

**WYSE**  
TECHNOLOGY

**WY-100**

## REFERENCE MANUAL



---

**WYSE WY-100 DISPLAY TERMINAL  
REFERENCE MANUAL**

**WYSE TECHNOLOGY  
3040 North First Street  
San Jose, California 95134  
Phone: 408-946-3075  
TWX: 910-338-2251**

©Copyright WYSE TECHNOLOGY, 1981.  
Specification and information subject to  
change without prior notification.  
WYSE NO. 88-001-01 Rev. C

---

---

### **SAFETY WARNING**

Dangerous voltages are present when the terminal is on, and may remain after the power is off. Use caution when working on the internal electronics. Do not work alone.

The internal phosphor of the cathode ray tube is toxic. Use caution (safety goggles and gloves) whenever the cathode ray tube is handled. If the tube breaks exposing skin or eyes to the phosphor, rinse the affected area with cold water and consult a physician.

The terminal power cable is supplied with a safety ground. Do not use the terminal with an ungrounded outlet. Do not connect or disconnect the keyboard cable when the power is on.

### **DISCLAIMER**

Wyse Technology makes no representations or warranties with respect to the contents of this Manual and disclaims any implied warranties or fitness for any particular application. Wyse Technology reserves the right to revise this Manual without obligation of Wyse Technology to notify any person or organization of such revision.

### **COPYRIGHT NOTICE**

Copyright©1981 by Wyse Technology. All Rights Reserved Worldwide. No part of this publication may be reproduced without the express written permission of Wyse Technology.

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

---

---

## TABLE OF CONTENTS

	page
<b>SECTION 1. INTRODUCTION</b> .....	1-1
<b>SECTION 2. INSTALLATION</b> .....	2-1
2.1 Unpacking and Receiving Inspection .....	2-1
2.2 Site Preparation .....	2-1
2.3 Installation Verification .....	2-1
2.4 Interface Connectors Configuration .....	2-2
<b>SECTION 3. GENERAL DESCRIPTION</b> .....	3-1
3.1 Keyboard .....	3-1
3.2 Display Format .....	3-1
3.3 Communication Modes .....	3-2
3.4 Data/Text Entry Facilities .....	3-2
3.5 Split Screen Modes .....	3-2
3.5.1 Display Windows .....	3-2
3.5.2 Memory Organization .....	3-5
3.5.3 External and Internal Scrolling .....	3-9
3.5.4 Internal Paging .....	3-9
3.6 Automatic Data Validation .....	3-9
3.7 Automatic Word-Wrapping .....	3-10
3.8 Special Form Control .....	3-10
3.9 Screen Enhancement Attribute .....	3-10
3.9.1 Special Display Enhancement .....	3-10
<b>SECTION 4. KEYBOARD</b> .....	4-1
4.1 SHIFT, CAPS LOCK, and CTRL Keys .....	4-1
4.2 ESCape Key .....	4-1
4.3 Alpha-Numeric Keys .....	4-1
4.3.1 Rubout .....	4-1
4.3.2 Numeric Key Pad and ENTER Key .....	4-2
4.4 Cursor Control Keys .....	4-2
4.4.1 HOME .....	4-2
4.4.2 Cursor Right .....	4-2
4.4.3 Cursor Left .....	4-2
4.4.4 Cursor Up .....	4-2
4.4.5 Cursor Down .....	4-3
4.4.6 BACKSPACE .....	4-3
4.4.7 RETURN .....	4-3
4.4.8 LINEFEED .....	4-3
4.4.9 NEW LINE .....	4-3
4.4.10 TAB (unshifted) .....	4-3
4.4.11 TAB (shifted) .....	4-3
4.4.12 SET TAB (unshifted) .....	4-3
4.4.13 SET TAB (shifted) .....	4-3
4.5 Data/Text Editing Keys .....	4-3
4.5.1 DEL/INS CHAR (unshifted) .....	4-4
4.5.2 DEL/INS CHAR (shifted) .....	4-4
4.5.3 LINE INSERT .....	4-4
4.5.4 LINE DELETE .....	4-4
4.5.5 LINE ERASE (unshifted) .....	4-4
4.5.6 LINE ERASE (shifted) .....	4-4
4.5.7 PAGE ERASE (unshifted) .....	4-4

---

---

**Table of Contents (continued)**

4.5.8	PAGE ERASE (unshifted)	4-4
4.5.9	INS/REP (unshifted)	4-4
4.5.10	INS/REP (shifted)	4-4
4.5.11	SCRN EDIT (unshifted)	4-5
4.5.12	SCRN EDIT (shifted)	4-5
4.6	Block Transmission Keys	4-5
4.6.1	SEND LINE (unshifted)	4-5
4.6.2	SEND LINE (shifted)	4-5
4.6.3	SEND PAGE (unshifted)	4-5
4.6.4	SEND PAGE (shifted)	4-5
4.6.5	SOM/EOM (unshifted)	4-5
4.6.6	SOM/EOM (shifted)	4-6
4.6.7	SEND MSG (unshifted)	4-6
4.6.8	SEND MSG (shifted)	4-6
4.7	BREAK Key	4-6
4.8	Programmable Function Keys	4-6
4.8.1	FUNCT	4-6
4.9	Local Print Keys	4-6
4.9.1	PRINT (unshifted)	4-6
4.9.2	PRINT (shifted)	4-6
4.10	SCRL Up	4-6
4.11	SCRL Down	4-7
4.12	PAGE Key	4-7
4.12.1	Internal Paging	4-7
4.12.2	Split Display Window Selection	4-7
4.13	RESET (unshifted)	4-7
4.13.1	RESET (shifted)	4-7
4.14	Key-click Control	4-7
<b>SECTION 5.</b>	<b>COMMUNICATION CONTROL</b>	<b>5-1</b>
5.1	ASCII Control Codes	5-1
5.2	ESCape Sequences	5-1
5.2.1	Cursor Set/Read	5-1
5.2.2	Keyboard Lock/Unlock	5-2
5.2.3	Communication Modes Controls	5-2
5.2.4	Text Editing Modes Controls	5-2
5.2.5	Protect Form Controls	5-2
5.2.6	Special Cursor Controls	5-2
5.2.7	Column Tabulation Controls	5-3
5.2.8	Data/Text Editing Controls	5-3
5.2.9	Scrolling and Paging Controls	5-3
5.2.10	Screen Attributes and Graphics	5-3
5.2.11	Message Fields and Labeling Fields	5-3
5.2.12	Display Mode Selection	5-4
5.2.13	Split Segment Selection	5-4
5.2.14	Vertical Split Column	5-4
5.2.15	Block Transmission	5-4
5.2.16	Local Print Controls	5-5
5.2.17	Programming the Function Keys	5-5
5.2.18	Two Character Function Key Sequences and Modified Cursor Up Key	5-5
5.2.19	Read Cursor Byte	5-5
5.3	Remote Computer Handshaking Protocol	5-6
<b>SECTION 6.</b>	<b>TERMINAL SELF-TEST FACILITIES</b>	<b>6-1</b>

<b>APPENDIX A. SETTING OF USER PROGRAMMABLE SWITCHES</b> .....	<b>A-1</b>
<b>APPENDIX B. TOP MESSAGE FIELD LINE AND BOTTOM FUNCTION KEY LABELING LINE</b> .....	<b>B-1</b>

**TABLES**

TABLE 1.	Terminal Control Codes .....	T-1
TABLE 2.	ASCII Codes .....	T-2
TABLE 3.	Default Function Key Code .....	T-2
TABLE 4.	Graphic Codes .....	T-3
TABLE 5.	Cursor Row or Column Code .....	T-3
TABLE 6.	Screen Attribute Codes .....	T-4
TABLE 7.	Vertical Split Column 40 Separation Code .....	T-5
TABLE 8.	Display Mode Selection .....	T-6
TABLE 9.	Terminal Escape Sequences .....	T-6
TABLE 10.	Keyboard Command Code Sequences .....	T-7

**LIST OF ILLUSTRATIONS**

Figure 1-1.	Specification .....	1-1
Figure 2-1.	Terminal Back Panel .....	2-2
Figure 2-2.	Internal Jumper Connections .....	2-2
Figure 2-3.	Switch Settings .....	2-3
Figure 3-1.	Terminal Display Arrangement .....	3-1
Figure 3-2.	Terminal Display Window Format .....	3-3
Figure 3-3.	Normal Display Mode With 3840 Characters Storage Memory .....	3-6
Figure 3-4.	Horizontal Split Display Model With 3840 Characters Storage Memory .....	3-7
Figure 3-5.	Vertical Split Display Mode With 3840 Characters Storage Memory .....	3-8
Figure 3-6.	Terminal Field Data Validation Arrangement .....	3-11
Figure 4-1.	Keyboard Layout .....	4-1

---

---

## SECTION 1. INTRODUCTION

The terminal is a microprocessor based smart terminal which provides high level performance at a low cost. It is designed for use in communicating information with most host computers in many different applications.

This manual is designed to aid the users in installing, operating, and programming the terminal. The manual is divided into sections based on these functions. The following paragraphs give a brief description of the scope of the sections.

**Section 1 - Specifications**

This section contains the introduction and product specifications.

**Section 2 - Installation**

This section contains detailed information on the receiving, unpacking, site preparation, and installation procedures for operating the terminal.

**Section 3 - General Description**

This section describes the terminal, its features and applications.

**Section 4 - Keyboard**

This section contains detailed information about the keyboard, its layout and the use of the alphanumeric keys, cursor control keys and function keys.

**Section 5 - Communication Control**

This section contains specific information on the use of the command sequences.

**Section 6 - Terminal Self-Test Facilities**

This section provides information on procedures for trouble-shooting terminal problems.

Additionally, there are appendices covering the various codes available in the terminal as well as a listing of the tables and illustrations in the manual.

### FIGURE 1.1

### SPECIFICATIONS

Power	
Standard	120 VAC 60 Hz, 60 watts (205 BTU/hour) approximately
Optional	220/240 VAC 50 Hz, 60 watts (205 BTU/hour) approximately
Dimensions	Keyboard - 2.6"H x 20.5"W x 7.1"D (6.604cm x 52.07 cm x 18.034cm) Display Cabinet - 12.5"H x 15"W x 12.5" D (31.75cm x 38.1 cm x 31.75cm)
Shipping Weight	38 lbs. (17.24kg) approximately
Finish	Beige and black
Keyboard	Detachable with 6' (1.8288m) coiled cable 105 keys, cursor control pad, numeric pad, and function keys
CRT	12" (30.48cm) diagonal with non-glare display
Display attributes	Dim, reverse, underscore, blink, blank, and combinations
Display format	24 rows x 80 characters per row plus two rows for message and function key identification
Character Matrix	8 x 10 matrix in a 10 x 11 cell
Character set	128 characters with graphics (96 displayable ASCII and 32 control codes)
Memory Capacity	1 page (supports 1920 characters); 2 page optional (supports 3840 characters)
Communication	
Type	EIA RS-232C standard 20 MA current loop optional
Baud Rates	50, 75, 110, 134.5, 150, 300, 600, 1200, 1800, 2000, 2400, 3600, 4800, 7200, 9600
Data size	7 or 8 bits
Stop bits	1 or 2 stop bits
Parity	Odd, even, mark or space
Operating temperature	32° - 122°F (0° - 50°C)
Storage temperature	-40° - +140°F (-40° - +60°C)
Humidity	10% - 90% without condensation
Altitude	Sea level - 15,000' (4.572m)

**SECTION 2. INSTALLATION**

**2.1 UNPACKING AND RECEIVING INSPECTION**

Each terminal is shipped in one container which encloses the display unit, keyboard, pedestal base, power cable, and the Terminal Reference Manual. The terminal is enclosed with preformed styrofoam inserts designed to provide maximum protection during shipping.

After receiving the terminal, the shipping containers should be inspected carefully for damage. All external damage should be noted on the waybill and should be acknowledged by the delivery agent. A written report should be given to the transfer company or carrier. If there is no external damage to the containers, unpack the terminals and inspect for internal damage.

Sharp instruments should NOT be used to open the container. Remove the terminal carefully and inspect it thoroughly for damage such as loose modules or components. *If any damage exists, notify the transfer company immediately. The company is not responsible for shipping damage.* Any repairs made for damages which occurred during shipping are billable to the customer. All packaging materials should be saved for possible future use in reshipping the terminal.

