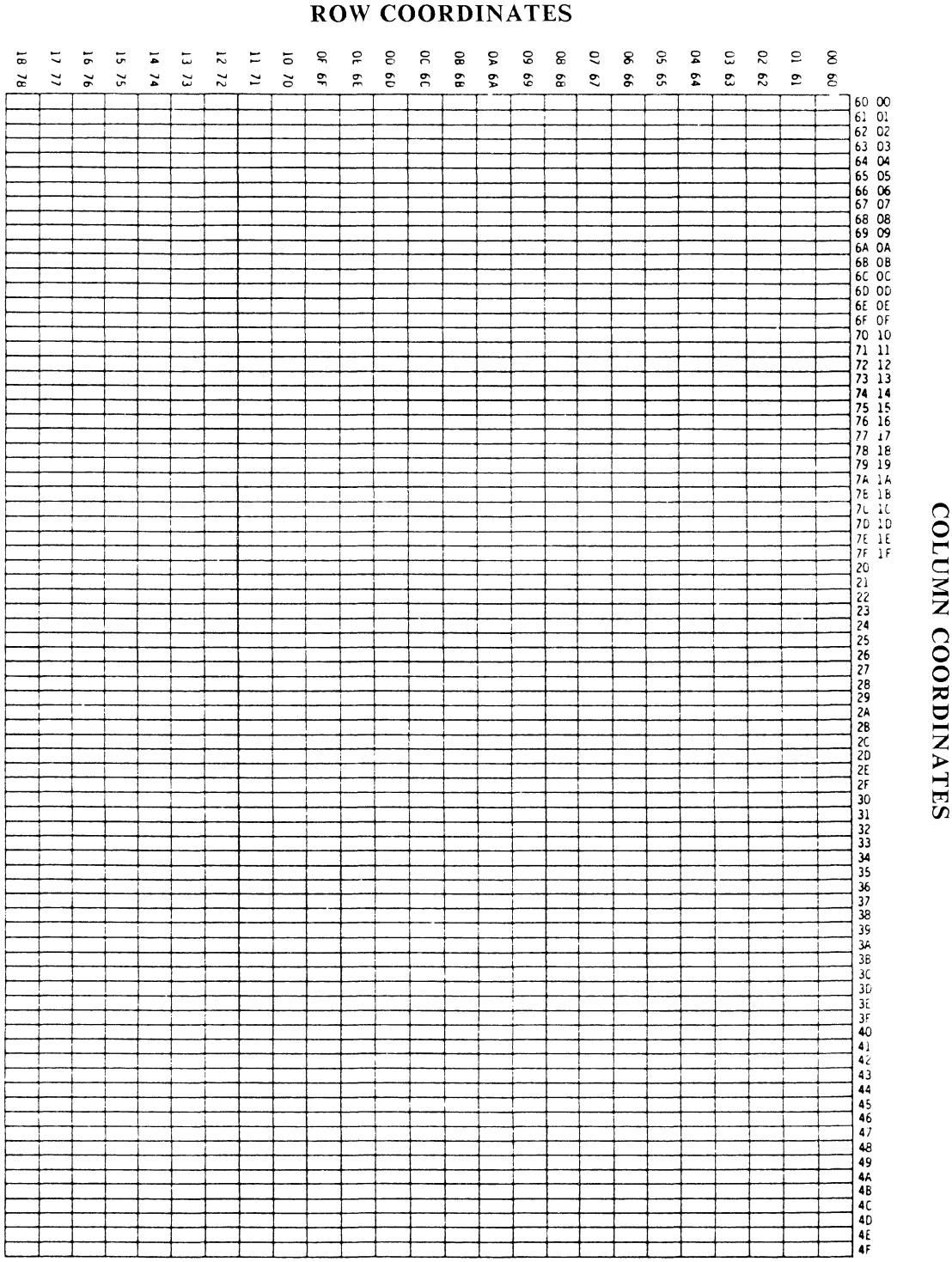


Figure 5-2 Video Display Coordinate System (Hazeltine Emulation)

5 4 35 Change Character Attributes Command

Function — This command (Change Character Attributes) is a remote command used to change a character's attribute

Hazeltine emulation — 7EH 03H <attribute byte No 1> <attribute byte No 2>
ANSI emulation — <ESC> [1,<attribute No 1>, <attribute No 2> p

The definition of attribute byte No 1 for the Industrial Terminals is shown in Table 5-3

Table 5-3 Attribute Byte 1

Bit No	Attribute
7 (MSB)	not used
6	not used
5	not used
4	double-wide
3	blink
2	underline
0 (LSB)	reverse video

The definition of attribute byte No 2 is shown in Table 5-4

Table 5-4 Attribute Byte 2

Bit No	Attribute
7 (MSB)	not used
6	not used
5	not used
4	not used
3	not used
2	character set bit 2
1	character set bit 1
0 (LSB)	character set bit 0

The settings of bits 2 to 0 of attribute byte No 2 can be any of the following (see Table 5-4)

Bit 2	Bit 1	Bit 0	Attribute
0	0	0	regular character
0	0	1	double-high character
0	1	0	quad-size character
0	1	1	process graphic symbols
1	1	1	utility graphics

5 4 36 Insert Spaces in a Line

Function This command inserts spaces into a line beginning at the cursor's current position Any characters from the cursor's current position to the end of the line will be removed

Hazeltine emulation N/A
ANSI emulation <ESC> [pn @

where ESC = 1BH
pn = the number of spaces to be inserted

5 4 37 Delete Characters in a Line

Function This command deletes characters from a line at the cursor's current position and inserts spaces at the end of the line

Hazeltine emulation N/A
ANSI emulation <ESC> [pn P

where ESC = 1BH
pn = the number of characters to be deleted

5 4 38 Saving Cursor Attributes

Function This command saves the cursor's current position, character set selection, autowrap flag state, and all attributes

Hazeltine emulation N/A
ANSI emulation <ESC> 7

where ESC = 1BH

5 4 39 Restoring Cursor Attributes

Function This command restore's the current cursor position, character set selection, autowrap flag state, and all attributes

Hazeltine emulation N/A
ANSI emulation <ESC> 8

where ESC = 1BH

5 5 Sample Screen Display

This example illustrates how to create a simple screen display by transmitting a sequence of characters to your Industrial Terminal This display prints the letters "WARNING" in quad size, reverse video inside a box

ANSI

```
"<ESC>[2,1,6,19,12,55p"    , Draw a box<CR>
"<ESC>[7,20H"              , Position cursor inside of box<CR>
"<ESC>[1,72,18p"           , Select quad size<CR>
"WARNING"                  , Message on Screen
```

Hazeltine

To create the same screen as the ANSI example, send the following characters to the terminal

```
7E 09 01 12 05             , Draw a box<CR>
7E 11 13 06                , Position cursor inside of box<CR>
7E 03 01 02                , Select quad size<CR>
"WARNING"                  , Message on Screen
```

Chapter 6

COMMUNICATIONS

6 1 INTRODUCTION

The terminal's communications capability allows data to be transferred between the terminal and a host device The terminal is equipped with an RS-232C communications port as standard equipment This can be replaced with an RS-422 or 20mA adapter module Installation of an expansion board will provide the user with a second port The type of port will depend on the type of expansion board selected

6 2 COMMUNICATIONS FORMAT

The communications ports available on the terminal support asynchronous serial data transfer using the ASCII code Data is transmitted and received at the same baud rate, and this parameter can be set to 300, 600, 1200, 2400, 4800, 9600, and 19200 for each available port

Each transmitted character includes one start bit, seven or eight data bits, one or no parity bit, and one stop bit (see Figure 6-1) The number of data bits and the parity are selected in the Configuration Menu (see Section 3 3)

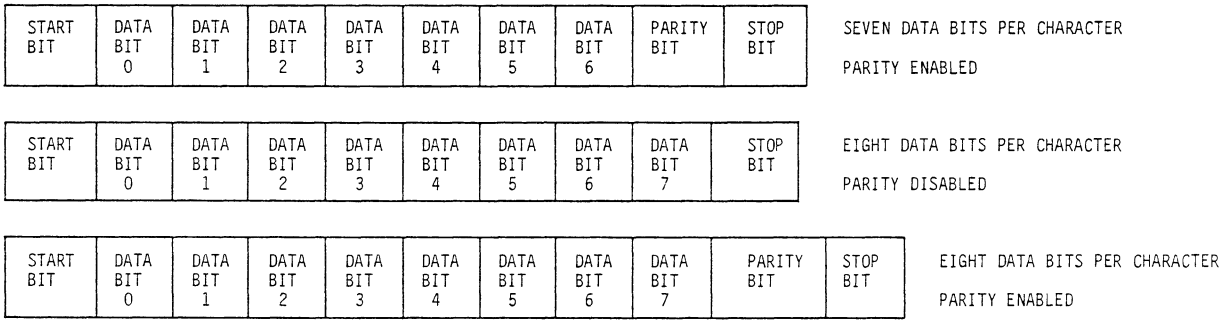


Figure 6-1 Character Format
(Start & Stop Bits, Data Bits, Parity Bit)

Seven data bits per character with parity disabled is not allowed

The parity condition can be set to one, zero, even, or odd, and the terminal can operate in full or half-duplex modes

6 3 PARITY CHECKING

The Configuration Menu allows the user to select whether parity will be employed or not If parity is employed, the user can select the parity condition to be used always one, always zero, even, or odd

When the terminal transmits any character (i e , when a key is pressed), the settings in the Configuration Menu will determine the character format and the value of the parity bit (if any)

The terminal only checks parity on received data if odd or even parity is selected If a character is received with an incorrect parity bit, a parity error symbol (P_E) will be shown on the video display at the cursor position and an audible alarm (beep) will sound

6 4 FULL AND HALF-DUPLEX OPERATION

When operating in full-duplex terminal mode, the terminal will only display information and execute commands that are received from the host device Alternately, information and commands can be entered using the terminal's keypad or optional keyboard, and echoed back to the terminal from the host device In full-duplex mode, the RTS signal will not go high until a key is pressed on the terminal, unless RTS/CTS handshaking is enabled

When operating in half-duplex terminal mode, the terminal will display information and execute commands that originate from the host device Everything typed on the terminal's keyboard will be echoed to the screen as well as the serial port The host should not echo characters back

6 5 HALF-DUPLEX OPERATION WITH A MODEM

Modem control signals are used if the DSR input to the terminal is high (active) This indicates that the terminal is connected to a modem

When data is entered using the terminal's keypad or optional keyboard, the terminal's RTS signal to the modem is set high

If DSR is detected as high (active), the terminal waits for CTS to go high (active) before transmitting the character If DSR is low (inactive), the character is transmitted immediately The terminal holds its RTS signal high (active) and entered data is transmitted until one of the following characters is entered from the keyboard or keypad

CR (0DH)
ETX (03H)
EOT (04H)

After one of the above characters is transmitted, the terminal's RTS signal is made low (inactive) and the modem enters the receive mode The sequence is repeated when data is again entered using the terminal's keypad or optional keyboard

6 6 INPUT BUFFER OVER FLOW PROTECTION

When the terminal receives a character, it is stored in a large input buffer (size greater than 1500 characters) until processed In unusual circumstances, if the terminal receives characters faster than it can process them, the input buffer can fill If the terminal's input buffer becomes full and more characters are received, those additional characters will be lost because there is no room to store them

One way to prevent this is to operate the terminal/host communications link at a baud rate low enough to give the terminal plenty of time to process a character before another is received

Another way to prevent input buffer overflow is to send the terminal fill characters between valid data The NUL character (00H) is used as the fill character When received by the terminal, the NUL character is ignored Commands for operations which require a relatively long time for the terminal to perform should be followed by fill characters if this method is used