**2.2 SITE PREPARATION**

The terminal has been specially designed to conform to most office environments. A table or desk with a nearby three pronged electric outlet supplying the required power (110 VAC 60 Hz or 220 VAC 50 Hz) and a communication interface located in the vicinity are sufficient to set up the terminal. Additional surface space is necessary if a local printer is also required.

**2.3 INSTALLATION VERIFICATION**

The terminal on/off rotary switch in the front should be in the fully counter clock-wise or off position before the terminal is plugged into the outlet. The following list should be checked before turning on the terminal:

1. Communication and printer cables are installed properly?
2. Power cable correctly plugged in? (Check serial # tag on back panel for 110 VAC or 220 VAC.)
3. Keyboard cable is secured?
4. Switch settings all correct? Baud rate? Parity? Stop bits? 50/60Hz? Communication mode? Blink cursor? 1 or 2 pages? (See Figure 2-3 and Appendix A.)

Note: SW1 switch settings cannot be all OFF (see Section 6).

**2.4 INTERFACE CONNECTORS CONFIGURATION**

There are two communication interface connectors on the terminal. Facing the terminal from the back, the connector on the left is for communicating with the host computer and the connector on the right is for sending data to the local printer (see Figure 2-1). Their configurations are as follows:

Host Computer Interface Connector Configuration:

<b>RS-232 C SIGNALS:</b>	
<b>PIN #</b>	<b>SIGNAL</b>
1	Frame ground
2	Transmit Data (output)
3	Receive Data (input)
4	Request To Send (output)
5	Clear To Send (input)
7	Signal Ground
8	Data Carrier Detect (input)
20	Data Terminal Ready (output)

**PRINTER INTERFACE CONNECTOR CONFIGURATION:**

<b>PIN #</b>	<b>SIGNAL</b>
1	Frame ground
3	Transmit Data (output to printer)
7	Signal ground
20	Printer ready (input from printer)

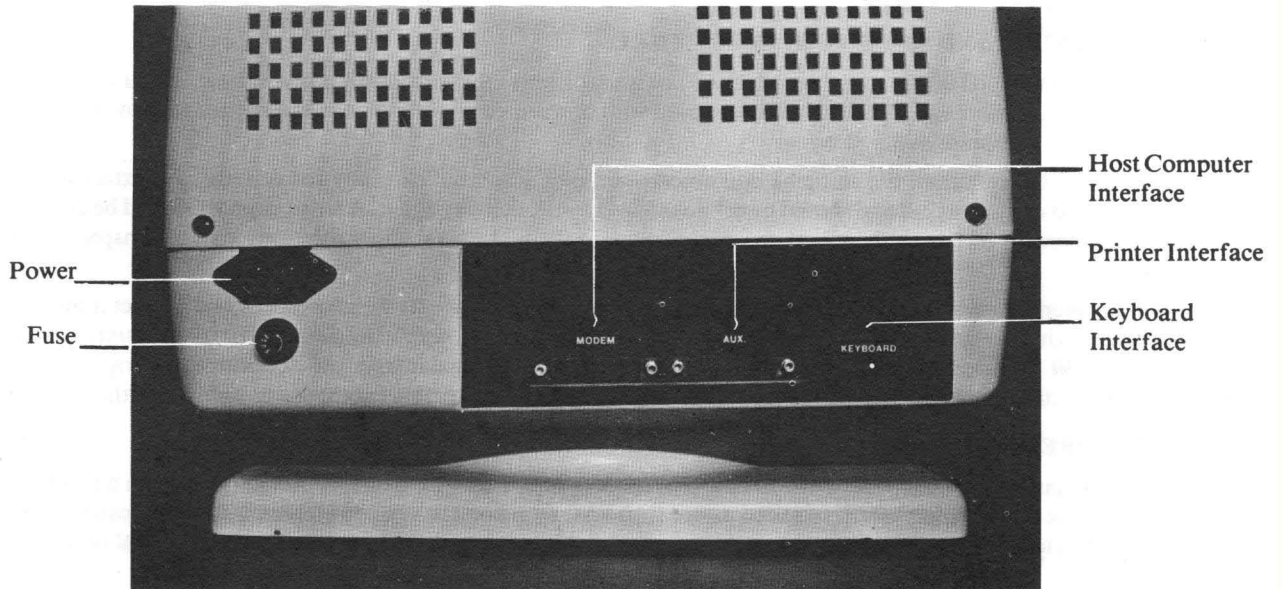
**CURRENT LOOP SIGNALS:\***

<b>PIN #</b>	<b>SIGNAL</b>
7	Signal ground
9	270-ohm resistor to +12 volts
10	Receive data 1 (input)
11	Receive data 2 (input)
12	Transmit data 1 (output)
13	Transmit data 2 (output)
14	270-ohm resistor to +12 volts

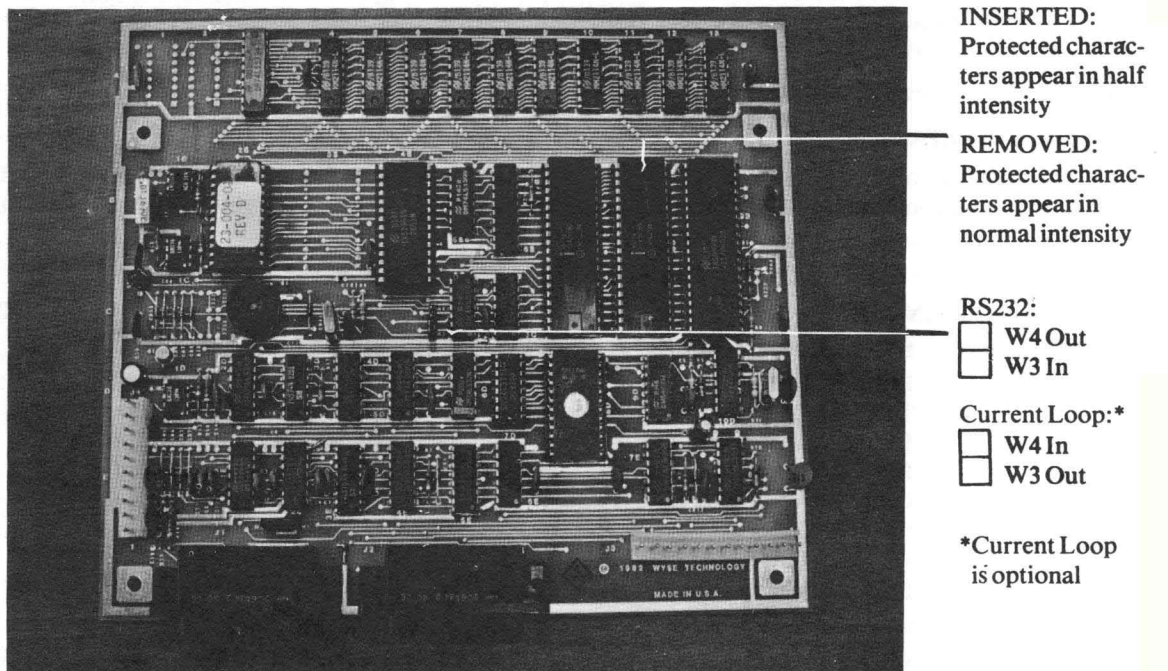
*\*Notes on current loop operation:*

1. Receive data 1 and 2 and Transmit data 1 and 2 are not polarity specific, i.e., they are interchangeable.
2. The 270-ohm resistors in conjunction with the signal ground can be used to provide active current sources for the current loop.
3. Both receive and transmit signals are opto-isolated, and can withstand  $\pm 750$  volts with respect to signal ground.



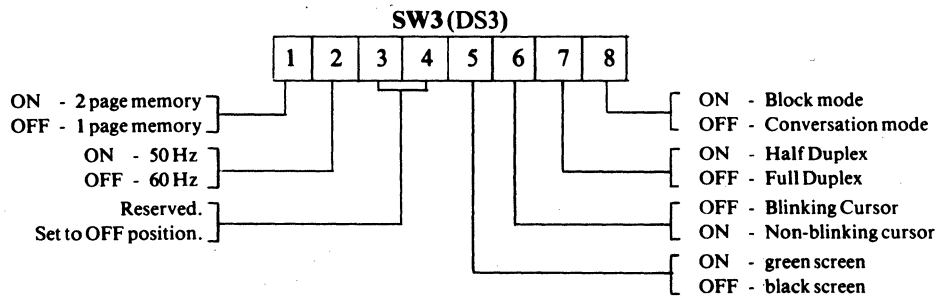
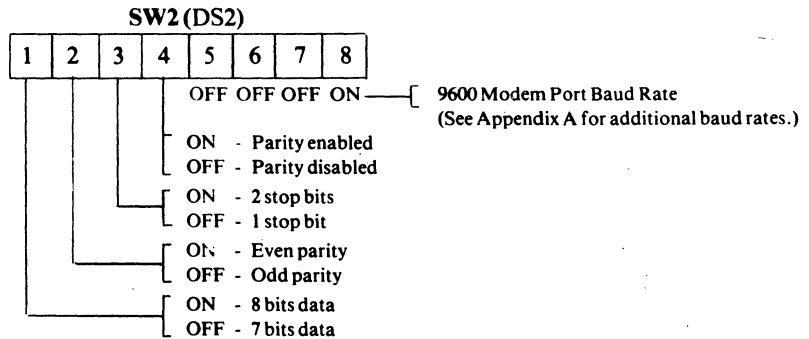
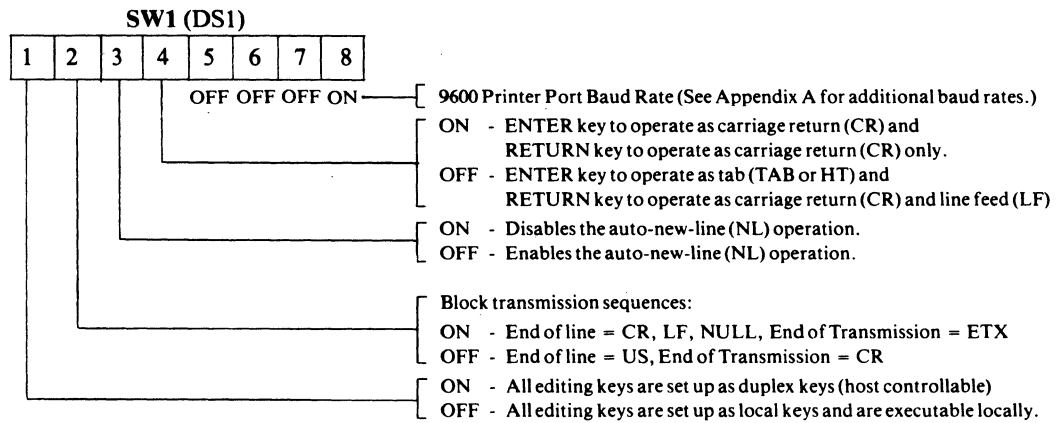


**FIGURE 2-1**  
**TERMINAL BACK PANEL**



**CONTROL LOGIC BOARD**

**FIGURE 2-2**  
**INTERNAL JUMPER CONNECTION**



See Appendix A for additional details & some examples.

**FIGURE 2-3**  
**SWITCH SETTINGS**

## GENERAL DESCRIPTION

### SECTION 3. GENERAL DESCRIPTION

The display terminal consists of a detached keyboard, a CRT display, internal microprocessor controlled video and logic, a communication interface, and a local printer interface. It allows the user full control in entering data/text onto the display and in communicating information with a host computer. Displayed data/text can be printed on a local printer via the standard local printer interface.

#### 3.1 KEYBOARD

The keyboard resembles a conventional typewriter with additional keys for local editing and remote communication with a computer. When used in combination with other keys, many of these keys can control several operations. The 105-key detachable keyboard is organized into four sections (see Table 4-1):

- the alphanumeric section
- the cursor control section
- the numeric data entry section
- the local editing and remote function section

#### 3.2 DISPLAY FORMAT

The terminal screen is organized into 24 rows of 80 characters each plus two rows for message and function key identification (see Figure 3-1).

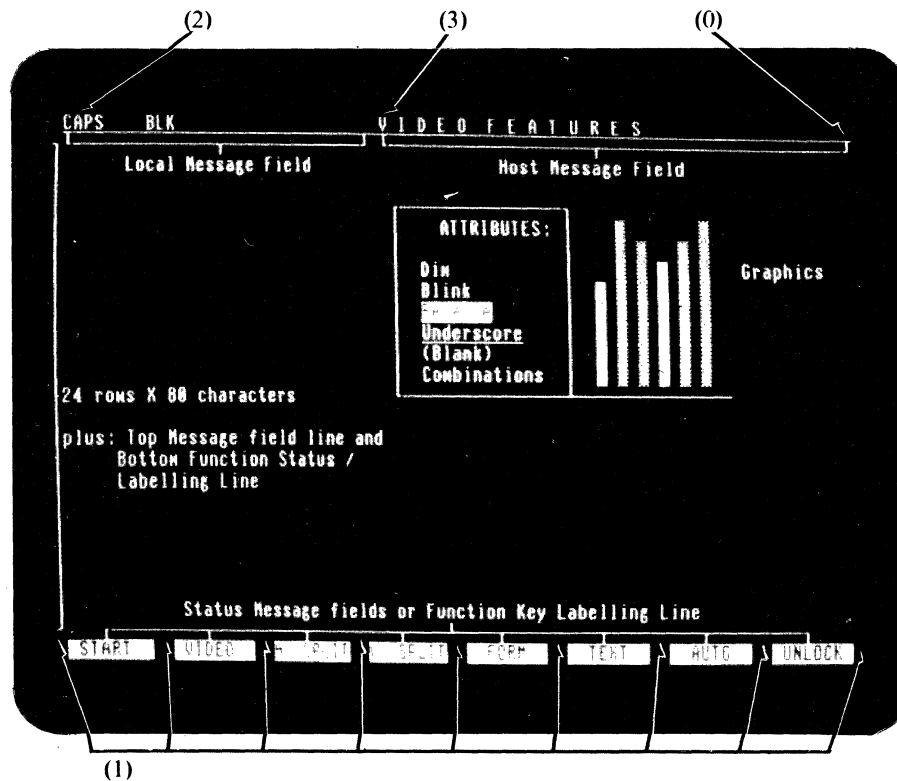


FIGURE 3-1

### TERMINAL DISPLAY ARRANGEMENT

The display contains 4 special display attributes which can be set in any combinations (see ESC A sequence in Section 5.2.10). Their relative positions on the screen are highlighted by (0), (1), (2), and (3). These special attributes are as follows:

- (0) Data/Text Entry Area start attribute.
- (1) Function Key Labeling Line Field separation attribute.
- (2) Local Message Field start attribute.
- (3) Host Message Field start attribute.

The bottom row consists of 8 labeling fields which can be used to display function key descriptions or any other status messages. The 8 labeling fields can also be used contiguously as one message field line. This row is the Function Key Labeling Line (see Appendix B).

The top row consists of two message fields and is called the Message Field Line. The two message fields are the Local Message field and the Host Message Field (see Appendix B).

The middle 24 rows of the display compose the data/text entry area which is the normal work space for the operator. This area is used to display data entered by the operator or data received from the host computer.

The terminal is capable of storing up to 48 rows of 80 characters each, a total of 3840 characters. The data/text entry area can be programmed in either normal, horizontal or vertical split screen display format for displaying the internal text storage (see Section 3.5).

Special display attributes can be used to enhance the Local Message Field, the Host Message Field, the Function Key Labeling Line and the data/text entry area. These enhancement attributes can be used to set up special effects on the screen (such as a reverse and underscored data/text entry area), or they can be used to hide a certain portion of the screen (such as the Local Message Field) from the operator (see Section 3.9).

### 3.3 COMMUNICATION MODES

The terminal can operate in one of three different communication modes: block, half duplex or full duplex conversation (or TTY mode).

Under block mode, text entry and editing are locally conducted to reduce the host computer overhead and communication line (e.g. phone line) hook-up time. No transmission to the computer is effected except when the break key, the function keys, or any one of the 'SEND' keys are depressed.

Under half duplex conversation mode, each character entered is sent to the host computer and displayed. Control keys are acted upon and are simultaneously sent to the computer. Local editing keys (e.g. DEL CHAR) are executed and their corresponding command sequence (ESC sequence) can be sent to the host computer if desired (see Appendix A and Table 10).

Under full duplex conversation mode, each character or control key entered is sent to the host computer but is not acted upon locally.

A monitor submode is available for the display and transmission of non-text/control codes to and from the host computer (see Section 5.2.3).

### 3.4 DATA TEXT ENTRY FACILITIES

For data entry applications, the terminal can be programmed so that fields of data can either be unprotected or protected. For example, the terminal can operate in a protect mode where only unprotected fields are accessible by the terminal operator. Special commands and line graphics capability are also provided to establish data entry 'forms' on the screen. The terminal also provides the user with a comprehensive set of text editing features:

- Character replace, insert, or delete
- Line delete or insert
- Clear partial line or text body
- Typewriter style column tabbing

In addition, the terminal provides the user with two advanced features which are normally available only in high-end terminals:

- Programmable local data entry validation to reduce computer overhead in checking for erroneous data, (requires second page option).
- Automatic word-wrapping to facilitate text generation. Terminal user does not have to worry about carriage return/line feed in the course of data entry.

### 3.5 SPLIT-SCREEN MODES

The terminal supports various display modes. The following sections describe their formats and memory organization.

#### 3.5.1. DISPLAY WINDOWS

The terminal can be programmed by the user or host computer to operate in one of four different display formats (Figures 3-2a, 3-2b, 3-2c, 3-2d):

**GENERAL DESCRIPTION**

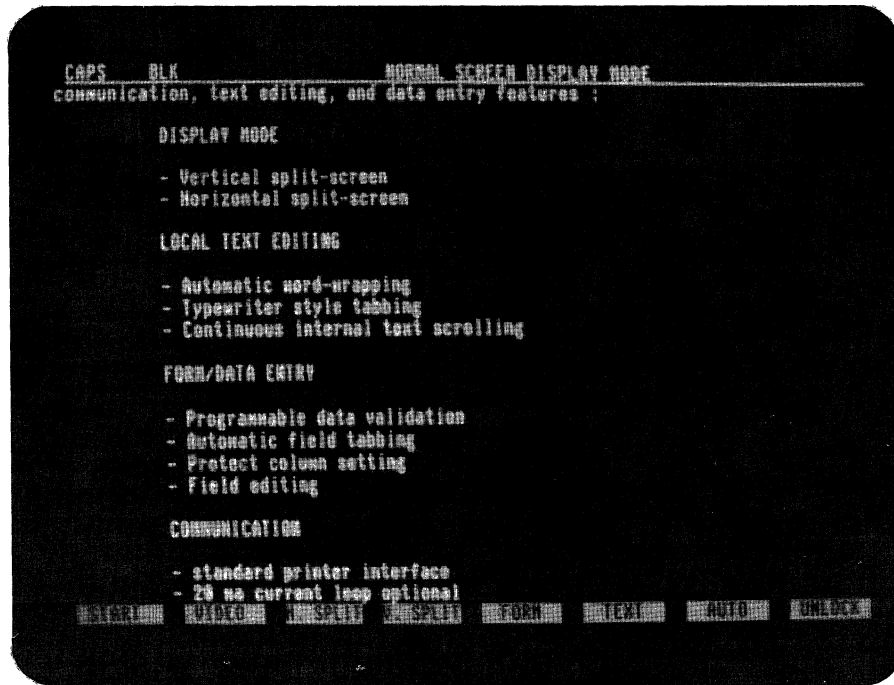


Fig. 3-2a Normal Display

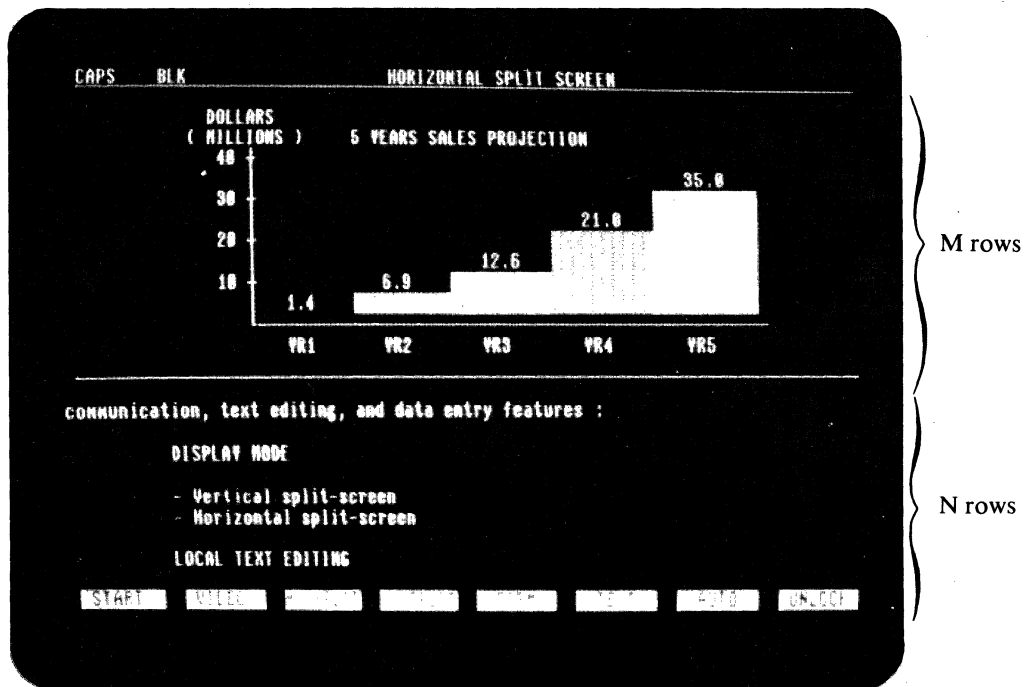
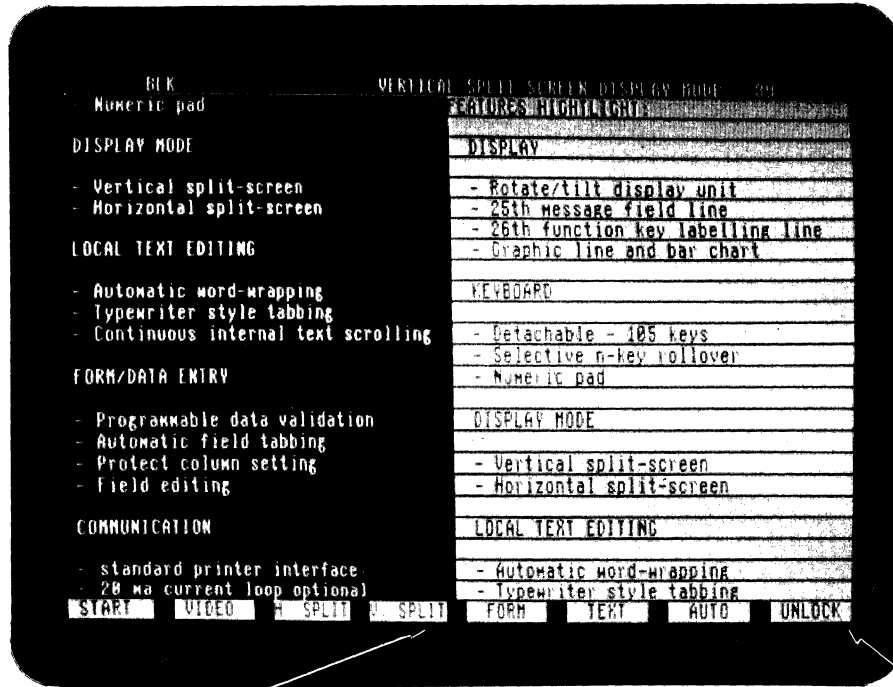


Fig. 3-2b Horizontal Split  
\*M + N = 24 rows, selectable.

**FIGURE 3-2  
TERMINAL DISPLAY WINDOWS' FORMATS**



\* Fig. 3-2c Vertical Split (39)  
 \*Vertical split column for separation and special screen effect enhancement. See also special attribute enhancement in Figure 3-1.