Table 6-1 Commands Whose Use May Require Input Buffer Protection

Clear Screen	Draw Box
Clear Foreground	Draw Vertical Line
Clear to End of Line	Draw Horizontal Line
Clear to End of Screen	Execute Screen (4800-E1 option installed)
Clear to End of Screen (background spaces)	Draw Bar (Up, Down, Left, Right)
Delete Line	Insert Spaces
Insert Line	Delete Characters
Display of double and quad-size characters	Clear Line
Clear to Beginning of Line	
Clear to beginning of Screen	

A third and preferred method for preventing input buffer overflow is to use either RTS/CTS or XON/XOFF control characters when operating in full-duplex mode If XON/XOFF generation is enabled, and if there are fewer than 32 free bytes remaining in the input buffer, the XOFF control character will be sent to the host device at this time When the XOFF signal is received, the host device should stop transmitting When the buffer again contains more than 1000 free bytes, the XON control character will be sent to the host device Transmission can then be resumed The following characters are used as the XON/XOFF characters

XON = DC1 (11H) XOFF = DC3 (13H)

NOTE

XON/XOFF should not be used in Hazeltine Mode

If RTS/CTS handshaking is selected (see Configuration Menu, Section 2 2 5), the terminal must have an active CTS before it will transmit any data, and will activate RTS when it is able to receive any data Note Lines 9 and 10 must be connected on RS-232 to disable optical isolation When RTS/CTS Handshaking is used

6 7 RS-232C COMMUNICATIONS PORT

The following chart shows pin numbers and signals for the RS-232C communications port All signals are positive logic (active high)

Pin Number	Direction of Signal	Designation	Function
1	- -	AA	Frame Ground
2	From Terminal	BA	Transmit Data
3	To Terminal	BB	Receive Data
4	From Terminal	CA	Request To Send (RTS) (1)
5	To Terminal	CB	Clear To Send (CTS) (1)
6	To Terminal	CC	Data Set Ready (DSR) (1)
7	- -	AB	Signal Ground
9			Disable Optical Isolation
10			Disable Optical Isolation
20	From Terminal	CD	Data Terminal Ready (DTR) (1)

NOTE:

(1) Modem Control

According to the EIA RS-232C specifications, there should be no more than 50 feet of cable between the host RS-232C port device and the terminal RS-232C port

Chapter 7

DIAGNOSTICS

7 1 INTRODUCTION

The terminal is capable of performing self-diagnostic tests when operating in diagnostic mode In this mode, the terminal displays a menu of diagnostic tests for the operator to select from If a problem is found, an error message will be shown on the video display

7 2 DIAGNOSTICS

To enter the Diagnostics Menu, type "2" when the Main Menu is displayed The Diagnostics Menu can be activated by pressing the appropriate key When a selected test is completed, the Diagnostics Menu will be redisplayed

```
        Diagnostics

     1) RAM
     2) ROM Checksum
     3) Character Attributes
     4) Serial Loop Back
     5) CRT Alignment Pattern
     6) CRT Brightness Pattern
     7) Continuous test
     8) Character Generator Test

<RET> or <ENTER> to quit
   >_
```

Figure 7-1 Diagnostics Menu

1) RAM

If the RAM test is selected, the terminal will check the CPU RAM (8031 RAM), then the external RAM, then the display RAM, and then the attribute RAM After checking the 8031 RAM the terminal will display one of the following messages

8031 RAM OK

or

8031 RAM failure

The next test checks the external RAM which is the serial input buffer After testing the external RAM, the terminal displays one of these messages

External RAM OK

or

External RAM failure ab/cd wxyz

The terminal will then test the display RAM, during which a pattern will be flashed on the video display followed by one of the messages

Display RAM OK

or

Display RAM failure ab/cd wxyz

where ab is the byte read from the failed memory address, cd is the byte that was written to the failed memory address, and wxyz is the failed memory address All of these numbers are in hexadecimal format

The terminal will then test the attribute RAM, again flashing a pattern on the video display followed by one of the messages

Attribute RAM 0 OK

or

Attribute RAM 0 failure ab/cd wxyz

where ab/cd wxyz has the same meaning as for the display RAM message

2) ROM checksum

Shows "ROM checksum is nnnn Should be mmmm" on status line The two checksums listed (nnnn) should match

3) Character attributes

The status line shows Reverse Video, Underline, Blink, and Double Wide Each word or phrase on the status line should be displayed with its corresponding attribute

4) Serial loop back

The serial port on the terminal can be tested by selecting #4 from the diagnostic menu "Serial loopback test" In order for the "loop back" test to function properly, the serial port must have certain signals looped-back for signal verification The recommended means for looping signals is via construction of a loop-back connector using a DB-25 connector, and several jumper wires (or solder bridges) Figure 7-2 shows the jumper configuration for the construction of a loop-back plug which can be used to test the serial port

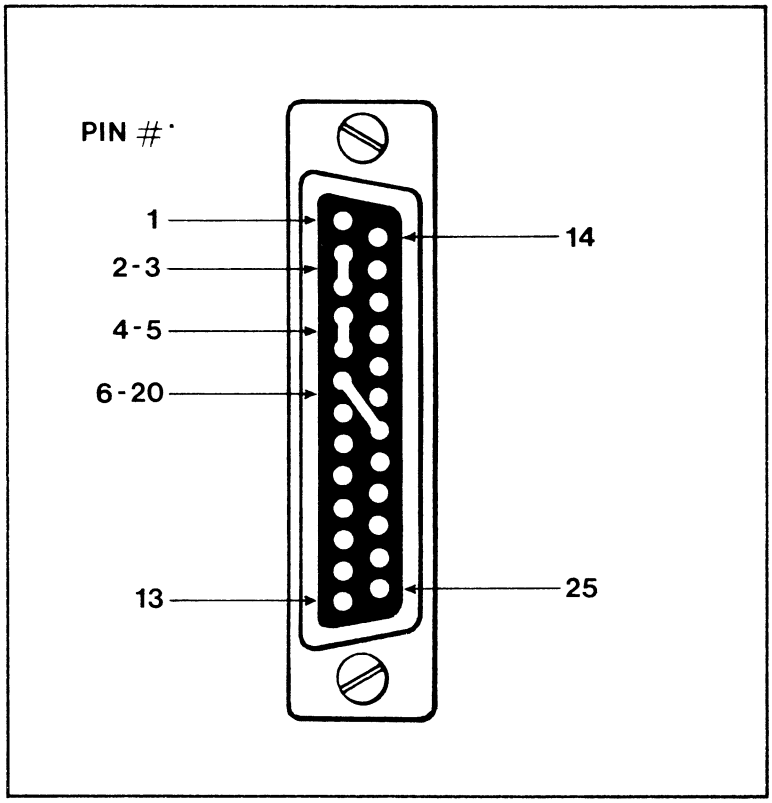


Figure 7-2 Serial Port Test Plug

If the serial port is operating correctly, the terminal will display the message

Controller serial port OK

If an error is found, the terminal will display one of the following messages

Controller port time out err
Controller port serial data err
Controller port CTS-RTS err
Controller port DTR-DSR err

5) EL alignment pattern

Shows alignment grid on video display until a key is pressed

6) EL brightness pattern

Displays foreground spaces on the entire screen

7) Continuous test

In this mode, the terminal continuously cycles through the RAM, serial port, and ROM tests If an error is found, the terminal stops testing and displays an appropriate error message along with the prompt

Press any key to continue

If a key is then pressed, testing will continue

To exit the continuous test mode, press any key several times

8) Character Generator test

Displays all displayable characters, including the block and bar graphics characters Pressing any key will cause the following screens to be displayed

- double-high characters
- quad-size special characters and numbers
- upper-case quad-size
- lower-case quad-size
- process control graphics (large)
- process control graphics (small)

Appendix A

4870 PANEL CUTOUT DIMENSIONS

A 1 PANEL-MOUNTING THE 4870

Figure A-1 below shows how a panel should be cut and drilled to panel-mount a 4870 terminal Note that the 1/4 inch holes in the panel will accommodate the 10-32 studs on the back of the terminal front panel

A 2 RACK-MOUNTING THE 4870

The figure below shows the locations of the studs on the back of the terminal front panel Since these studs are threaded, the terminal is designed to be installed in a rack in which the holes are not threaded If the rack has threaded holes, they must be drilled out Using a #3 bit (213 inches) will provide clearance for the 10-32 studs

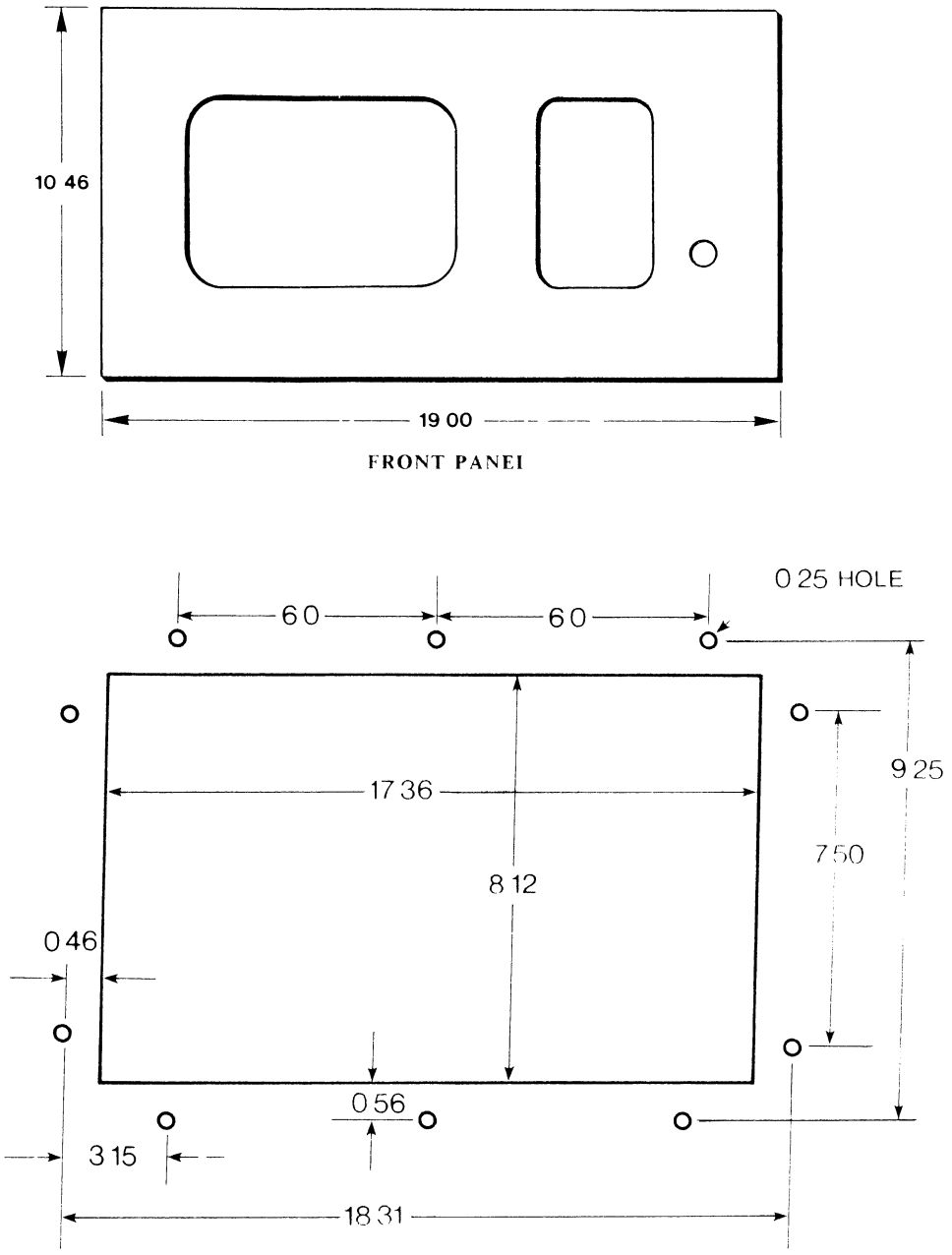


Figure A-1 Panel Cutout for the 4870

Appendix B

VT100/220 CODES NOT SUPPORTED

The 4870 Industrial Terminal supports all VT100 codes except for those listed in this Section If these codes are received by the terminal, they will be ignored