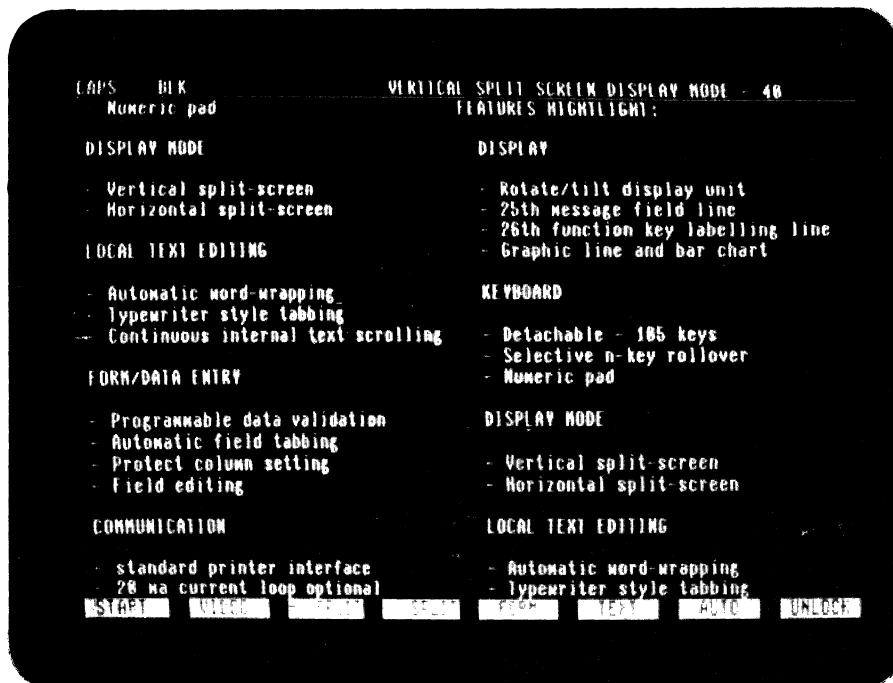


Fig. 3-2d Vertical Split (40)

FIGURE 3-2 (cont.)  
 TERMINAL DISPLAY WINDOWS FORMATS

## GENERAL DESCRIPTION

---

1. Normal display window of 24 rows of 80 characters each
2. Horizontal split-screen format with upper and lower display windows organized into:  
Upper window: M rows X 80 characters  
Lower window: N rows X 80 characters  
*where M can be programmed to be any number from 1 to 24, N can be programmed to be any number from 0 to 23, and  $M + N = 24$  (see Table 8).*
3. Vertical split-screen format with left and right display windows organized into:  
Left window: 24 rows X 39 characters  
Right window: 24 rows X 39 characters
4. Vertical split-screen format with left and right display windows organized into:  
Left window: 24 rows X 40 characters  
Right window: 24 rows X 40 characters

These display formats can be selected by command sequences from the keyboard or the host computer (see Section 4.12.2). The 'Normal' display format is similar to most conventional terminals supporting only one text segment internally in the terminal.

The split display formats support two text segments simultaneously in the terminal. However, only one display window can be selected to be active at one time either by the local shifted 'PAGE' key or by special 'ESCAPE' sequences from the host computer.

The active display window is identified by the current screen cursor position. All data/text editing and communication operations are performed on the active display window only. The inactive window is not affected.

Under the vertical split-screen format with text segments organized into 39 character columns there is a separation column at columns 40 and 80 of the screen which can be used to separate the two split windows. In addition to the default vertical separation line, screen effect attributes can also be used to highlight the two split windows (see Sections 3.9.1 and 5.2.14 on setting the vertical split column).

### 3.5.2 MEMORY ORGANIZATION

The terminal supports an internal display memory of either 1920 characters (standard) or 3840 characters (optional). Together with the 4 display window formats described previously, the terminal display memory can be programmed into 8 different storage formats (see Section 5.2.12).

In the 'normal' display format, the internal display memory is organized as one contiguous text segment. In any of the split-screen formats, the display memory is organized as 2 text segments, namely segment 0 and segment 1:

DISPLAY MODE SELECTION	MEMORY STORAGE SEGMENT	DISPLAY WINDOW
Horizontal Split:	Split Segment 0	Upper display window
	Split Segment 1	Lower display window
Vertical Split:	Split Segment 0	Left display window
	Split Segment 1	Right display window

If the terminal supports a 1920 character display memory, each text segment is equal in size to its corresponding display window. However, when the terminal supports 3840 characters of display memory, all text segments except that of Split Segment 0 of the horizontal split are larger in size than their corresponding display windows. Internal scrolling and paging capabilities are provided to position the desired portion of the text segment within the display window (see Figures 3-3, 3-4, 3-5).

Only one text segment is active at a time. The cursor location identifies the active text segment. All keyboard and communication functions in subsequent sections apply only to the active text segment. The following notations will be used frequently in the subsequent discussion about cursor movement:

Start-of-text:	<i>the first character in the text.</i>
Start-of-text row:	<i>the first row of characters in the text.</i>
End-of-text:	<i>the last character in the text.</i>
End-of-text row:	<i>the last row of characters in the text.</i>
Start-of-row or start-of-line:	<i>the first character of a row, usually the row where the cursor is.</i>
End-of-row or end-of-line:	<i>the last character of a row, usually the row where the cursor is.</i>

Display Window:

Storage Memory: One Text Segment

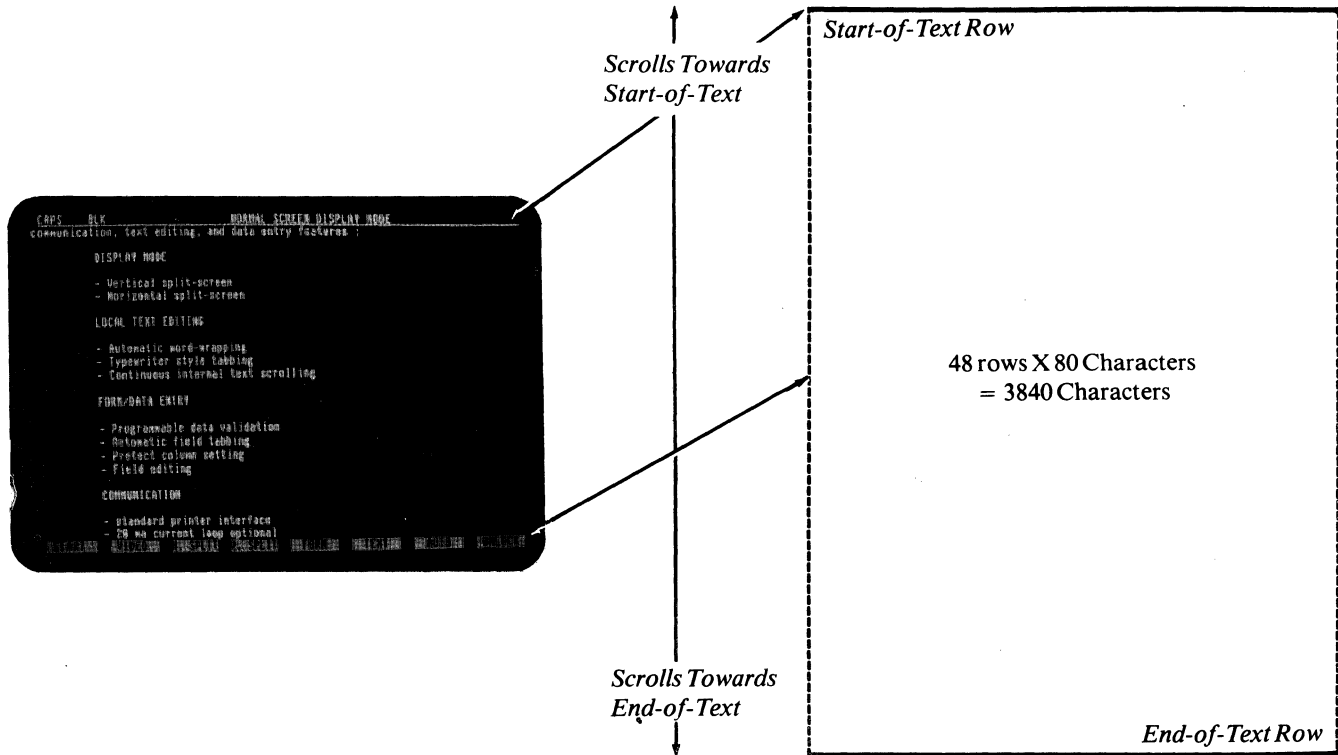
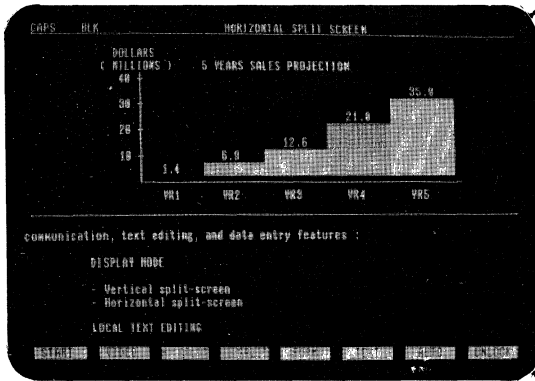


FIGURE 3-3  
NORMAL DISPLAY MODE WITH 3840 CHARACTERS STORAGE MEMORY



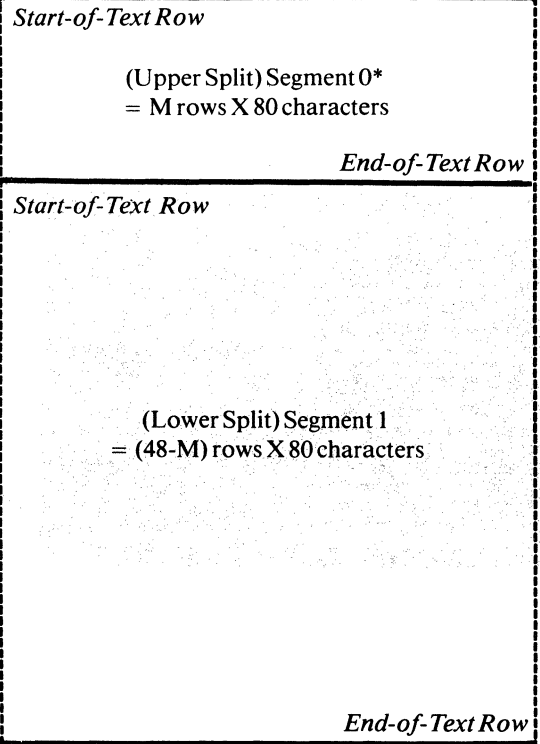
**GENERAL DESCRIPTION**

Display Window:



Storage Memory: Two Text Segments  
Total Storage = 3840 characters

External scroll  
only for upper  
split



\*Size of the Upper Split Segment (Segment 0) is equal to the Upper Split Window.

**FIGURE 3-4**

**HORIZONTAL SPLIT DISPLAY MODE WITH 3840 CHARACTERS STORAGE MEMORY**

Display Window:

Both display windows can perform scrolling to review text within the corresponding text segment.

Storage Memory: Two Text Segments  
Total Storage = 3840 characters

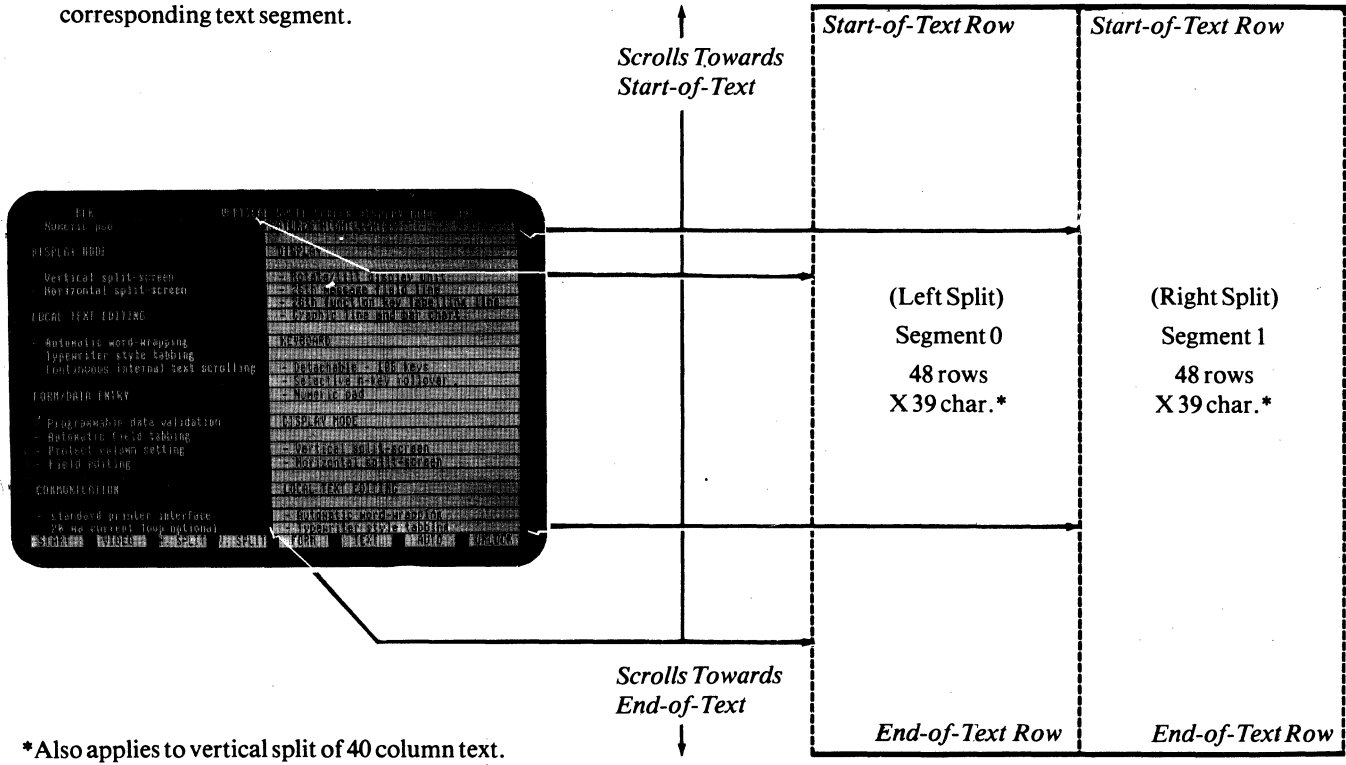


FIGURE 3-5  
VERTICAL SPLIT DISPLAY MODE WITH 3840 CHARACTERS STORAGE MEMORY

## GENERAL DESCRIPTION

---

### 3.5.3. EXTERNAL AND INTERNAL SCROLLING

The terminal supports two types of scrolling operations:

When the data entered passes the bottom or last row of the display window (i.e.; the 24th row of the 'Normal' display window or the 'M'th row of the upper display window under the horizontal split format), the display window scrolls up.