Control Characters Not Supported

0E - select G1 character set
0F - select GO character set

Digital Equipment Corporation Private Configuration Commands Not Supported

ESC [? 1 h - enable application interp of cursor keys
ESC [? 2 h - enable ANSI mode
ESC [? 3 h - enable 132 column mode
ESC [? 5 h - enable reverse screen mode
ESC [? 6 h - enable origin mode
ESC [? 8 h - enable auto repeat
ESC [? 9 h - enable interlace
ESC [? 18 h - print form feed enabled
ESC [? 19 h - full screen print extent
ESC [? 42 h - national character set
ESC [? 1 l - disable application interp of cursor keys
ESC [? 2 l - enable VT52 mode
ESC [? 3 l - enable 80 column mode
ESC [? 5 l - enable normal screen
ESC [? 6 l - enable absolute mode
ESC [? 8 l - disable auto repeat
ESC [? 9 l - disable interlace
ESC [? 18 l - print form feed disable
ESC [? 19 l - scrolling region print extent
ESC [? 42 l - multinational character set

Configuration Commands Not Supported

ESC [4 h - insert mode enable
ESC [12 h - local echo disabled
ESC [4 l - replace mode enable
ESC [12 l - local echo enabled

Select Characters Set Codes Not Supported

ESC (A	- UK G0
ESC (B	- USASCII G0
ESC (0	- special chars and lines G0
ESC (1	- alternate ROM G0
ESC (2	- alternate ROM and special graphics G0
ESC (A	- UK G1
ESC (B	- USASCII G1
ESC (0	- special chars and lines G1
ESC (1	- alternate ROM G1
ESC (2	- alternate ROM and special graphics G1
ESC N	- single shift 2
ESC O	- single shift 3

Scrolling Region Command Not Supported

ESC [pt,pb r	- Set top and bottom margin

Line Attribute Commands Not Supported

ESC # 3	- double-high top half
ESC # 4	- double-high bottom half
ESC # 5	- single-wide, single-high
ESC # 6	- double-wide, single-high
ESC # 8	- fill screen with e's

Test Commands Not Supported

ESC [1,1 y	- invoke power-up test
ESC [2,2 y	- data loopback test
ESC [2,9 y	- continuous power-up testing
ESC [2,10 y	- continuous loopback test

Keyboard LED Commands Not Supported

ESC [0 q	- all LEDs off
ESC [1 q	- LED 1 on
ESC [2 q	- LED 2 on
ESC [3 q	- LED 3 on
ESC [4 q	- LED 4 on

Aux Keypad Codes in Application Mode Not Generated

-	- ESC 0 m
,	- ESC 0 l

Report Commands Not Supported

ESC [? 15n	- what is printer status
ESC [? 25n	- what is status of user-defined keys
ESC [? 26n	- what is keyboard language

Appendix C

PROCESS GRAPHIC'S CHART

Appendix C (Table C-2) shows the symbols displayed in the various character set modes

In general, a character or a number representing a character (character code) is sent to the display The symbol displayed depends on the character set selected

The characters and the corresponding character codes are shown along the top axis of the table

The character set choices are shown along the left axis of the table The designation of the character set differs depending on whether or not a XYCOM Expansion Module with Operator Interface Language (OIL) is installed in the terminal

If the OIL option is not installed, the base terminals Character Set can be selected by one of two methods The table's Base Terminal character set is indicated along the left axis using attribute Byte 2, bits 2-0, and attribute Byte 1, bit 4 (See Table C-1)

Table C-1

Attribute Bytes 2 and 1
Base Terminal Character Sets

	Attribute Byte 2 Bits 2 - 0	Byte 1 Bit 4	Set/Reset Attribute Code
Regular	000	0	50
Double-Wide	000	1	54
Double-High	001	0	51
Double-Size	001	1	55
Quad-Size	010	0	52
Process Graphics	011	0	53
Utility Graphics	111	0	57

Table C-2 Process Graphics Chart

CHARACTER		NONE	NONE	NONE	NONE	NONE	NONE	NONE	NONE	NONE
HEXADECIMAL		00	01	02	03	04	05	06	07	08
DECIMAL		0	1	2	3	4	5	6	7	8
CHARACTER SET										
BASE TERMINAL *	OIL OPTION									
000 (1)	REG	N_U	S_H	S_X	E_X	E_T	E_Q	A_K	B_L	B_S
000 ** (2)	DW	N_U	S_H	S_X	E_X	E_T	E_Q	A_K	B_L	B_S
001	DH									
001 **	DS									
010	QS									
011	G1									
N/A	G2									
N/A	G3									
N/A	G4									
111	N/A									

NOTES

1 - ONLY DISPLAYABLE WHEN CONFIGURATION OPTION "DISPLAY CONTROL CODES" IS ENABLED

2 - NOT POSSIBLE ON BASE TERMINAL

* ATTRIBUTE BYTE 2, BITS 2-0

** ATTRIBUTE BYTE 1, BIT 1 MUST BE SET FOR THIS MODE

CHARACTER		NONE	NONE	NONE	NONE	NONE	NONE	NONE	NONE	NONE
HEXADECIMAL		09	0A	0B	0C	0D	0E	0F	10	11
DECIMAL		9	10	11	12	13	14	15	16	17
CHARACTER SET										
BASE TERMINAL *	OIL OPTION									
000 (1)	REG	H_T	L_F	V_T	F_F	C_R	S_0	S_1	D_L	D_1
000 ** (2)	DW	H_T	L_F	V_T	F_F	C_R	S_0	S_1	D_L	D_1
001	DH									
001 **	DS									
010	QS									
011	G1									
N/A	G2									
N/A	G3									
N/A	G4									
111	N/A									

NOTES

1 - ONLY DISPLAYABLE WHEN CONFIGURATION OPTION "DISPLAY CONTROL CODES" IS ENABLED

2 - NOT POSSIBLE ON BASE TERMINAL

* ATTRIBUTE BYTE 2, BITS 2-0

** ATTRIBUTE BYTE 1, BIT 1 MUST BE SET FOR THIS MODE

CHARACTER	NONE	NONE	NONE	NONE	NONE	NONE	NONE	NONE	NONE
HEXADECIMAL	12	13	14	15	16	17	18	19	1A
DECIMAL	18	19	20	21	22	23	24	25	26
CHARACTER SET									
BASE TERMINAL *	OIL OPTION								
000 (1)	REG	D_2	D_3	D_4	N_K	S_Y	E_B	C_N	E_M	S_B
000 ** (2)	DW	D_2	D_3	D_4	N_K	S_Y	E_B	C_N	E_M	S_B
001	DH									
001 **	DS									
010	QS									
011	G1									
N/A	G2									
N/A	G3									
N/A	G4									
111	N/A									

NOTES

1 - ONLY DISPLAYABLE WHEN CONFIGURATION OPTION "DISPLAY CONTROL CODES" IS ENABLED

2 - NOT POSSIBLE ON BASE TERMINAL

* ATTRIBUTE BYTE 2, BITS 2-0

** ATTRIBUTE BYTE 1, BIT 1 MUST BE SET FOR THIS MODE

CHARACTER	NONE	NONE	NONE	NONE	NONE	SPACE	!	"	#
HEXADECIMAL	1B	1C	1D	1E	1F	20	21	22	23
DECIMAL	27	28	29	30	31	32	33	34	35
CHARACTER SET									
BASE TERMINAL *	OIL OPTION								
000	REG	E_C (1)	F_S (1)	G_S (1)	R_S (1)	U_S (1)	SPACE	!	"	#
000 **	DW	E_C (2)	F_S (2)	G_S (2)	R_S (2)	U_S (2)	SPACE	!	"	#
001	DH						SPACE	!	"	#
001 **	DS						SPACE	!	"	#
010	QS						SPACE	!	"	#
011	G1						SPACE			
N/A	G2									
N/A	G3									
N/A	G4									
111	N/A									

NOTES

1 - ONLY DISPLAYABLE WHEN CONFIGURATION OPTION "DISPLAY CONTROL CODES" IS ENABLED

2 - NOT POSSIBLE ON BASE TERMINAL

* ATTRIBUTE BYTE 2, BITS 2-0

** ATTRIBUTE BYTE 1, BIT 1 MUST BE SET FOR THIS MODE

△ CHARACTER CELL SHOWN LARGER THAN ACTUAL SIZE

CHARACTER		$	%	&	'	(	)	*	+	,
HEXADECIMAL		24	25	26	27	28	29	2A	2B	2C
DECIMAL		36	37	38	39	40	41	42	43	44
CHARACTER SET										
BASE TERMINAL *	OIL OPTION									
000	REG	$	%	&	'	(	)	*	+	,
000 **	DW	$	%	&	'	(	)	*	+	,
001	DH	$	%	&	'	(	)	*	+	,
001 **	DS	$	%	&	'	(	)	*	+	,
010	QS	$	%	&	'	(	)	*	+	,
011	G1									
N/A	G2									
N/A	G3									
N/A	G4									
111	N/A									

NOTES

* ATTRIBUTE BYTE 2, BITS 2-0

** ATTRIBUTE BYTE 1, BIT 1 MUST BE SET FOR THIS MODE

△ CHARACTER CELL SHOWN LARGER THAN ACTUAL SIZE

CHARACTER	-		/	0	1	2	3	4	5
HEXADECIMAL	2D	2E	2F	30	31	32	33	34	35
DECIMAL	45	46	47	48	49	50	51	52	53

CHARACTER SET										
BASE * TERMINAL	OIL OPTION									
000	REG	-		/	0	1	2	3	4	5
000 **	DW	—	.	⁄	Ø	1	2	3	4	5
001	DH	-	·	/	0	1	2	3	4	5
001 **	DS	—	·	/	0	1	2	3	4	5
010	QS	—	·	/	0	1	2	3	4	5
011	G1									
N/A	G2									
N/A	G3									
N/A	G4									
111	N/A									

NOTES

* ATTRIBUTE BYTE 2, BITS 2-0

** ATTRIBUTE BYTE 1, BIT 1 MUST BE SET FOR THIS MODE

△ CHARACTER CELL SHOWN LARGER THAN ACTUAL SIZE

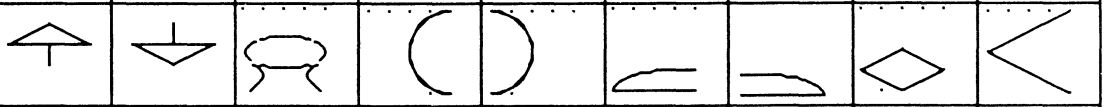
CHARACTER	6	7	8	9		,	<	=	>
HEXADECIMAL	36	37	38	39	3A	3B	3C	3D	3E
DECIMAL	54	55	56	57	58	59	60	61	62
CHARACTER SET									
BASE TERMINAL *	OIL OPTION								
000	REG	6	7	8	9		,	<	=	>
000 **	DW	6	7	8	9	:	;	<	=	>
001	DH	6	7	8	9	:	;	<	=	>
001 **	DS	6	7	8	9	:	;	<	=	>
010	QS	6	7	8	9	:	;	<	=	>
011	G1									
N/A	G2									
N/A	G3									
N/A	G4									
111	N/A									

NOTES

* ATTRIBUTE BYTE 2, BITS 2-0

** ATTRIBUTE BYTE 1, BIT 1 MUST BE SET FOR THIS MODE

△ CHARACTER CELL SHOWN LARGER THAN ACTUAL SIZE

CHARACTER		?	@	A	B	C	D	E	F	G
HEXADECIMAL		3F	40	41	42	43	44	45	46	47
DECIMAL		63	64	65	66	67	68	69	70	71
CHARACTER SET										
BASE TERMINAL *	OIL OPTION									
000	REG	?	@	A	B	C	D	E	F	G
000 **	DW	?	@	A	B	C	D	E	F	G
001	DH	?	@	A	B	C	D	E	F	G
001 **	DS	?	@	A	B	C	D	E	F	G
010	QS	?	@	A	B	C	D	E	F	G
011	G1									
N/A	G2									
N/A	G3									
N/A	G4									
111	N/A									