If the bottom row of the display window corresponds to the end-of-text row, the scrolling operation causes the loss of the start-of-text row. This kind of scrolling action is defined as external scrolling because it is usually activated during communication with the host computer.

External scrolling is prohibited when the terminal is in the PROTECT, EDIT, or data validation submode. (Note: these are normally local operating modes). If the data entered passes the end-of-text row, the cursor is positioned to the start-of-text row.

If the bottom row of the display window does not correspond to the end-of-text row, the display window scrolls up one row to review the immediately following data line. There is no data lost under this kind of scrolling operation and hence is defined as internal scrolling.

Internal scrolling applies only if the size of the display window is smaller than the size of its corresponding text segment and is normally used for local editing and text viewing.

### 3.5.4. INTERNAL PAGING

The terminal provides the user with an internal paging mechanism which forces the display window to scroll up or down the text segment a number of rows equal to that of the display window. For example, if the display window supports 12 rows of text, the display window scrolls through 12 rows of text with one internal paging operation.

## 3.6 AUTOMATIC DATA VALIDATION

The automatic data validation feature is provided by the terminal to perform local error checking of operator inputs under block mode. Using this feature can substantially reduce host computer overhead. To use the data validation feature, the second page option is required.

The data validation feature is initiated by the ESC< sequence. Once initiated, the terminal internal memory is split into two equal text segments of 24 rows of 80 characters each as if the following horizontal split mode command had been issued:

**sequence: ESC x 5 8** (see the ACTION OF CHANGING DISPLAY MODE for ESC x, Section 5.2.12)

The two split segments are used as a foreground 'form' and background storage for the validation parameters.

Split segment 0 is used to build the data entry 'form' with protected and unprotected fields and split segment 1 is used to set up the validation parameters. Each character position in segment 0 corresponds to a character position in segment 1 in the same row/column position, i.e. the character in row 2 column 18 in segment 1 contains the data entry validation parameters for the character position in row 2 column 18 of segment 0 (Figure 3-6). The validation parameters set up in segment 1 can be entered by activating segment 1 and then entering as any normal data. The parameters can be:

- space - any key code is acceptable, or
- A - alphabet only, or
- B - numeric only, or
- D - numeric plus special characters including: + . \* / ( ) ,

*It should be noted that data validation can be applied only to the data entry area or the unprotected fields and that the data validation check applies only to keyboard entries.*

External scrolling is disabled when this mode is activated. The split screen selection key (the shifted PAGE key) is ignored to protect the validation parameters stored in segment 1. The following steps should be used to activate the data validation mode:

1. Turn on data validation mode with ESC< (The host computer should wait for acknowledgement (ACK) from the terminal for the completion of the set-up operation of the data validation mode before going to step 2).
2. Place the terminal into block mode.
3. Lock keyboard with ESC # and turn protect mode off with ESC '.

4. Activate segment 1 with ESC } and set up the validation parameters. The data/text entry area can be 'hidden' with the ESC A 0 3 sequence prior to setting up the parameters so that the operator will not see the validation parameters as they are entered onto the screen.
5. Activate segment 0 with ESC ], turn on the data/text entry area with ESC A 0 0 and then set up the protected 'form' on segment 0.
6. Turn on protect mode with ESC & and unlock the keyboard with ESC " for data entry.

*Note: For subsequent setups, only steps 3 through 6 are necessary.*

### 3.7 AUTOMATIC WORD-WRAPPING

The word-wrapping feature is activated under the EDIT mode and only affects keyboard entries. When text is entered at the end of a row (i.e. the 80th, 39th, or 40th column, depending on the selected display mode), the word-wrapping operation is initiated. A 'line-break' position is determined by backtracking from the end of the row. The 'line-break' position is defined as the character immediately after any non-alphabet character or protected character. Characters starting from the 'line-break' position to the end of the row are moved to the beginning of the next row. The cursor is placed to the right of the last character position of the wrapped characters. Any character positions at the beginning of the next row will be overwritten. If the insert mode is on the wrap characters will be inserted instead. The word-wrapping feature should be used only during block mode.

### 3.8 SPECIAL FORM CONTROL

A command sequence which allows all unprotect positions to be written with a specified code:

#### ESC . Code

For example, if code = A, then all the unprotected positions are written with 'A's.

This command sequence is very useful in data entry application. One example is an application which requires all unprotected positions to be written with underline symbols so that the operator can visualize the exact size of a data field. Upon completion of sending all the entered data to the host computer, the host computer can send an 'ESC . \_' sequence to reset all unprotected positions with underline symbols. This enhances the speed in transaction oriented systems.

Another command sequence allows all unprotected positions to be written with a specified attribute code:

#### ESC ! ATTR

where ATTR is any attribute code which is specified for the ESC G attribute generation sequence (see Table 6). This command sequence is provided for the generation of forms which require attribute enhancements. The following are two examples where this sequence applies:

1. The entire screen can be written with 'normal' attribute codes before a form is generated. By doing this, it eliminates the 'flashing' effect which occurs in most terminals when a form is being generated.
2. If a form is composed of primarily one kind of attribute for its fields, this sequence can reduce the time and code required by presetting the screen with the required attribute.

### 3.9 SCREEN ENHANCEMENT ATTRIBUTES

The terminal provides five screen enhancement attributes: dim (or half intensity), reverse, underscore, blink, and blank (no show). These attributes can be used selectively in any combination to highlight data fields and text blocks. The code that generates these screen enhancement attributes occupies one screen position and is displayed as a 'blank'. It enhances the displaying of all character positions to the right and subsequent rows until the next screen attribute is encountered (see Section 5.2.10).

The terminal also supports a 'dim' or half intensity enhancement attribute which does not occupy any screen position. Characters generated under the write protect mode are displayed in half intensity and the enhancement affects the display on a character by character basis. This dim attribute can be turned off if desired (see Figure 2-2).

#### 3.9.1 SPECIAL SCREEN DISPLAY ENHANCEMENT

In addition to enhancing data or text with screen enhancement attributes, the terminal provides special screen enhancement attributes for different screen appearances (see Section 3.2 and Figure 3-1). These 4 special screen attributes can be set up by the ESC A command sequence and they are (see Section 5.2.10):

1. The data/text entry area attribute which is at the last character position of the top message field line.  
*Example: ESC A 0 / Creates a 'green' and underscore screen.*

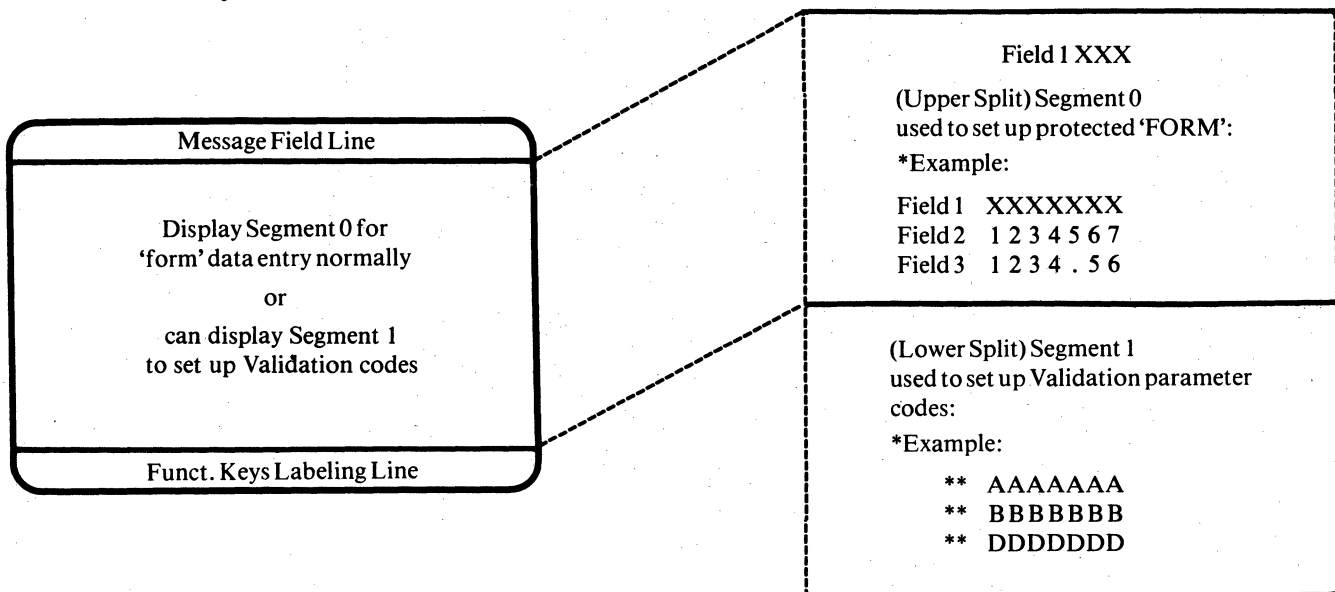
## GENERAL DESCRIPTION

---

Display Window:

Format: Horizontal Split at row 25

Storage Memory: Two Text Segments  
Each segment = 24 rows X 80 ch.  
Total Storage = 3840 characters



\*The field data validation feature requires the terminal internal storage to be arranged under the horizontal split display format so that the 3840 characters storage memory are split into two equal segments of 24 rows X 80 characters each. The protected form is set up in segment 0 and the validation codes are set up in segment 1 in a one character to one validation code basis at the same row/column coordinate. This example shows that field 1 is an alphabet field and field 2 is a numeric field only.

\*\*Validation code is as follows:

- space = any key in accepted
- A = alphabet only
- B = numeric only
- D = numeric, . + \*/(or)

**FIGURE 3-6**  
**TERMINAL FIELD DATA VALIDATION ARRANGEMENT**

2. The function key labeling line attribute which is at the start and end of each function key labeling field.  
*Example: ESC A 1 t Creates reverse and dim labeling fields.*
3. The local message field attribute which is at the start of the message field line.  
*Example: ESC A 2 1 Blanks the local message field.*
4. The host message field attribute which is at the start of the host message field.  
*Example: ESC A 3 2 Blinks the host message field.*

Upon power up, these attributes are set up as follows:

1. The data/text entry attribute is set to normal or reverse depending on the green/black screen selection (see Appendix A).
2. The function key labeling field attribute is set to 'dim' or half intensity.
3. The local message field attribute is set to underscore.
4. The host message field attribute is set to a space character.