NOTES

* ATTRIBUTE BYTE 2, BITS 2-0

** ATTRIBUTE BYTE 1, BIT 1 MUST BE SET FOR THIS MODE

△ CHARACTER CELL SHOWN LARGER THAN ACTUAL SIZE

CHARACTER	H	I	J	K	L	M	N	O	P
HEXADECIMAL	48	49	4A	4B	4C	4D	4E	4F	50
DECIMAL	72	73	74	75	76	77	78	79	80

CHARACTER SET										
BASE TERMINAL *	OIL OPTION									
000	REG	H	I	J	K	L	M	N	O	P
000 **	DW	H	I	J	K	L	M	N	O	P
001	DH	H	I	J	K	L	M	N	O	P
001 **	DS	H	I	J	K	L	M	N	O	P
010	QS	H	I	J	K	L	M	N	O	P
011	G1									
N/A	G2									
N/A	G3									
N/A	G4									
111	N/A									

NOTES

* ATTRIBUTE BYTE 2, BITS 2-0

** ATTRIBUTE BYTE 1, BIT 1 MUST BE SET FOR THIS MODE

△ CHARACTER CELL SHOWN LARGER THAN ACTUAL SIZE

CHARACTER		Q	R	S	T	U	V	W	X	Y
HEXADECIMAL		4F	50	53	54	55	56	57	58	59
DECIMAL		81	82	83	84	85	86	87	88	89
CHARACTER SET										
BASE TERMINAL *	OIL OPTION									
000	REG	Q	R	S	T	U	V	W	X	Y
000 **	DW	Q	R	S	T	U	V	W	X	Y
001	DH	Q	R	S	T	U	V	W	X	Y
001 **	DS	Q	R	S	T	U	V	W	X	Y
010	QS	Q	R	S	T	U	V	W	X	Y
011	G1									
N/A	G2									
N/A	G3									
N/A	G4									
111	N/A									

NOTES

* ATTRIBUTE BYTE 2, BITS 2-0

** ATTRIBUTE BYTE 1, BIT 1 MUST BE SET FOR THIS MODE

△ CHARACTER CELL SHOWN LARGER THAN ACTUAL SIZE

CHARACTER	Z	[	\	]	↑	_	`	a	b
HEXADECIMAL	5A	5B	5C	5D	5E	5F	60	61	62
DECIMAL	90	91	92	93	94	95	96	97	98
CHARACTER SET									
BASE TERMINAL *	OIL OPTION								
000	REG	Z	[	\	]	↑	_	`	a	b
000 **	DW	Z	[	\	]	↑	_	`	a	b
001	DH	Z	[	\	]	↑	_	`	a	b
001 **	DS	Z	[	\	]	↑	_	`	a	b
010	QS	Z	[	\	]	↑	_	`	a	b
011	G1									
N/A	G2									
N/A	G3									
N/A	G4									
111	N/A									

NOTES

* ATTRIBUTE BYTE 2, BITS 2-0

** ATTRIBUTE BYTE 1, BIT 1 MUST BE SET FOR THIS MODE

△ CHARACTER CELL SHOWN LARGER THAN ACTUAL SIZE

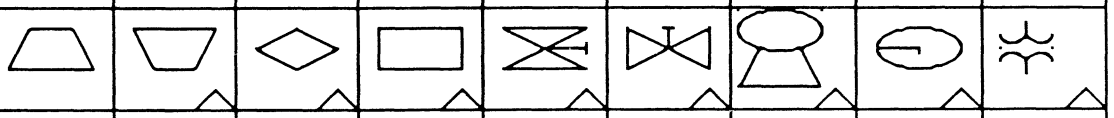
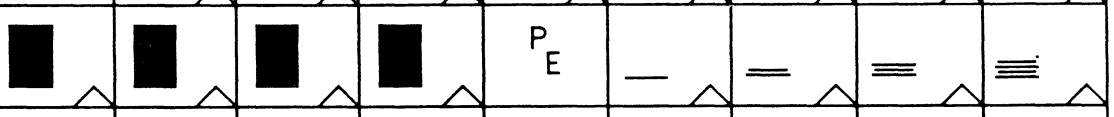
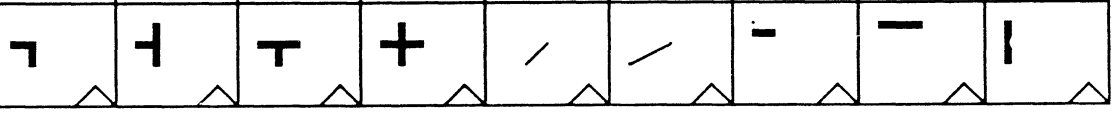
CHARACTER	c	d	e	f	g	h	i	j	k
HEXADECIMAL	63	64	65	66	67	68	69	6A	6B
DECIMAL	99	100	101	102	103	104	105	106	107
CHARACTER SET									
BASE TERMINAL *	OIL OPTION								
000	REG	c	d	e	f	g	h	i	j	k
000 **	DW	c	d	e	f	g	h	i	j	k
001	DH	c	d	e	f	g	h	i	j	k
001 **	DS	c	d	e	f	g	h	i	j	k
010	QS	c	d	e	f	g	h	i	j	k
011	G1									
N/A	G2									
N/A	G3									
N/A	G4									
111	N/A									

NOTES

* ATTRIBUTE BYTE 2, BITS 2-0

** ATTRIBUTE BYTE 1, BIT 1 MUST BE SET FOR THIS MODE

△ CHARACTER CELL SHOWN LARGER THAN ACTUAL SIZE

CHARACTER	l	m	n	o	p	q	r	s	t
HEXADECIMAL	6C	6D	6E	6F	70	71	72	73	74
DECIMAL	108	109	110	111	112	113	114	115	116
CHARACTER SET									
BASE TERMINAL *	OIL OPTION								
000	REG	l	m	n	o	p	q	r	s	t
000 **	DW	l	m	n	o	p	q	r	s	t
001	DH	l	m	n	o	p	q	r	s	t
001 **	DS	l	m	n	o	p	q	r	s	t
010	QS	l	m	n	o	p	q	r	s	t
011	G1									
N/A	G2									
N/A	G3									
N/A	G4									
111	N/A									

NOTES

* ATTRIBUTE BYTE 2, BITS 2-0

** ATTRIBUTE BYTE 1, BIT 1 MUST BE SET FOR THIS MODE

△ CHARACTER CELL SHOWN LARGER THAN ACTUAL SIZE

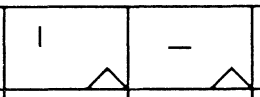
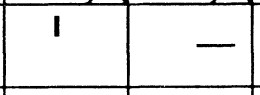
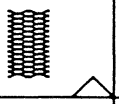
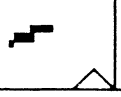
CHARACTER		u	v	w	x	y	z	{	¦	}
HEXADECIMAL		75	76	77	78	79	7A	7B	7C	7D
DECIMAL		117	118	119	120	121	122	123	124	125
CHARACTER SET										
BASE * TERMINAL	OIL OPTION									
000	REG	u	v	w	x	y	z	{	¦	}
000 **	DW	u	v	w	x	y	z	{	¦	}
001	DH	u	v	w	x	y	z	{	¦	}
001 **	DS	u	v	w	x	y	z	{	¦	}
010	QS	u	v	w	x	y	z	{	¦	}
011	G1									
N/A	G2									
N/A	G3									
N/A	G4									
111	N/A									

NOTES

* ATTRIBUTE BYTE 2, BITS 2-0

** ATTRIBUTE BYTE 1, BIT 1 MUST BE SET FOR THIS MODE

△ CHARACTER CELL SHOWN LARGER THAN ACTUAL SIZE

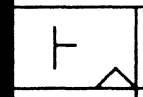
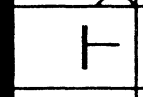
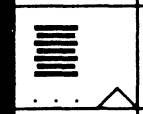
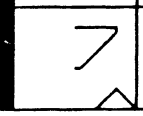
CHARACTER	~	NONE	NONE	NONE	NONE	NONE	NONE	NONE	NONE
HEXADECIMAL	7E	7F	80	81	82	83	84	85	86
DECIMAL	126	127	128	129	130	131	132	133	134
CHARACTER SET									
BASE TERMINAL *	OIL OPTION								
000	REG	~								
000 **	DW	~								
001	DH	~								
001 **	DS	~								
010	QS									
011	G1									
N/A	G2									
N/A	G3									
N/A	G4									
111	N/A									

NOTES

* ATTRIBUTE BYTE 2, BITS 2-0

** ATTRIBUTE BYTE 1, BIT 1 MUST BE SET FOR THIS MODE

△ CHARACTER CELL SHOWN LARGER THAN ACTUAL SIZE

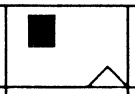
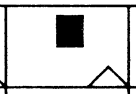
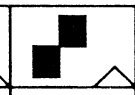
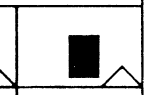
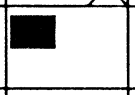
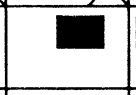
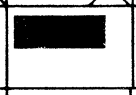
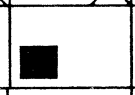
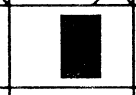
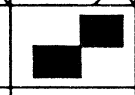
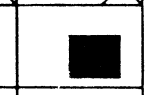
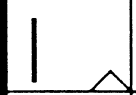
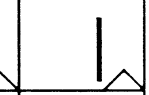
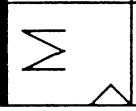
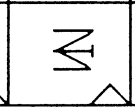
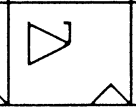
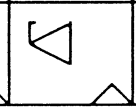
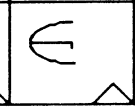
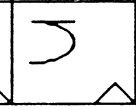
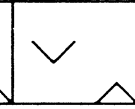
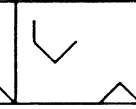
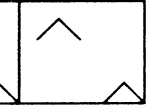
CHARACTER		NONE	NONE	NONE	NONE	NONE	NONE	NONE	NONE	NONE
HEXADECIMAL		87	88	89	8A	8B	8C	8D	8E	8F
DECIMAL		135	136	137	138	139	140	141	142	143
CHARACTER SET										
BASE TERMINAL *	OIL OPTION									
000	REG									
000 **	DW									
001	DH									
001 **	DS									
010	QS									
011	G1									
N/A	G2									
N/A	G3									
N/A	G4									
111	N/A									

NOTES

* ATTRIBUTE BYTE 2, BITS 2-0

** ATTRIBUTE BYTE 1, BIT 1 MUST BE SET FOR THIS MODE

△ CHARACTER CELL SHOWN LARGER THAN ACTUAL SIZE

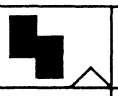
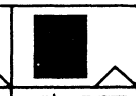
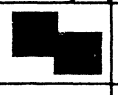
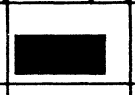
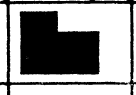
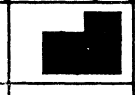
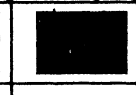
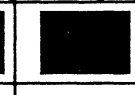
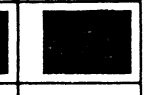
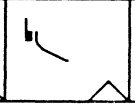
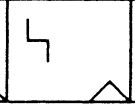
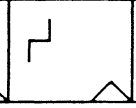
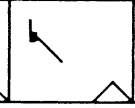
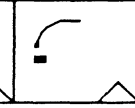
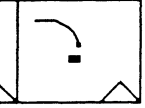
CHARACTER		NONE	NONE	NONE	NONE	NONE	NONE	NONE	NONE	NONE
HEXADECIMAL		90	91	92	93	94	95	96	97	98
DECIMAL		144	145	146	147	148	149	150	151	152
CHARACTER SET										
BASE TERMINAL *	OIL OPTION									
000	REG									
000 **	DW									
001	DH									
001 **	DS									
010	QS									
011	G1									
N/A	G2									
N/A	G3									
N/A	G4									
111	N/A									

NOTES

* ATTRIBUTE BYTE 2, BITS 2-0

** ATTRIBUTE BYTE 1, BIT 1 MUST BE SET FOR THIS MODE

△ CHARACTER CELL SHOWN LARGER THAN ACTUAL SIZE

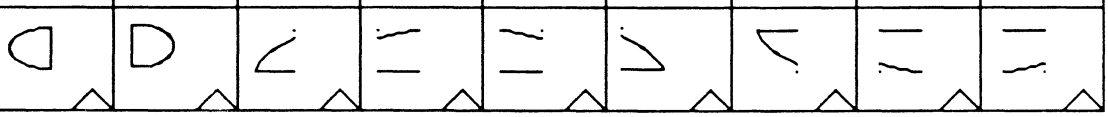
CHARACTER		NONE	NONE	NONE	NONE	NONE	NONE	NONE	NONE	NONE
HEXADECIMAL		99	9A	9B	9C	9D	9E	9F	A0	A1
DECIMAL		153	154	155	156	157	158	159	160	161
CHARACTER SET										
BASE TERMINAL *	OIL OPTION									
000	REG									
000 **	DW									
001	DH									
001 **	DS									
010	QS									
011	G1									
N/A	G2									
N/A	G3									
N/A	G4									
111	N/A									

NOTES

* ATTRIBUTE BYTE 2, BITS 2-0

** ATTRIBUTE BYTE 1, BIT 1 MUST BE SET FOR THIS MODE

△ CHARACTER CELL SHOWN LARGER THAN ACTUAL SIZE

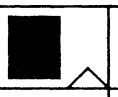
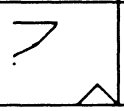
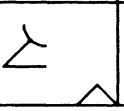
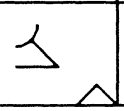
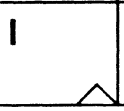
CHARACTER		NONE	NONE	NONE	NONE	NONE	NONE	NONE	NONE	NONE
HEXADECIMAL		A2	A3	A4	A5	A6	A7	A8	A9	AA
DECIMAL		162	163	164	165	166	167	168	169	170
CHARACTER SET										
BASE TERMINAL *	OIL OPTION									
000	REG									
000 **	DW									
001	DH									
001 **	DS									
010	QS									
011	G1									
N/A	G2									
N/A	G3									
N/A	G4									
111	N/A									

NOTES

* ATTRIBUTE BYTE 2, BITS 2-0

** ATTRIBUTE BYTE 1, BIT 1 MUST BE SET FOR THIS MODE

△ CHARACTER CELL SHOWN LARGER THAN ACTUAL SIZE

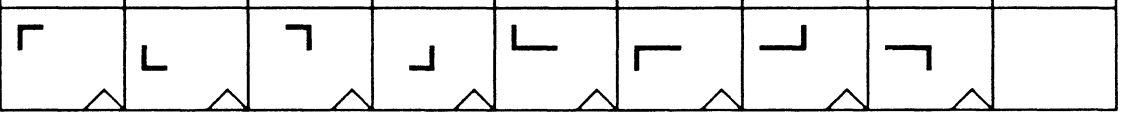
CHARACTER	NONE	NONE	NONE	NONE	NONE	NONE	NONE	NONE	NONE
HEXADECIMAL	AB	AC	AD	AE	AF	B0	B1	B2	B3
DECIMAL	171	172	173	174	175	176	177	178	179
CHARACTER SET									
BASE TERMINAL * / OIL OPTION									
000 / REG									
000 ** / DW									
001 / DH									
001 ** / DS									
010 / QS									
011 / G1									
N/A / G2									
N/A / G3									
N/A / G4									
111 / N/A									

NOTES

* ATTRIBUTE BYTE 2, BITS 2-0

** ATTRIBUTE BYTE 1, BIT 1 MUST BE SET FOR THIS MODE

△ CHARACTER CELL SHOWN LARGER THAN ACTUAL SIZE

CHARACTER	NONE	NONE	NONE	NONE	NONE	NONE	NONE	NONE	NONE
HEXADECIMAL	B4	B5	B6	B7	B8	B9	BA	BB	BC
DECIMAL	180	181	182	183	184	185	186	187	188
CHARACTER SET									
BASE TERMINAL *	OIL OPTION								
000	REG								
000 **	DW								
001	DH								
001 **	DS								
010	QS								
011	G1								
N/A	G2								
N/A	G3								
N/A	G4								
111	N/A								

NOTES

* ATTRIBUTE BYTE 2, BITS 2-0

** ATTRIBUTE BYTE 1, BIT 1 MUST BE SET FOR THIS MODE

△ CHARACTER CELL SHOWN LARGER THAN ACTUAL SIZE

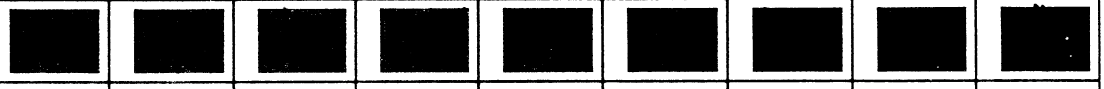
CHARACTER		NONE	NONE	NONE	NONE	NONE	NONE	NONE	NONE	NONE
HEXADECIMAL		BD	BE	BF	C0	C1	C2	C3	C4	C5
DECIMAL		189	190	191	192	193	194	195	196	197
CHARACTER SET										
BASE TERMINAL *	OIL OPTION									
000	REG									
000 **	DW									
001	DH									
001 **	DS									
010	QS									
011	G1									
N/A	G2									
N/A	G3									
N/A	G4									
111	N/A				Γ	L				

NOTES

* ATTRIBUTE BYTE 2, BITS 2-0

** ATTRIBUTE BYTE 1, BIT 1 MUST BE SET FOR THIS MODE

△ CHARACTER CELL SHOWN LARGER THAN ACTUAL SIZE

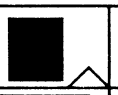
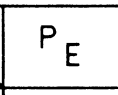
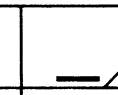
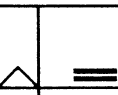
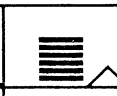
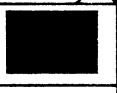
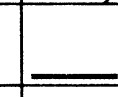
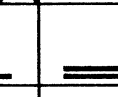
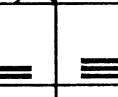
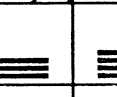
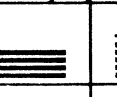
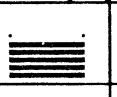
CHARACTER		NONE	NONE	NONE	NONE	NONE	NONE	NONE	NONE	NONE
HEXADECIMAL		C6	C7	C8	C9	CA	CB	CC	CD	CE
DECIMAL		198	199	200	201	202	203	204	205	206
CHARACTER SET										
BASE TERMINAL *	OIL OPTION									
000	REG									
000 **	DW									
001	DH									
001 **	DS									
010	QS									
011	G1									
N/A	G2									
N/A	G3									
N/A	G4									
111	N/A									

NOTES

* ATTRIBUTE BYTE 2, BITS 2-0

** ATTRIBUTE BYTE 1, BIT 1 MUST BE SET FOR THIS MODE

△ CHARACTER CELL SHOWN LARGER THAN ACTUAL SIZE

CHARACTER	NONE	NONE	NONE	NONE	NONE	NONE	NONE	NONE	NONE
HEXADECIMAL	CF	D0	D1	D2	D3	D4	D5	D6	D7
DECIMAL	207	208	209	210	211	212	213	214	215
CHARACTER SET									
BASE TERMINAL *	OIL OPTION								
000	REG									
000 **	DW									
001	DH									
001 **	DS									
010	QS									
011	G1									
N/A	G2									
N/A	G3									
N/A	G4									
111	N/A									

NOTES

* ATTRIBUTE BYTE 2, BITS 2-0

** ATTRIBUTE BYTE 1, BIT 1 MUST BE SET FOR THIS MODE

△ CHARACTER CELL SHOWN LARGER THAN ACTUAL SIZE

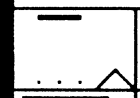
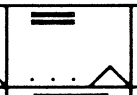
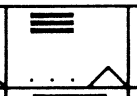
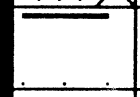
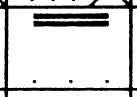
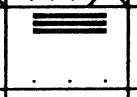
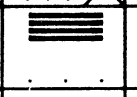
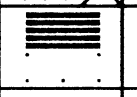
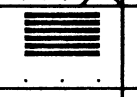
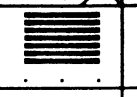
CHARACTER	NONE	NONE	NONE	NONE	NONE	NONE	NONE	NONE	NONE
HEXADECIMAL	D8	D9	DA	DB	DC	DD	DE	DF	E0
DECIMAL	216	217	218	219	220	221	222	223	224
CHARACTER SET									
BASE TERMINAL * / OIL OPTION									
000 / REG									
000 ** / DW									
001 / DH									
001 ** / DS									
010 / QS									
011 / G1									
N/A / G2									
N/A / G3									
N/A / G4									
111 / N/A									

NOTES

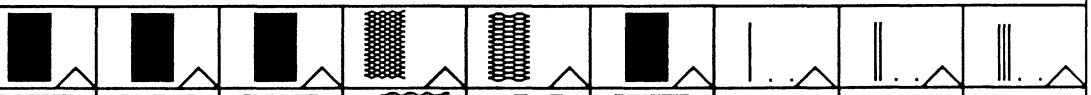
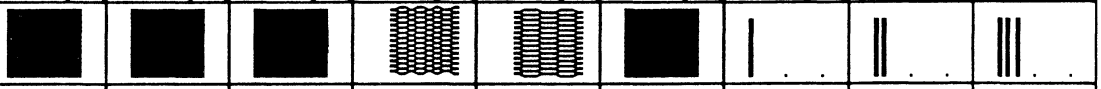
* ATTRIBUTE BYTE 2, BITS 2-0

** ATTRIBUTE BYTE 1, BIT 1 MUST BE SET FOR THIS MODE

△ CHARACTER CELL SHOWN LARGER THAN ACTUAL SIZE

CHARACTER		NONE	NONE	NONE	NONE	NONE	NONE	NONE	NONE	NONE
HEXADECIMAL		E1	E2	E3	E4	E5	E6	E7	E8	E9
DECIMAL		225	226	227	228	229	230	231	232	233
CHARACTER SET										
BASE TERMINAL *	OIL OPTION									
000	REG									
000 **	DW									
001	DH									
001 **	DS									
010	QS									
011	G1									
N/A	G2									
N/A	G3									
N/A	G4									
111	N/A									

NOTES
* ATTRIBUTE BYTE 2, BITS 2-0
** ATTRIBUTE BYTE 1, BIT 1 MUST BE SET FOR THIS MODE
△ CHARACTER CELL SHOWN LARGER THAN ACTUAL SIZE

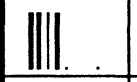
CHARACTER		NONE	NONE	NONE	NONE	NONE	NONE	NONE	NONE	NONE
HEXADECIMAL		EA	EB	EC	ED	EE	EF	F0	F1	F2
DECIMAL		234	235	236	237	238	239	240	241	242
CHARACTER SET										
BASE TERMINAL *	OIL OPTION									
000	REG									
000 **	DW									
001	DH									
001 **	DS									
010	QS									
011	G1									
N/A	G2									
N/A	G3									
N/A	G4									
111	N/A									