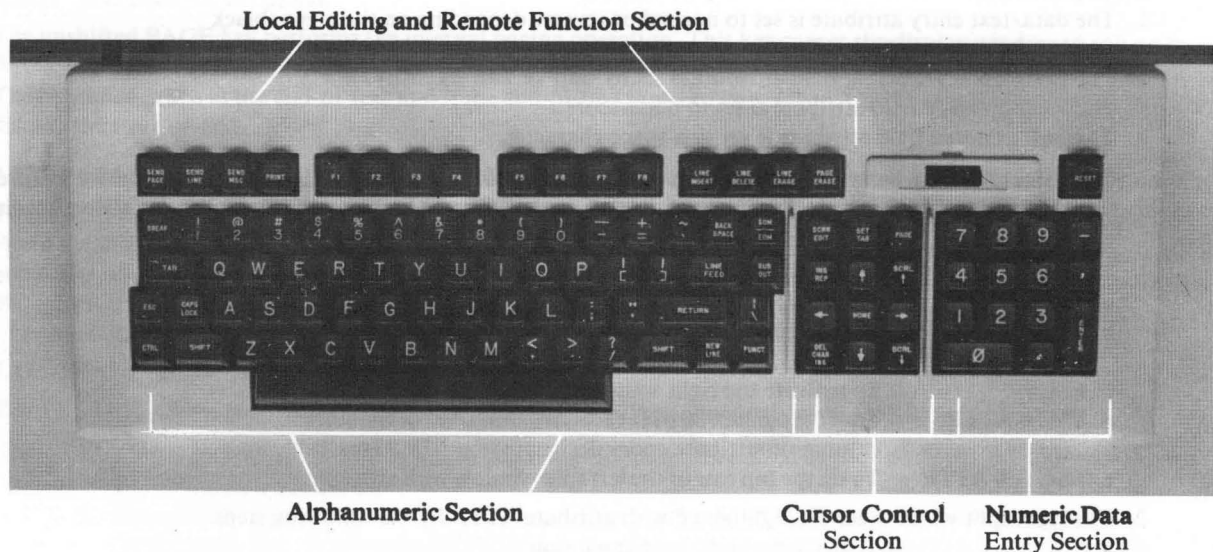
A column of attribute codes can be written into column 40 of the active display window. This screen enhancement feature allows the two split windows of the vertical split screen display mode (39) to be highlighted. The following shows how it can be implemented:

1. The left split window can be highlighted with the attribute ATTR by the following step:
  - a. **ESC ]** To activate the left split window.
  - b. **ESC n 0** To enter at column 40 of the screen with a column of 'normal' display attributes.
  - c. **ESC }** To activate the right split window.
  - d. **ESC n ATTR** To set at column 80 of the screen (column 40 of the active split window) with a column of attribute codes designated by ATTR (see Table 6).
  - e. **ESC A 0 ATTR** To set the top row of the left split window with attribute enhancement ATTR.
2. The right split window can be highlighted with attribute ATTR by the following steps:
  - a. **ESC }** To activate the right split window.
  - b. **ESC n 0** To enter at column 80 of the screen with a column of 'normal' display attributes.
  - c. **ESC ]** To activate the left split window.
  - d. **ESC n ATTR** To enter at column 40 of the screen with a column of attribute codes designated by ATTR (see Table 6).

## KEYBOARD

### SECTION 4. KEYBOARD

The keyboard is styled after that of a conventional typewriter. Most of the key functions are self-explanatory. The programmable function keys can be tailored by the host computer for specific applications and can be used in conjunction with the function key labeling line. Alphanumeric keys, control keys, and cursor keys are repeatable if the key is depressed for more than a half second. The terminal keyboard layout is presented in Figure 4-1.



**FIGURE 4.1**  
**KEYBOARD**

#### 4.1 SHIFT, CAPS LOCK, AND CTRL KEYS

The SHIFT key is identical to that of a conventional typewriter shift key except that the shift action also applies to other local editing and function keys. Depressing the shift key alone will not cause anything to happen.

The CAPS LOCK key only controls the alpha keys. Depressing the CAPS LOCK key turns the CAPS mode on and off in a toggling action. When the CAPS mode is on, 'CAPS' is displayed on the local message field and all alpha keys are entered in upper case.

The CTRL key is used in conjunction with certain other keys to generate ASCII control codes (see Table 1).

#### 4.2 ESC KEY and ESCAPE SEQUENCES

The ESC key is used to generate the ASCII ESC (escape) code which is normally used in conjunction with one or more other ASCII key codes to formulate special escape sequences that control certain terminal functions.

Escape sequences generated from the keyboard are executed if the terminal is under block or half duplex conversation mode. These functions are discussed in detail in the communication control section (see Section 5).

#### 4.3 ALPHANUMERIC KEYS

The ALPHANUMERIC keys are used to enter any of the standard ASCII characters (Table 2) into the terminal. All alphanumeric keys can operate in conjunction with the SHIFT key. However, only the alpha keys are affected by the CAPS LOCK key. Certain alphanumeric keys generate ASCII control codes when the CTRL key is pressed simultaneously with them (see Table 1).

##### 4.3.1 RUB OUT

The RUB OUT key generates a rub out (DEL or X'7F'). The rub out code is normally not displayed except under the monitor mode.

#### 4.3.2. NUMERIC KEY PAD and ENTER KEY

The numeric key pad is used mainly for convenience in data entry. The '-', '.', and ',' keys generate ASCII equivalent codes. The ENTER key is switch selectable to generate either a RETURN key or a TAB key. With the shift key depressed, the ENTER key turns the key-click on/off.

#### 4.4. CURSOR CONTROL KEYS

The cursor indicates the next position available on the screen for data entry. It can be switch selectable to be either a blinking or non-blinking rectangular block. (See Appendix A). As characters are entered the cursor moves from left to right across the display. The cursor automatically moves to the beginning of the row immediately below when it reaches the end of a row. Scrolling may occur when the cursor reaches the bottom row of the display window. Cursor movement can be controlled either by the operator or the computer.

**The cursor skips all protected data on the screen if the protect mode is on:**

- If cursor movement is towards the right of the screen, the cursor skips all protected positions until it reaches the first unprotected position to the right. In some cases, the cursor may have to skip several rows of protected data before 'landing' on an unprotected position.
- If cursor movement is towards the left of the screen, the cursor skips all protected positions until it reaches the first unprotected position to the left. In some cases, the cursor may have to skip several rows of protected data before 'landing' on an unprotected position.
- If cursor movement is towards the bottom of the screen and a protected position is immediately below the original cursor position, the cursor moves to the right of the protected position in its search for the next unprotected position.
- If cursor movement is towards the top of the screen and a protected position is immediately above the original cursor position, the cursor moves to the left of the protected position in its search for the next unprotected position.

Command code sequences generated by cursor control keys are outlined in Table 10.

#### 4.4.1 HOME

*When unshifted*, this key moves the cursor to the first character position at the upper left hand corner of the display window.

*When shifted*, this key moves the cursor to the first character position of the text segment or the start-of-text. The shifted HOME key forces the display window to display the start-of-text. In either case, the cursor moves to the first unprotected position at the upper left corner of the display window.

#### 4.4.2 CURSOR RIGHT

This key moves the cursor one character position to the right. When the cursor reaches the lower right hand corner of the display window scrolling occurs. When the cursor reaches the end-of-text and the terminal is not in the EDIT, protect or data validation modes, external scrolling occurs. That is, data is scrolled off the screen and internal memory; therefore, it is subsequently lost.

If the protect mode is on, the cursor moves right to the first unprotected position. When the cursor reaches the end-of-text and when the terminal is in the EDIT, protect, or data validation modes, the cursor moves to the start-of-text of the current text segment. If the start-of-text is not currently displayed, this forces the terminal to display the start-of-text.

#### 4.4.3 CURSOR LEFT

This key moves the cursor one character position to the left. When the cursor reaches the upper left hand corner of the display window scrolling occurs.

If the protect mode is on, the cursor moves left to the first unprotected position. When the cursor reaches the start-of-text, it moves to the end-of-text and forces the terminal to display the end-of-text.

#### 4.4.4 CURSOR UP

This key moves the cursor to the screen position immediately above the present cursor location. When the cursor reaches the top of the display window, scrolling may occur.

If protect mode is on and the position immediately above is protected, the cursor moves left starting from the protected position above to the first unprotected position that it finds. When the cursor reaches the start-of-text, it moves to the end-of-text and forces the end-of-text to be displayed.



## KEYBOARD

---

### 4.4.5 CURSOR DOWN

This key moves the cursor to the screen position immediately below the current cursor location. If the cursor passes the bottom of the display window, scrolling may occur.

If protect mode is on and the position immediately below is protected, the cursor moves right starting from the protected position immediately below to the first unprotected position it finds.

If the cursor passes the end of text row and the terminal is in the EDIT, protect, or data validation modes, it moves to the start-of-text and forces the start-of-text to be displayed.

### 4.4.6 BACKSPACE

The operation of this key is similar to that of the CURSOR LEFT key.

### 4.4.7 RETURN

This key performs the carriage return (CR) operation which moves the cursor to the first position on the same row. If protect mode is on, the cursor moves to the first unprotected position on the same row.

This key is switch selectable to perform the combination of the carriage return (CR) and the LINE FEED (LF) keys. (See Appendix A).

### 4.4.8 LINE FEED

The operation of this key is similar to that of the CURSOR DOWN key.

### 4.4.9 NEW LINE

This key moves the cursor to the first character position of the row immediately below. This operation is a combination of that of the LINE FEED key and the carriage return operation.

### 4.4.10 TAB (unshifted)

This key moves the cursor right to the next column position where the tab has been set. If no column tab position can be found, the cursor stops at the first position of the row immediately below. If the cursor passes the last character of the display window, scrolling may occur. This tab operation is disabled if the column-tab mode is off.

If protect mode is on, the cursor moves right to the next unprotected field. If the cursor passes the end-of-text, it moves to the start-of-text and forces the start of text to be displayed.

### 4.4.11 TAB (shifted)

This key performs the BACK TAB operation. The cursor moves left to the first column tab position it finds on the current row. If there is no column tab position to the left, the cursor stops at the first position of the current row. If the cursor is at the first position of the row when the TAB (shifted) key is depressed, the cursor moves to the right-most column tab position of the row immediately above. This back tab operation is disabled if the column-tab mode is off.

If protect mode is on, the cursor moves left to the next unprotected field. If the cursor passes the start of text, it moves to the end of text and forces the end of text to be displayed.

### 4.4.12 SET TAB (unshifted)

This key causes the current cursor column position to be stored as a column-tab position. The operation is similar to that of setting the tab on a typewriter. Setting the tab turns on the column-tab mode.

This typewriter-styled tab setting operation affects the cursor movement only when the protect mode is off. Column tabbing under protect mode can be accomplished by setting a column of protected spaces at the cursor column position using the ESC V sequence (see Section 5.2.7).

### 4.4.13 SET TAB (shifted)

This key causes the column-tab position previously stored at the current cursor column position to be cleared. Again, the operation is similar to that of a typewriter tab clear. Clearing the column-tab at the first column turns the column-tab mode off.

## 4.5 TEXT/DATA EDITING KEYS

The terminal provides a comprehensive set of editing features for form/data entry and text editing for the terminal user and the host computer. Most of these features have been associated with an editing key on the keyboard. These editing keys have been grouped functionally according to their frequency of use.

---

The user programmable switches allow the operator to determine whether these editing keys are totally/partially controlled by the host computer, or whether they are totally local facilities of the terminal (see Appendix A for switch setting).

If selected as host controllable (or duplex keys), their associated command sequences are sent to the host computer when the terminal operates under conversation mode. Command code sequences generated by these editing keys are outlined in Table 10.

#### **4.5.1 DEL/INS CHAR (unshifted)**

This key moves all the characters right one position starting from the cursor position to the end of the row. A space character is placed at the cursor position. The last character on the row is lost.

If protect mode is on, this operation halts at the end of an unprotected field or the end of row, whichever is encountered first.

#### **4.5.2 DEL/INS CHAR (shifted)**

This key deletes the character at the cursor position. All characters, starting from the cursor location to the end of the row are moved one character position to the left. A space is placed at the last character position of the row.

If protect mode is on, this operation halts at the end of an unprotected field or the end of row, whichever is encountered first.

#### **4.5.3 LINE INSERT**

This key causes all rows, starting from the cursor row to the end-of-text row of the text segment to be moved down one row. A row of spaces is inserted at the original cursor row and the cursor is placed at the start of that row.

This operation is prohibited if the protect mode is on.

#### **4.5.4 LINE DELETE**

This key causes the cursor row to be deleted. All rows, starting from the row immediately below the cursor row to the end-of-text row of the text segment is moved up one row.

This operation is prohibited if the protect mode is on.