NOTES

* ATTRIBUTE BYTE 2, BITS 2-0

** ATTRIBUTE BYTE 1, BIT 1 MUST BE SET FOR THIS MODE

△ CHARACTER CELL SHOWN LARGER THAN ACTUAL SIZE

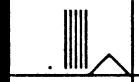
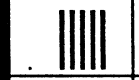
CHARACTER		NONE	NONE	NONE	NONE	NONE	NONE	NONE	NONE	NONE
HEXADECIMAL		F3	F4	F5	F6	F7	F8	F9	FA	FB
DECIMAL		243	244	245	246	247	248	249	250	251
CHARACTER SET										
BASE TERMINAL *	OIL OPTION									
000	REG									
000 **	DW									
001	DH									
001 **	DS									
010	QS									
011	G1									
N/A	G2									
N/A	G3									
N/A	G4									
111	N/A									

NOTES

* ATTRIBUTE BYTE 2, BITS 2-0

** ATTRIBUTE BYTE 1, BIT 1 MUST BE SET FOR THIS MODE

△ CHARACTER CELL SHOWN LARGER THAN ACTUAL SIZE

CHARACTER		NONE	NONE	NONE	NONE
HEXADECIMAL		FC	FD	FE	FF
DECIMAL		252	253	254	255
CHARACTER SET					
BASE TERMINAL *	OIL OPTION				
000	REG				
000 **	DW				
001	DH				
001 **	DS				
010	QS				
011	G1				
N/A	G2				
N/A	G3				
N/A	G4				
111	N/A				

NOTES

* ATTRIBUTE BYTE 2, BITS 2-0

** ATTRIBUTE BYTE 1, BIT 1 MUST BE SET FOR THIS MODE

△ CHARACTER CELL SHOWN LARGER THAN ACTUAL SIZE

Appendix D

QUICK REFERENCE GUIDE

Table D-1 Two-letter Abbreviations of ASCII Control Codes

Hexadecimal Code	ASCII Code	Two-letter Abbreviation
00	NUL	NL
01	SOH	SH
02	STX	SX
03	ETX	EX
04	EOT	ET
05	ENQ	EQ
06	ACK	AK
07	BEL	BL
08	BS	BS
09	HT	HT
0A	LF	LF
0B	VT	VT
0C	FF	FF
0D	CR	CR
0E	SO	SO
0F	SI	SI
10	DLE	DL
11	DC1 (XON)	D1
12	DC2	D2
13	DC3 (XOFF)	D3
14	DC4	D4
15	NAK	NK
16	SYN	SY
17	ETB	EB
18	CAN	CN
19	EM	EM
1A	SUB	SB
1B	ESC	EC
1C	FS	FS
1D	GS	GS
1E	RS	RS
1F	US	US

Table D-2 Membrane Keypad ASCII Codes

Key	Hexadecimal Code	ASCII Code	Notes
0	30	0	
1	31	1	
2	32	2	
3	33	3	
4	34	4	
5	35	5	
6	36	6	
7	37	7	
8	38	8	
9	39	9	
A	41	A	
B	42	B	
C	43	C	
D	44	D	
E	45	E	
F	46	F	
F1	47	G	
F2	48	H	
F3	49	I	
F4	4A	J	
F5	4B	K	
F6	4C	L	
	2E		
ENTER	0D	<CR>	
	Hazeltine	ANSI	
↑	7E 0C	<ESC>[A	In Hazeltine half-duplex, this key sends no code, but does move cursor on screen
↓	0A	<ESC>[B	
←	08	<ESC>[D	
→	10	<ESC>[C	In Hazeltine half-duplex, this key sends no code, but does move cursor on screen

Table D-3 Membrane Keypad Codes (Application Mode)
(4870)

Key	Hexadecimal Code(1) (Hazeltine 1500 Emulation)	ASCII Code(2) (ANSI Emulation)
0	B0	<ESC>Op
1	B1	<ESC>Oq
2	B2	<ESC>Or
3	B3	<ESC>Os
4	B4	<ESC>Ot
5	B5	<ESC>Ou
6	B6	<ESC>Ov
7	B7	<ESC>Ow
8	B8	<ESC>Ox
9	B9	<ESC>Oy
A	C1	<ESC>Oa
B	C2	<ESC>Ob
C	C3	<ESC>Oc
D	C4	<ESC>Od
E	C5	<ESC>Oe
F	C6	<ESC>Of
F1	C7	<ESC>Og
F2	C8	<ESC>Oh
F3	C9	<ESC>Oi
F4	CA	<ESC>Oj
F5	CB	<ESC>Ok
F6	CC	<ESC>Ol
	AE	<ESC>On
↑	91	<ESC>OA
↓	94	<ESC>OB
←	92	<ESC>OD
→	93	<ESC>OC
ENTER	8D	<ESC>OM

NOTES

(1) Same as Table D-2, except that bit 7 is set to 1, and the arrow codes differ
(2) Same as the codes returned by a VT-100 keypad in application mode

Table D-4 Codes for Keyboard Alphanumeric Keys
(Full and Half-duplex)

	no CTRL, no SHIFT		no CTRL, SHIFT		CTRL, no SHIFT		CTRL, SHIFT	
Key	Hex	ASCII	Hex	ASCII	Hex	ASCII	Hex	ASCII
A	61	a	41	A	01	<SOH>	01	<SOH>
B	62	b	42	B	02	<STX>	02	<STX>
C	63	c	43	C	03	<ETX>	03	<ETX>
D	64	d	44	D	04	<EOT>	04	<EOT>
E	65	e	45	E	05	<ENQ>	05	<ENQ>
F	66	f	46	F	06	<ACK>	06	<ACK>
G	67	g	47	G	07	<BEL>	07	<BEL>
H	68	h	48	H	08	<BS>	08	<BS>
I	69	i	49	I	09	<HT>	09	<HT>
J	6A	j	4A	J	0A	<LF>	0A	<LF>
K	6B	k	4B	K	0B	<VT>	0B	<VT>
L	6C	l	4C	L	0C	<FF>	0C	<FF>
M	6D	m	4D	M	0D	<CR>	0D	<CR>
N	6E	n	4E	N	0E	<SO>	0E	<SO>
O	6F	o	4F	O	0F	<SI>	0F	<SI>
P	70	p	50	P	10	<DLE>	10	<DLE>
Q	71	q	51	Q				
R	72	r	52	R				
S	73	s	53	S	See Table 3-5			
T	74	t	54	T				

Table D-4 (continued)

	no CTRL, no SHIFT		no CTRL, SHIFT		CTRL, no SHIFT		CTRL, SHIFT	
Key	Hex	ASCII	Hex	ASCII	Hex	ASCII	Hex	ASCII
U	75	u	55	U	15	<NAK>	15	<NAK>
V	76	v	56	V	16	<SYN>	16	<SYN>
W	77	w	57	W	17	<ETB>	17	<ETB>
X	78	x	58	X	18	<CAN>	18	<CAN>
Y	79	y	59	Y	19	<EM>	19	<EM>
Z	7A	z	5A	Z	1A	<SUB>	1A	<SUB>
					PC/Mem	Key	PC/Mem	Key
1	31	1	21	!	31 <DC1> 11	1	21 <SH> 01	!
2	32	2	40	@	32 <DC2> 12	2	00 <NULL> 00	<NULL
3	33	3	23	#	33 <DC3> 13	3	23 <ETX> 03	#
4	34	4	24	$	34 <DC4> 14	4	24 <EOT> 04	$
5	35	5	25	%	35 <NAK> 15	5	25 <ENQ> 05	%
6	36	6	5E	^	36 <SYN> 16	6	1E <RS> 1E	<RS>
7	37	7	26	&	37 <ETB> 17	7	26 <ALK> 06	&
8	38	8	2A	*	38 <CAN> 18	8	2A <LF> 0A	*
9	39	9	28	(	39 <EM> 19	9	28 <BS> 08	(
0	30	0	29	)	30 <DLE> 10	0	29 <HT> 09	)
Backspace	08	<BS>	08	<BS>	08 <BS> 08	<BS>	08 <BS> 08	<BS>
ESC	1B	<ESC>	1B	ESC	1B <ESC> 1B	<ESC>	1B <ESC> 1B	<ESC>
SPACE	20		20		20 <NULL> 00		20 <NULL> 00	
'	27	'	22	"	27 <BEL> 07	'	22 <STX> 02	"

Table D-4 (continued)

Key	no CTRL, no SHIFT		no CTRL, SHIFT		CTRL, no SHIFT		CTRL, SHIFT	
	Hex	ASCII	Hex	ASCII	Hex	ASCII	Hex	ASCII
					PC/Mem	Key	PC/Mem	Key
PRT SCN	2A	*	AA	*	2A <LF> 0A	*	AA <LF> 0A	*
,	2C	,	3C	<	2C <FF> 0C	,	3C <FS> 1C	<
-	2D	-	5F	_	1F <CR> 0D	<US>	1F <US> 1F	<US>
	2E		3E	>	2E <SO> 0E		3E <RS> 1E	>
/	2F	/	3F	?	2F <SI> 0F	/	3F <US> 1F	?
,	3B	,	3A		3B <ESC> 1B	,	3A <SB> 1A	
=	3D	=	2B	+	3D <GS> 1D	=	2B <VT> 0B	+
[	5B	[	7B	{	1B <ESC> 1B	<ESC>	1B <ESC> 1B	<ESC>
\	5C	\	7C	|	1C <FS> 1C	<FS>	1C <FS> 1C	<FS>
]	5D	]	7D	}	1D <GS> 1D	<GS>	1D <GS> 1D	<GS>
`	60	`	7E	~	60 <NULL> 00	`	7E <RS> 1E	~
<DEL>	7F	<DEL>	2E		7F <US> 1F	<DEL>	2E <SO> 0E	

Where

PC/Mem = PC/Membrane
< > = ASCII Character
nn = Hex character

Table D-5 Codes for Keyboard Control Keys

	Half-duplex		Full-duplex		Notes
Key	Hex	ASCII	Hex	ASCII	
TAB	09	<HT>	09	<HT>	
BACK SPACE	08	<BS>	08	<BS>	
DEL	See Table 3-3				
RETURN	0D	<CR>	OD	<CR>	
ENTER	0D	<CR>	0D	<CR>	
<left arrow>					
<right arrow>	See Table 3-5				
<up arrow>					
<down arrow>					
BREAK	00	<NUL>	00	<NUL>	(1)
ESC	1B	<ESC>	1B	<ESC>	
HOME			7E 12	~<DC2>	Haz emulation (2)(4)
			1B 5B 48	<ESC>[H	ANSI emulation (4)
F1					
F2	See Table 3-5				
F3					
F4					
F5 (CLEAR)			7E 1C	~<FS>	Haz emulation (2)(4)
			1B 4F 50	<ESC>OP	ANSI emulation (4)
F6 (CLEAR FOREGROUND)			7E 1D	~<GS>	Haz emulation (2)(4)
			1B 4F 51	<ESC>OQ	ANSI emulation (4)
F7 (CLEAR TO END OF LINE)			7E 0F	~<SI>	Haz emulation (2)(4)
			1B 4F 52	<ESC>OR	ANSI emulation (4)
F8 (CLEAR TO END OF SCREEN			7E 18	~<CAN>	Haz emulation (2)(4)
WITH FOREGROUND SPACES)			1B 4F 53	<ESC>OS	ANSI emulation (4)
F9	88		88		
F10					(3)

NOTES

(1) The communications line is held low (0) for 200-250 milliseconds
(2) This is not transmitted when in Hazeltine half-duplex mode
(3) Does not transmit a character
(4) When "4x7(A)" keypad is selected in ANSI mode, keys will respond in Hazeltine mode ANSI response not available on "4x7(A)"

Table D-6 Cursor Control and "F" Keys on Keyboard
(Full-duplex)

Key	Standard(1) Hazeltine 1500		Alternate(1) Hazeltine 1500		Standard(1) ANSI		Alternate(1) ANSI	
	Hex	ASCII	Hex	ASCII	Hex	ASCII	Hex	ASCII
CNTL-Q	7E 0C	~<FF>	11	<DC1>	1B 5B 41	<ESC>[A	11	<DC1>
CNTL-R	08	<BS>	12	<DC2>	1B 5B 44	<ESC>[D	12	<DC2>
CNTL-S	10	<DLE>	13	<DC3>	1B 5B 43	<ESC>[C	13	<DC3>
CNTL-T	0A	<LF>	14	<DC4>	1B 5B 42	<ESC>[B	14	<DC4>
up arrow	7E 0C	~<FF>	11	<DC1>	1B 5B 41	<ESC>[A	11	<DC1>
left arrow	08	<BS>	12	<DC2>	1B 5B 44	<ESC>[D	12	<DC2>
right arrow	10	<DLE>	13	<DC3>	1B 5B 43	<ESC>[C	13	<DC3>
down arrow	0A	<LF>	14	<DC4>	1B 5B 42	<ESC>[B	14	<DC4>
F1 (2)	11	<DC1>	7E 0C	~<FF>	11	<DC1>	1B 5B 41	<ESC>[A
F2 (2)	12	<DC2>	0A	<LF>	12	<DC2>	1B 5B 42	<ESC>[B
F3 (2)	13	<DC3>	08	<BS>	13	<DC3>	1B 5B 44	<ESC>[D
F4 (2)	14	<DC4>	10	<DLE>	14	<DC4>	1B 5B 43	<ESC>[C

NOTE

(1) The difference between standard and alternate is discussed in Chapter 2, Section 2 2 5, under "Keyboard Translation"

(2) Do not confuse the keyboard keys with F1-F6 keys on the keypad

Table 3-7 Numeric Pad (with NUM LOCK Off)(1)

Key	no SHIFT Hex	no SHIFT ASCII	SHIFT Hex	SHIFT ASCII
0	B0		30	0
1	B1		31	1
2	see down arrow (Table 3-5)		32	2
3	B3		33	3
4	see left arrow (Table 3-5)		34	4
5	35	5	35	5
6	see right arrow (Table 3-5)		36	6
7(2)	see home (Table 3-4)		37	7
8	see up arrow (Table 3-5)		38	8
9	B9		39	9

NOTE

(1) With NUM LOCK on, ASCII numbers from 0 through 9 will be generated, and the SHIFT will have no effect CTRL has no effect on the numeric keypad keys

(2) In the "NO SHIFT" mode, Key 7 is "HOME"

Table D-8 Process Graphic Symbols

Hex Value	ASCII Character	Process Control Symbol
20H		4x4 space
21H	!	motor in 4x3 cell
22H	"	not used
23H	#	left tank top in 4x1 cell
24H	$	right tank tope in 4x1 cell
25H	%	small diamond in 4x2 cell
26H	&	left tank bottom in 4x1 cell
27H	'	right tank bottom in 4x1 cell
28H	(	left arrow in 4x2 cell
29H	)	right arrow in 4x2 cell
2AH	*	small box in 4x2 cell
2BH	+	up valve in 4x2 cell
2CH	,	right/left facing valve in 4x2 cell
2DH	-	pump/compressor in 4x2 cell
2EH		up arrow in 4x2 cell
2FH	/	down arrow in 4x2 cell
30H	0	small circle in 4x2 cell
31H	1	circuit breaker type 1 in 2x4 cell
32H	2	fuse in 2x4 cell
33H	3	disconnect in 3x4 cell
34H	4	pump/blower in 4x2 cell
35H	5	circuit breaker type 2 in 4x2 cell
36H	6	left turbine in 3x2 cell
37H	7	right turbine in 3x2 cell
38H	8	left medium box in 4x2 cell
39H	9	right medium box in 4x2 cell
3AH	,	left medium circle in 4x3 cell
3BH		right medium circle in 4x3 cell
3CH	<	mini circle in 2x1 cell
3DH	=	mini left arrow in 2x1 cell
3EH	>	mini right arrow in 2x1 cell
3FH	?	mini up arrow in 2x1 cell
40H	@	mini down arrow in 2x1 cell
41H	A	motor
42H	B	large circle (left)
43H	C	large circle (right)
44H	D	tank top (left)
45H	E	tank top (right)
46H	F	small diamond
47H	G	large diamond (left)
48H	H	large diamond (right)
49H	I	tank bottom (left)
4AH	J	tank bottom (right)
4BH	K	left arrow
4CH	L	right arrow
4DH	M	small box

Table D-8 Process Graphic Symbols (CONT'D)

Hex Value	ASCII Character	Process Control Symbol
4EH	N	up valve
4FH	O	right/left facing valve
50H	P	pump/compressor
51H	Q	up arrow
52H	R	down arrow
53H	S	small circle
54H	T	transformer
55H	U	circuit breaker (type 1)
56H	V	fuse
57H	W	disconnect
58H	X	pump/blower
59H	Y	circuit breaker (type 2)
5AH	Z	turbine (left)
5BH	[	turbine (right)
5CH	\	large box (left)
5DH	]	large box (right)
5EH	^	medium box (left)
5FH	_ (underscore)	medium box (right)
60H	` (grave)	medium circle (left)
61H	a	medium circle (right)
62H	b	top left quarter of large circle in 4x2 cell
63H	c	top right quarter of large circle in 4x2 cell
64H	d	bottom left quarter of large circle in 4x2 cell
65H	e	bottom right quarter of large circle in 4x2 cell
66H	f	top left quarter of small circle in 2x1 cell
67H	g	top right quarter of small circle in 2x1 cell
68H	h	bottom left quarter of small circle in 2x1 cell
69H	i	bottom right quarter of small circle in 2x1 cell
6AH	j	small tank top in 4x1 cell
6BH	k	small tank bottom in 4x1 cell
6CH	l	mini tank top in 2x1 cell
6DH	m	mini tank bottom in 2x1 cell
6EH	n	mini diamond in 2x1 cell
6FH	o	mini box in 2x1 cell
70H	p	mini right valve in 2x1 cell
71H	q	mini up valve in 2x1 cell
72H	r	mini motor in 2x2 cell
73H	s	mini pump/blower in 2x1 cell
74H	t	mini transformer in 2x2 cell
75H	u	mini circuit breaker type 1 in 1x2 cell
76H	v	mini fuse in 1x2 cell
77H	w	mini disconnect in 1x2 cell
78H	x	mini blower/compressor in 2x1 cell
79H	y	mini circuit breaker type 2 in 2x1 cell
7AH	z	mini left turbine in 1x1 cell
7BH	{	mini right turbine in 1x1 cell

Table D-9 Utility Graphics

Character Codes	Graphics Description
32-79 (20-4FH)	Process Graphics Pieces
80-87 (50-57H)	Process Graphic Connectors (Thin)
88-95 (58-5FH)	Process Graphic Connectors (Thick)
96-111 (60-6FH)	Thick Line Graphics
112-175 (70-AFH)	Process Graphic Pieces
176-187 (B0-BBH)	Miscellaneous Connectors

Table D-10 Remote Commands
(Hazeltine 1500 Emulation)

Commands	ASCII	Hex
Control Characters		
Bell	<BEL>	07
Backspace	<BS>	08
Cursor to Next Foreground Field	<HT>	09
Linefeed	<LF>	0A
Carriage Return	<CR>	0D
Configuration Commands		
Enable Application Mode	~	7E 2E
Disable Application Mode	~ /	7E 2F
Cursor Off	~<S0H>	7E 01
Cursor On	~<STX>	7E 02
Scrolling Off	~<BEL>	7E 07
Scrolling On	~<BS>	7E 08
Unlock Keyboard	~<ACK>	7E 06
Lock Keyboard	~<NAK>	7E 15
Attribute Commands		
Set/Reset Attributes	~6<attribute#>*	7E 36 <attribute #>
Change Char Attributes	~<ETX> <attr-1> <attr-2>	7E 03 <attr-1> <attr-2>
Cursor Movement Commands		
Cursor Right (no scroll)	<DLE>	10
Return Cursor Position	~<ENQ>	7E 05
Cursor Down (no scroll)	~<VT>	7E 0B
Cursor Up	~<FF>	7E 0C
Cursor to X,Y	~<DC1> X Y	7E 11 X Y
Home Cursor	~<DC2>	7E 12

Table D-10 Remote Commands (continued)
(Hazeltine 1500 Emulation)

Commands	ASCII	Hex
Clear Commands		
Clear to EOL with Background Spaces	~<SI>	7E 0F
Clear to EOS with Background Spaces	~<ETB>	7E 17
Clear to EOS with Foreground Spaces	~<CAN>	7E 18
Clear Foreground	~<GS>	7E 1D
Clear Screen	~<FS>	7E 1C
Background Field Follows	~<EM>	7E 19
Foreground Field Follows	~<US>	7E 1F
Delete Commands		
Delete Line	~<DC3>	7E 13
Insert Line	~<SUB>	7E 1A
Draw Commands		
Draw Box	~<HT> <char> <xstart> <ystart> <xend> <yend>	7E 09 <char> <xstart> <ystart> <xend> <yend>
Draw Vertical Line (upward)	~<LF> <char> <xstart> <ystart> <length>	7E 0A <char> <xstart> <ystart> <length>
Draw Horizontal Line (left to right)	~<CR> <char> <xstart> <ystart> <length>	7E 0D <char> <xstart> <ystart> <length>
Draw Bar Chart	~<S0> <xstart> <ystart> <length1> <length2>	7E 0E <xstart> <ystart> <length1> <length2>
Draw Bar Chart Down	~<space> <xstart> <ystart> <length1> <length2>	7E 20 <xstart> <ystart> <length1> <length2>
Draw Bar Chart Right	~! <xstart> <ystart> <length1> <length2>	7E 21 <xstart> <ystart> <length1> <length2>
Draw Bar Chart Left	~" <xstart> <ystart> <length1> <length2>	7E 22 <xstart> <ystart> <length1> <length2>
Additional Commands		
Pause	~ # <time>	7E 23 <time>
Return Password	~ %	7E 25
Plot Point	~0XY	7E 30 X Y
Unplot Point	~1XY	7E 31 X Y

* See Section 5 4 32

Table D-11 Remote Commands (ANSI Emulation)

Control Characters

00 - ignored
05 - answer-back request
07 - ring bell
08 - move cursor left 1 position
09 - go to next tab stop
0A - linefeed or new line
0B - same as 0A
0C - same as 0A
0D - move cursor to left margin of current line (carriage return)
18 - cancel current ESC sequence
1A - same as 18
1B - ESC

Configuration Commands (See Notes 2 and 3)

ESC [? 7 h - enable autowrap
ESC [? 25 h - cursor on
ESC [? 7 l - disable autowrap
ESC [? 25 l - cursor off
ESC [? 4 h - smooth scroll
ESC [? 4 l - pop scroll

ESC [2 h - lock keyboard
ESC [2 l - unlock keyboard
ESC [20 h - enable auto line-feed
ESC [20 l - disable auto line-feed

ESC [= 1 h - cursor on
ESC [= 2 h - scrolling on
ESC [= 3 h - treat tab as ANSI tab
ESC [= 1 l - cursor off
ESC [= 2 l - scrolling off
ESC [= 3 l - treat tab as Hazeltine tab

Attribute Commands (See Note 1)

ESC [m - attributes off
ESC [0 m - attributes off
ESC [4 m - underline
ESC [5 m - blink
ESC [7 m - reverse video on
ESC [24 m - underline disable
ESC [25 m - blink disable
ESC [27 m - reverse video off

Table D-11 Remote Commands (continued)
(ANSI Emulation)

Attribute Commands (continued) (See Note 1)

ESC [50 m - select regular character set
ESC [51 m - select double-high characters
ESC [52 m - select quad-sized characters
ESC [53 m - select process control symbols
ESC [54 m - select double-wide characters
ESC [55 m - select double-size characters
ESC [56 m - select quad-sized characters
ESC [57 m - select utility graphics
ESC [1 ,attr1,attr2 p - change character attributes