#### **4.5.5 LINE ERASE (unshifted)**

This key causes all characters, starting at the cursor to the end of the row to be cleared to spaces (X'20').\*

If the protect mode is on, this operation terminates at the end of an unprotected field or end of row, whichever is encountered first.

#### **4.5.6 LINE ERASE (shifted)**

This key is the same as the above except it clears all characters to nulls (X'00').

#### **4.5.7 PAGE ERASE (unshifted)**

This key causes all characters, starting at the cursor position to the end of text to be cleared to spaces (X'20').

If protect mode is on, only the unprotected characters are cleared.

#### **4.5.8 PAGE ERASE (shifted)**

This key performs the same function as PAGE ERASE (unshifted) except that it clears all characters to nulls (X'00').

#### **4.5.9 INS/REP (unshifted)**

This key places the terminal into the replace mode under which any character entered replaces the character at the cursor position.

#### **4.5.10 INS/REP (shifted)**

This key places the terminal into the insert mode under which any character entered is inserted at the cursor position. All characters on the same row, starting from the cursor to the end of row (or end of unprotected field under protect mode), are moved to the right by one position. The last character on the row (or unprotected field) is lost.

\*These are ASCII codes, see Table 2.

## KEYBOARD

---

### 4.5.11 SCRN EDIT (unshifted)

This key places the terminal into the EDIT mode:

1. The automatic word-wrapping operation is activated. If character entry from the keyboard passes the end of a row, the partially entered word is wrapped to the next row with the cursor position correctly adjusted so that efficient text entry is possible. The user does not have to worry about text placement to ensure that a word is not broken up at the end of a row (see Section 3.7).
2. When the cursor passes the end-of-text, it moves to the start-of-text and vice versa. This 'modified' cursor movement method avoids accidental data loss at the start-of-text during local editing.

The EDIT submode is for local editing purposes when the terminal is in block mode.

### 4.5.12 SCRN EDIT (shifted)

This key turns off the EDIT mode and:

1. deactivates automatic word-wrapping at the end of a row, and
2. if neither the protect mode nor the data validation mode is on, external scrolling is allowed when the cursor passes the end of text.

## 4.6 BLOCK TRANSMISSION KEYS

The terminal transmits data to the host computer using standard ASCII codes (see Table 2) using the following rules:

1. Null characters (X'00') shown on the display are not transmitted.
2. The line-ending code sequence and the end-of-block sequence transmitted to the remote computer are switch selectable (see Appendix A) to be one of the following sequences:
  - a. 'End-of-line' = X'1F' (the New Line Operation Code)  
'End-of-transmission' = X'0D' (the Carrage Return Code)
  - b. 'End-of-line' = X'0D', X'0A', X'00' (CR/LF sequence)  
'End-of-transmission' = X'03' (the End-Of-Text or ETX)
3. Screen enhancement attributes are sent as spaces.
4. If the protect mode is on, graphic codes are sent as spaces; otherwise, they are sent as control codes (see Table 4.)

### 4.6.1 SEND LINE (unshifted)

If protect mode is off, this key causes all characters, starting from the start-of-row up to and including the cursor position, to be sent to the host computer. The end-of-transmission sequence is sent as the terminator.

If protect mode is on, protected fields are replaced by a field separator code, X'1C'.

### 4.6.2 SEND LINE (shifted)

If protect mode is off, this key behaves the same as the SEND LINE (unshifted) key.

If protect mode is on, the protected fields are sent and bracketed by 'write protect on' code, ESC ) and 'write protect off' code, ESC ( .

### 4.6.3 SEND PAGE (unshifted)

If protect mode is off, this key causes all characters, starting from the start-of-text up to and including the cursor position, to be sent to the host computer. The 'end of line' sequence is sent at the end of each row, and the 'end of transmission' sequence is sent as the transmission terminator.

If protect mode is on, each protected field is replaced by a field separator code, X'1C'.

### 4.6.4 SEND PAGE (shifted)

If protect mode is off, this key behaves the same as the SEND PAGE (unshifted) key.

If protect mode is on, protected fields are sent and bracketed by 'write protect on' code, ESC ) and 'write protect off' code, ESC ( .

### 4.6.5 SOM/EOM (unshifted)

This key enters an 'End Of Message' (X'03') mark at the cursor position.

#### 4.6.6 SOM/EOM (shifted)

This key enters a 'Start of Message' (X'01') mark at the cursor position.

#### 4.6.7 SEND MSG (unshifted, read as send message)

If protect mode is off, this key causes all characters, starting from the first SOM mark found to the left of the cursor position to the first EOM mark encountered (starting from the SOM mark), to be sent to the host computer. The 'end-of-line' sequence is sent at the end of each row, and the 'end-of-transmission' sequence is sent as the transmission terminator.

If protect mode is on, each protected field is replaced by a field separator code, X'1C'. The SOM (X'01') and EOM (X'03) marks are not sent.

#### 4.6.8 SEND MSG (shifted, read as send message)

If protect mode is off, this key behaves the same as the SEND MSG (unshifted) key.

If protect mode is on, each protected field is sent and bracketed 'write protect on' code, ESC ( . The SOM (X'01') and EOM (X'03') marks are not sent.

#### 4.7 BREAK KEY

This key causes a break signal of roughly 250 milliseconds to be sent to the host computer. The BREAK key is disabled when block transmission or local print is in progress.

#### 4.8 PROGRAMMABLE FUNCTION KEYS

There are a total of 8 function keys on the terminal which can send up to 16 function key sequences to the host computer when used in either the unshifted or shifted positions. These 16 function key sequences are programmable up to 8 characters/codes each or 16 characters/codes each if the terminal operates with the second page option (see Section 5.2.17).

Upon power-up, a default function key sequence is assigned with each of the 16 function key positions (see Table 3). The host computer can independently and selectively program these 16 function keys with different sequences. Once programmed, depressing the appropriate function key causes the associated function key sequence to be sent to the host computer. The function key positions that are not programmed send out the default function key sequence.

A programmed function key position can be 'deprogrammed' by using the function key programming sequence with a "null" sequence. Once deprogrammed, the default function key sequence applies again.

##### 4.8.1 FUNCT

This key can be used in combination with any other alphanumeric key to generate additional function code sequences. Depressing the FUNCT key followed by any alphanumeric key X generates a function key sequence of SOH X CR to the host computer (see Section 5.2.18).

#### 4.9 LOCAL PRINT KEYS

For all local print operations, screen enhancement attributes are printed as spaces. Graphic characters are printed as spaces under protect mode. If protect mode is off, they are sent to the local printer as normal ASCII control codes (see graphic ASCII codes in Table 4).

If there is no local printer attached or the printer is off when any print operation is initiated, the print operation is aborted so that the terminal will not hang up.

##### 4.9.1 PRINT (unshifted)

This key causes all characters, starting from the start-of-text up to and including the cursor position, to be transmitted to the printer port. Each row sent is terminated by the following sequence:

*Carriage Return (CR), Line Feed (LF), and Null (X'00')*

If protect mode is on, only unprotected data is transmitted so that pre-printed forms can be used.

##### 4.9.2 PRINT (shifted)

This key causes both protected and unprotected characters to be printed regardless of the mode setting.

##### 4.10 SCRL↑ (scroll up key)

This key causes the display window to scroll towards the start of the text by one row. Scrolling is defeated if the start-of-text is displayed.

## KEYBOARD

---

### 4.11 SCRL↓ (scroll down key)

This key causes the display window to scroll towards the end of text by one row. Scrolling is defeated if the end-of-text is displayed.

### 4.12 PAGE key

The PAGE key performs two entirely different functions in the unshifted and shifted positions; 1) unshifted: internal paging, 2) shifted: split display window selection.

#### 4.12.1 INTERNAL PAGING

The unshifted PAGE key performs the internal paging operation. This key causes the display window to roll up by one display window. For example, if the display window displays 12 rows of text, it scrolls 12 rows of text in one operation. The operation stops at the end-of-text and if the end-of-text is already displayed, this key forces the display window to display the start-of-text.

#### 4.12.2 SPLIT DISPLAY WINDOW SELECTION

The shifted PAGE key activates the alternate text segment if the terminal operates under the split screen display mode. When the shifted PAGE key is depressed, the cursor moves up to the display window of the alternate text segment. Subsequent text editing and communication will be performed on this text segment. The previous protect mode status (on or off) which this text segment is operated under will be restored.

### 4.13 RESET (unshifted)

The RESET key is provided to help the terminal operator in the following ways:

1. Assisting the operator to recover from a hang-up condition caused by the failure of the transmission line or the host computer.
2. Allowing the operator to re-configure four of the user programmable switches during operation of the terminal. (i.e., to select the RETURN key to perform the combination of the carriage return (CR) and the LINE FEED (LF) keys).
3. Terminating a block transmission to the host computer or local printer before completion of the operation.

In general, the RESET key causes the terminal to perform the following three functions:

1. Unlocks the keyboard.
2. Resets (re-initializes) the communication electronics.
3. Reads the following user programmable switches (see Appendix A):
  - a. *selection of edit keys that are transmitted to the host computer.*
  - b. *auto new line (NL) option.*
  - c. *ENTER and RETURN key set up.*
  - d. *block transmission line ending and termination sequences selection.*

If there is a block transmission to the host computer or printer in progress, depressing the RESET key only terminates the transmission and does not cause the terminal to perform the above three functions.

Under normal operation, the RESET key should be used cautiously because it will cause the loss of data if there is communication from the host computer while the reset operation is in progress.

#### 4.13.1 RESET (shifted)

This key generates a 'set block mode' sequence (ESC B) which places the terminal under the block mode so that local operation can be activated. Under the block mode, the terminal can be selected to operate in any other mode via the appropriate sequence.

### 4.14 KEY-CLICK CONTROL

The terminal keyboard audio feed-back or key-click is controlled by the shifted ENTER key. Depressing the shifted ENTER key enables or disables the key-click in a toggling action. Enabling or disabling the key-click is only a local operation and will not affect communication with the host computer. The keyboard key-click is turned on upon power-up of the terminal.

## SECTION 5. COMMUNICATION CONTROL

The remote host computer can control the terminal operation via a series of command sequences. These command sequences fall into two categories: 1) Commands using single byte ASCII control codes, 2) Commands using ESCape sequences.

Most of these command sequences correspond to keys on the keyboard. These keys and their corresponding command sequences are listed in Table 10. Their effects on communication are governed by the following rules:

1. If the terminal is operating under block mode, all command sequences generated from the keyboard are only executed locally.
2. If the terminal is operating under half-duplex conversation mode, all command sequences generated from the keyboard are locally executed and subsequently sent to the host computer. Keys whose functions correspond to escape sequences (e.g. DEL CHAR) can be programmed so that their escape sequences are not sent to the host computer (see Appendix A for switch setting).
3. If the terminal is operating under full-duplex conversation mode, all command sequences are sent to the host computer only. Keys whose functions correspond to escape sequences (e.g. DEL CHAR) can be programmed so that they are only executed locally (see Appendix A for switch setting and Table 10 for keyboard command code sequences).

### 5.1 ASCII CONTROL CODES

The control code commands are listed in Table 1. Control codes can be generated from the keyboard by using the CTRL key in conjunction with certain alphanumeric keys. They can also be generated by some special keys on the keyboard such as the HOME key (see Table 10).

Control codes generated from the keyboard are executed if the terminal is under block or half-duplex conversation mode. If the terminal is under full-duplex conversation mode, they are only sent to the host computer and are not acted upon locally.

### 5.2 ESCAPE SEQUENCES

Escape sequences are command sequences to instruct the terminal to perform certain operations. They are all preceded by an ESC code. Escape sequences can be generated from the keyboard by the following two means:

1. By depressing the ESC key followed by one or more keys to generate the required sequence.
2. By using certain editing keys such as DEL CHAR to generate the specific escape sequence associated with each of these keys (see Table 10).

Escape sequences generated from the keyboard are only locally executed or only transmitted to the host computer, or both, depending on the current operating mode of the terminal. Escape sequences received from the host computer are always executed (see Section 5).

The following sections list all the terminal escape sequences according to their usages. Spaces between codes within each sequence are used for clarity only. These spaces are not part of the escape sequences. For quick reference to all these escape sequences, see Table 9.