Cursor Movement Commands

ESC [pn A - cursor up pn lines
ESC [pn B - cursor down pn lines
ESC [pn C - cursor right pn characters
ESC [pn D - cursor left pn characters
ESC [y,x H - cursor to position x,y
ESC [H - cursor home (1,1)
ESC [y,x f - cursor to position x,y
ESC [f - cursor home (1,1)
ESC D - cursor down with scroll
ESC M - cursor up with scroll
ESC E - cursor to beginning of next line with scroll
ESC 7 - save cursor and attributes
ESC 8 - restore cursor and attributes

Tab Stop Commands (See Note 4)

ESC H - set tab stop at current column
ESC [g - clear tab stop at current column
ESC [0 g - clear tab stop at current column
ESC [3 g - clear all tab stops

Clear Commands

ESC [pn X - clear pn characters on current line with background spaces
ESC [K - clear to end of line with background spaces
ESC [? K - clear to end of line with background spaces
ESC [0 K - clear to end of line with background spaces
ESC [? 0 K - clear to end of line with background spaces
ESC [1 K - clear to beginning of line with background spaces
ESC [? 1 K - clear to beginning of line with background spaces
ESC [2 K - clear entire line with background spaces
ESC [? 2 K - clear entire line with background spaces
ESC [J - clear to end of screen with background spaces

Table D-11 Remote Commands (continued)
(ANSI Emulation)

Clear Commands (continued)

ESC [? J - clear to end of screen with background spaces
ESC [0 J - clear to end of screen with background spaces
ESC [? 0 J - clear to end of screen with background spaces
ESC [1 J - clear to beginning of screen with background spaces
ESC [? 1 J - clear to beginning of screen with background spaces
ESC [2 J - clear entire screen with background spaces
ESC [? 2 J - clear entire screen with background spaces
ESC [8 p - clear to end-of-screen with foreground spaces
ESC [9 p - background follows
ESC [10 p - clear foreground
ESC [11 p - foreground follows

Insert/Delete Commands

ESC [pn L - insert pn blank line(s) at current cursor position
ESC [pn M - delete pn line(s) from cursor position
ESC [pn @ - insert pn space(s) in line at cursor position
ESC [pn P - delete pn character(s) from line at cursor position

Report Commands

05 <ENQ> - answer-back
device returns message - XYCOM TERMINAL followed by 4 spaces
ESC [5 n - device status report
device ok returns - ESC [0 n
device not ok returns - ESC [3 n
ESC [6 n - report cursor x,y position
returns - ESC [y,xR
ESC [c - return options
ESC [0 c - return options
returns - ESC [? 1,0c

Additional Commands

ESC c - reset to initial state
ESC = - select application mode for keypad keys
ESC > - select normal mode for keypad keys
ESC b - unlock keyboard
ESC ' - lock keyboard
ESC [18,time p - pause
ESC [20 p - return password

Table D-11 Remote Commands (continued)
(ANSI Emulation)

Draw Commands

ESC [2 ,char,ystrt,xstrt,yend,xend p	- draw box
ESC [3 ,char,ystrt,xstrt,length p	- draw vertical line
ESC [4 ,char,ystrt,xstrt,length p	- draw horizontal line
ESC [5 ,ystrt,xstrt,len1,len2 p	- draw bar chart up
ESC [25,ycor,xcor p	- plot point
ESC [26,ycor,xcor p	- unplot point
ESC [15,ystrt,xstrt,len1,len2 p	- draw bar chart down
ESC [16,ystrt,xstrt,len1,len2 p	- draw bar chart right
ESC [17,ystrt,xstrt,len1,len2 p	- draw bar chart left

NOTES

(1) Multiple attributes can be selected in a single attribute command
ESC [50,40,31m

(2) Multiple configurations can be specified in a single configuration command Example
ESC [= 1,2,3 h
ESC [? 7,25 h
ESC [2,20 h

(3) Configuration options that can be set by both the remote commands and th Configuration Menu are not saved on power-down unless the Configuration Menu is entered and exited

(4) Tab stops set/reset with remote commands are not saved on power-down unless the "S Tab Stop" menu is entered and exited

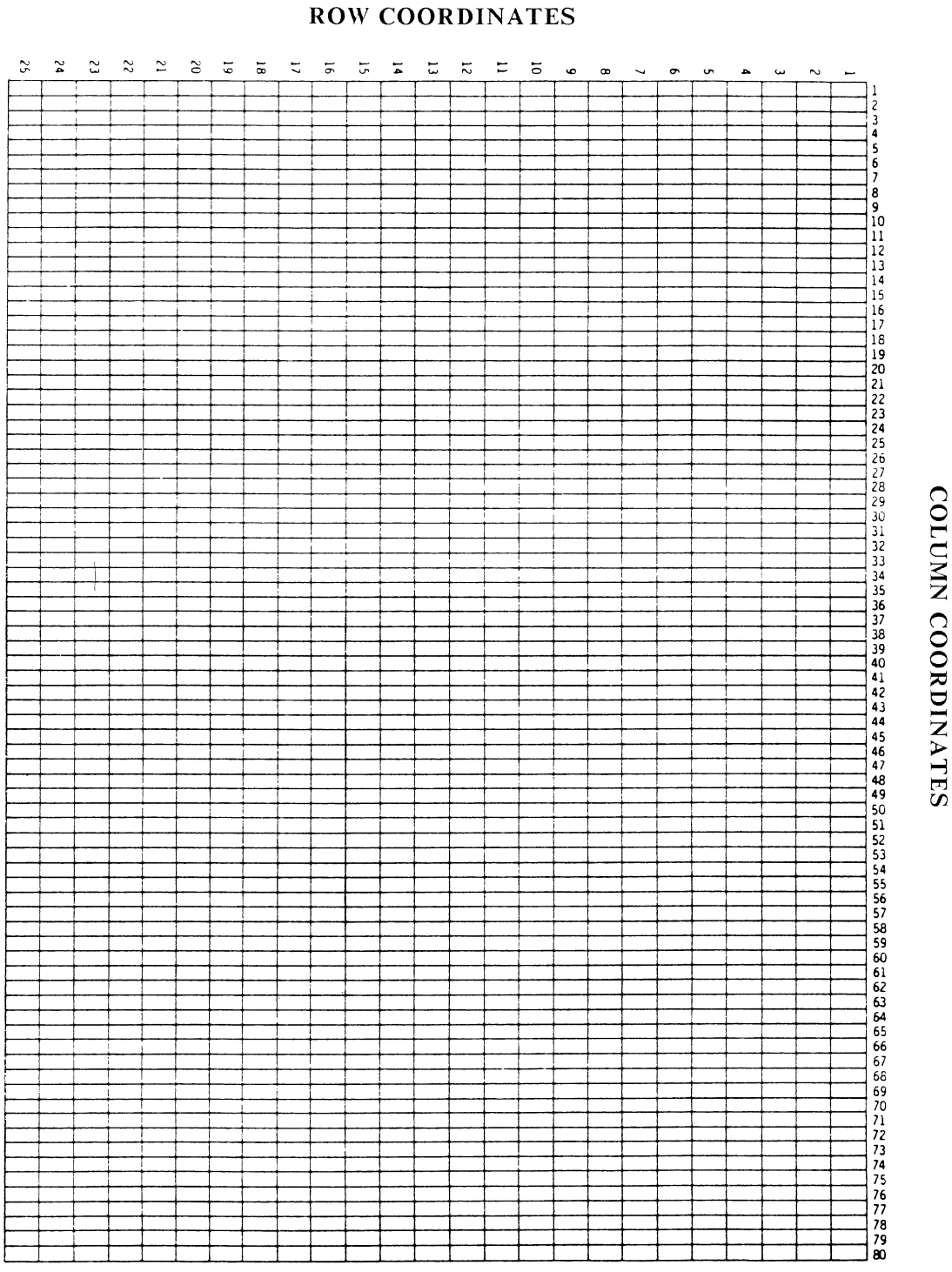


Figure D-1 Video Display Coordinate System (ANSI Emulation)

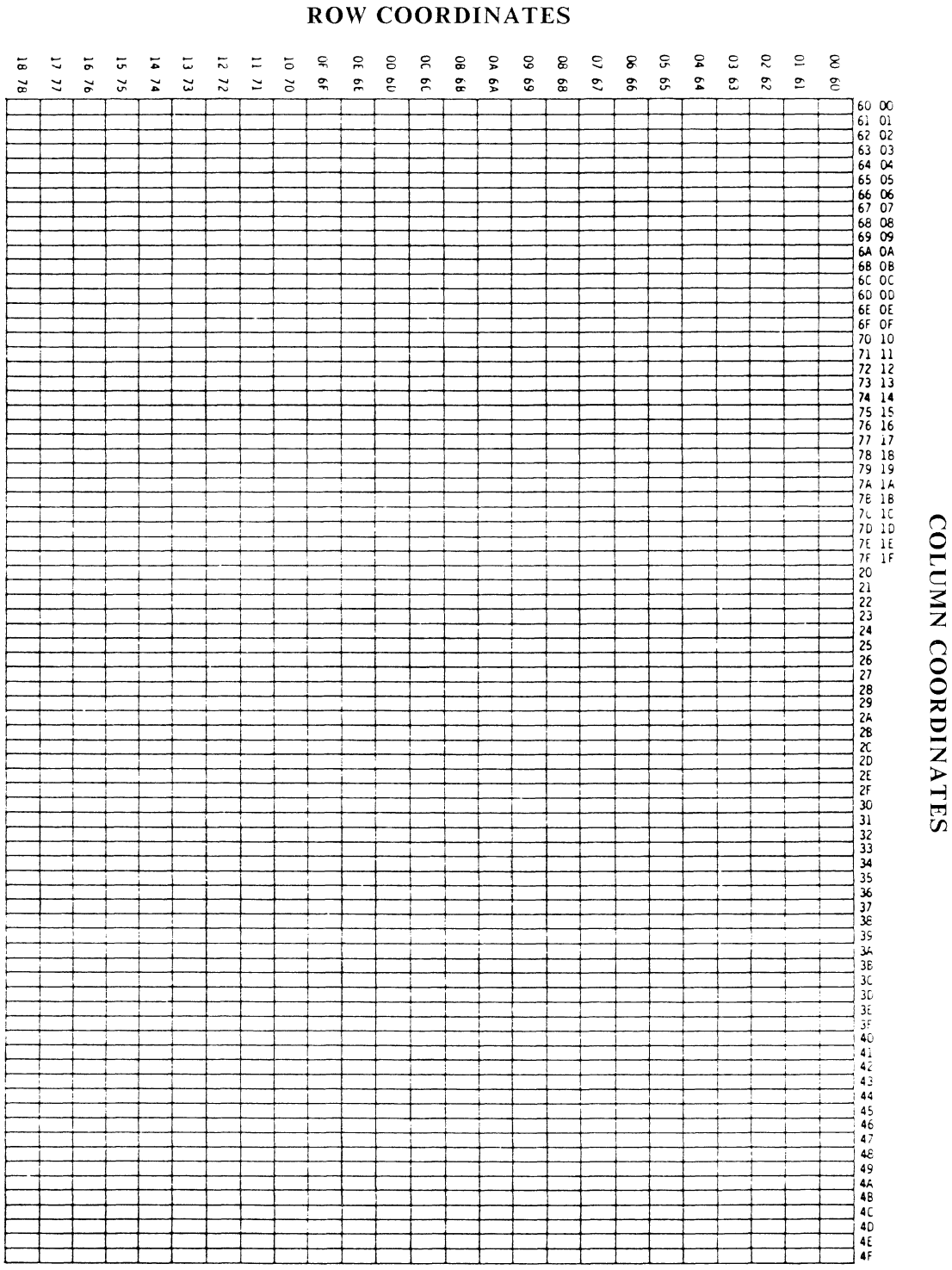


Figure D-2 Video Display Coordinate System (Hazeltine Emulation)