#### 5.2.1 CURSOR SET/READ

The host computer can control the terminal cursor by the following sequences:

**ESC =** Sets cursor in the active window. This is a multiple code sequence.

*ESC = r c* where *r* = cursor row (see Table 5)  
*c* = cursor column (see Table 5)

**ESC ?** Reads cursor of the active window. The terminal returns to the host computer with:

*r c CR* where *r* = cursor row (see Table 5)  
*c* = cursor column (see Table 5)  
*CR* = 'X'0D'

**ESC -** Sets cursor to the specific split window. This is a multiple code sequence:

*ESC - n r c* where *n* = 0 or 1 = split segment number  
*r* = cursor row (see Table 5)  
*c* = cursor column (see Table 5)

## COMMUNICATION CONTROL

---

**ESC /** Read the active split segment number and the cursor position. The terminal replies with:  
*n r c CR* where *n = 0 or 1 = split segment number*  
*r = cursor row (see Table 5)*  
*c = cursor column (see Table 5)*

In addition, the cursor can be moved from the active display window to the other split display window by using the split segment selection escape sequence (see Section 5.2.13).

### 5.2.2 KEYBOARD LOCK/UNLOCK

The terminal keyboard can be locked and unlocked by the following sequences:

**ESC "** Unlock (enable) keyboard.  
**ESC #** Lock (disable) keyboard, all key entries are ignored except:  
1. the *RESET* key,  
2. the *BREAK* key, and  
3. the *FUNCTION* keys

### 5.2.3 COMMUNICATION MODES CONTROLS

**ESC B** Places the terminal into the block mode.  
**ESC C** Places the terminal into the conversation mode.  
**ESC D** Sets the terminal into either half or full duplex communication. This is a multiple code sequence:  
*ESC D x* where *x = H if for half duplex*  
*= F if for full duplex*  
**ESC U** 'On' monitor mode. All entries to the terminal are displayed, but none are interpreted or executed except for the 'off monitor mode' sequence.  
**ESC u** 'Off' monitor mode.  
**ESC X** 'Off' monitor mode.

### 5.2.4 TEXT EDITING MODES CONTROLS

**ESC N** 'On' EDIT mode (see Section 4.5.11).  
**ESC O** 'Off' EDIT mode (see Section 4.5.12).  
**ESC q** 'On' insert mode (see Section 4.5.10).  
**ESC r** 'Off' insert mode (see Section 4.5.9).

### 5.2.5 PROTECT FORM CONTROLS

**ESC &** 'On' protect mode. If the terminal is under protect mode, all cursor movement will skip protected data (see Section 4.4). In addition, external scrolling is prohibited.  
**ESC '** 'Off' protect mode.  
**ESC )** 'On' 'write protect' mode. All characters entered will be displayed in half intensity and will be protected under the protect mode. The 'write protect' mode can be activated when the terminal is under any other mode.  
**ESC (** 'Off' 'write protect' mode.  
**ESC <** 'On' data validation mode (see Section 3.6).  
**ESC >** 'Off' data validation mode (see Section 3.6).  
**ESC . CODE** All unprotected positions are written with the specified code, CODE (see Section 3.8).  
**ESC ! ATTR** All unprotected positions are written with the specified attribute code, ATTR (see Section 3.8 and Table 6).

### 5.2.6 SPECIAL CURSOR CONTROLS

**ESC i** Performs the TAB operation.  
**ESC I** Performs the BACK TAB operation (see Section 4.4.11).  
**ESC }** Moves the cursor to the start of active text segment.

**5.2.7 COLUMN TABULATION CONTROLS**

There are two kinds of column tabulation control: 1) The 'typewriter styled' column-tab setting which operates when protect mode is off, 2) The protect-space-column setting which facilitates the generation of data entry 'forms' with column fields.

- ESC 0** Turns 'Off' the column-tab mode and clears all column tab settings.
- ESC 1** Turns 'On' the column-tab mode and stores the current cursor column position as a tab position (see Section 4.4.11 and 4.4.12).
- ESC 2** Clears the column-tab position at the current cursor position.
- ESC V** Sets a column of protected spaces at the cursor column, starting from the row where the cursor is down to the end-of-text row.

**5.2.8 DATA/TEXT EDITING CONTROLS**

- ESC \*** Clears all text with nulls (X'00'). The active text segment is cleared to nulls (X'00'), the protect mode is turned off and the cursor is placed at the HOME position.
- ESC +** Clears all text with spaces. The active segment is cleared to spaces (X'20'), the protect mode is turned off and the cursor is placed at the HOME position.
- ESC ,** Clears all text with protected spaces. The active text segment is cleared with protected spaces (X'A0'), the protect mode is turned off and the cursor is placed at the HOME position.
- ESC :** Clears all unprotected text with nulls (X'00').
- ESC ;** Clears all unprotected text with spaces (X'20').
- ESC E** Inserts a line, see Section 4.5.3.
- ESC R** Deletes a line, see Section 4.5.4.
- ESC Q** Inserts a character, see Section 4.5.1.
- ESC W** Deletes a character, see Section 4.5.2.
- ESC T** Erases to the end of the line with spaces, see Section 4.5.5.
- ESC t** Erases to the end of the line with nulls (X'00'), see Section 4.5.6.
- ESC Y** Erases to the end of the text segment with spaces, see Section 4.5.7.
- ESC y** Erases to the end of the text segment with nulls (X'00'), see Section 4.5.8.

**5.2.9 SCROLLING AND PAGING CONTROLS**

- ESC v** Performs internal scrolling up (towards the end-of-text), see Section 4.10.
- ESC w** Performs internal scrolling down (towards the start-of-text), see Section 4.11.
- ESC h** Performs internal paging of the active text segment, see Section 4.12.1.

**5.2.10 SCREEN ATTRIBUTES AND GRAPHICS**

- ESC G** Enters the attribute code at the cursor position. This is a multiple code sequence (see Section 3.9):  
*ESC G ATTR* where *ATTR* = attribute code (see Table 6)
- ESC A** Sets special screen attribute. This is a multiple code sequence used to set the four special attributes on the screen (Figure 3-1 and Section 3.9.1).  
*ESC A n ATTR* where *n* = 0 for the data/text entry area  
= 1 for the funct. key labeling line  
= 2 for the local message field  
= 3 for the host message field  
*ATTR* = attribute code (see Table 6)
- ESC H** Enters the graphic code at the cursor position. This is a multiple code sequence:  
*ESC H GRAPH* where *GRAPH* = graphic code (see Table 4)

**5.2.11 MESSAGE FIELDS AND LABELING FIELDS**

- ESC F** Enters message to the host message field. This is a multiple code sequence (see Appendix B):  
*ESC F aaaa CR* where *aaaa* is a character string of up to 46 characters, and  
*CR* is the ASCII Carriage Return Code. Required if *aaaa* is less than 46 characters.



## COMMUNICATION CONTROL

---

**ESC z** Enters message to the selected field in the Function Key Labeling Line. This is a multiple code sequence (see Appendix C):

*ESC z n aaaa CR*      where *n* = the field number of the labeling field from 0 to 7 corresponding to each field from left to right, and  
*aaaa* is a character string of up to 8 characters, and  
*CR* is the Carriage Return Code. Required only if *aaaa* is less than 8 characters.

If *n* = ( the 8 function key labeling fields can be used contiguously as one message field of 78 characters. An example of using this message field is a ruler line which displays the margin settings and tab positions for word processing application.

Under either the ESC F or the ESC z sequence, embedded control codes in the message are displayed so that graphic characters can be used to enhance the message.

### 5.2.12 DISPLAY MODE SELECTION

**ESC x** Changes display mode. This is a multiple code sequence:  
*ESC, x, mode*      where *mode* = the display mode      (see Table 8)

If the mode selected is the horizontal split mode, the sequence is:  
*ESC, x, mode, HSR*      where *HSR* = the horizontal split row number      (see Table 8)

WHILE CHANGING THE DISPLAY MODE THE FOLLOWING THINGS WILL HAPPEN:

1. All the text segments are cleared except for the top Message Field Line, the Function Key Labeling Line, and the tab settings.
2. The communication electronics are reset, but the block/conversation mode and the HDX/FDX sub-modes are maintained.
3. The operating switches (SW1 - 1,2,3,4) are read.
4. The EDIT, protect, write-protect, insert, and monitor modes are all turned off.
5. The new display mode is set up with the text segment 0 activated.
6. Enables the CR code in the default function key sequence and resumes the cursor up key to generate a 'Control K' code.
7. An ACK is sent upon completion to the host computer if the change display mode command originates from the host.

### 5.2.13 SPLIT SEGMENT SELECTION

**ESC J** Activates the alternate split window or text segment.

**ESC K** Activates the alternate split window or text segment.

**ESC I** Activates split segment 0.

**ESC }** Activates split segment 1.

### 5.2.14 VERTICAL SPLIT COLUMN

**ESC n** Sets a column of characters as specified at column 40 of the active display window. This sequence is used to set the separation column for the vertical split mode of 39 columns with a character other than the default vertical line (see Sections 3.5 and 3.9.1). This is a multiple code sequence:  
*ESC n COLCODE*      where *COLCODE* is the column code (see Table 7).

### 5.2.15 BLOCK TRANSMISSION

**ESC 4** Send line unprotected. Sends all unprotected characters starting from the start-of-row up to and including the cursor character to the host computer (see the SEND LINE key operation in Section 4.6.1).

**ESC 5** Send page unprotected. Sends all unprotected characters starting from the start-of-text up to and including the cursor character to the host computer (see the SEND PAGE key operation in Section 4.6.3).

**ESC 6** Send line all. Sends all characters starting from the start-of-row up to and including the cursor character to the host computer (see the SEND LINE key operation in Section 4.6.2).

**ESC 7** Send page all. Sends all characters starting from the start-of-text up to and including the cursor character to the host computer (see the SEND PAGE key operation in Section 4.6.4).

**ESC 8** Enters a start of message mark (STX) at the cursor position.

**ESC 9** Enters an end of message mark (ETX) at the cursor position.

- ESC S** Send message unprotected (see the SEND MSG key operation in Section 4.6.7).  
**ESC s** Send message all (see the SEND MSG key operation in Section 4.6.8).

**5.2.16 LOCAL PRINT CONTROL**

- ESC @** All unprotected characters starting from the start-of-text up to and including the cursor character are sent to the local printer port. Each row is terminated with:  
*CR (carriage return), LF (line feed), and NULL (X'00')*  
 Protected characters and attributes are replaced by spaces. Graphic characters are replaced by spaces if the protect mode is on, otherwise they are sent as their corresponding control codes (see the PRINT key operation in Section 4.9.1).
- ESC P** Prints all (see the PRINT key operation in Section 4.9.2).  
**ESC p or ESC L** Prints unformatted. The sequence is the same as the 'print all' sequence (ESC P) except all characters are sent as they are. Attribute codes are replaced by spaces and no line-ending sequence is sent.

**5.2.17 PROGRAMMING THE FUNCTION KEYS**

- ESC z** The 8 function keys can be programmed independently and selectively with a total of 16 function key sequences for their shifted and unshifted positions with the following command sequence (see Section 4.8):  
*ESC z KCODE sequence DEL (or RUBOUT code).*

Where *KCODE* = key code of the key position to be programmed as follows:

UNSHIFTED	KCODE	SHIFTED	KCODE
F1	@	F1	H
F2	A	F2	I
F3	B	F3	J
F4	C	F4	K
F5	D	F5	L
F6	E	F6	M
F7	F	F7	N
F8	G	F8	O

*Sequence* = function key sequence up to 8 bytes, or 16 bytes if the terminal operates with the second page option. If *sequence* is a null sequence, the default function key code is used (see Table 3).

**5.2.18 TWO CHARACTERS FUNCTION KEY SEQUENCES AND MODIFIED CURSOR UP KEY**

The 'CR' code generated in the default function key sequences can be disabled and enabled by command sequences. If it is selected to be disabled, the cursor-up key is also modified to generate a 'control C' rather than a 'control K' code. These command sequences are as follows:

- ESC %** Disables the 'CR' code in the default function key sequences and modifies the cursor-up key to generate a 'control C' code (X'03').  
**ESC \$** Enables the 'CR' code in the default function key sequence and resumes the cursor up key to generate a 'control K' code (X'0B').  
 The 'CR' code and the 'Control K' codes are enabled upon power-up or completion of a display mode selection (see Section 5.2.12).

**5.2.19 READ CURSOR BYTE**

The host computer can read the character/byte where the cursor resides:

- ESC M** Causes the terminal to send to the host computer the character at the cursor position.

This is a very useful command sequence for word processing and data entry when selective data is desired to be sent to the host computer. This sequence, in conjunction with the function key, can create new functions such as send a sentence, send a field, or send a word to the computer.

### 5.3 REMOTE COMPUTER HAND SHAKING PROTOCOL

#### **XON/XOFF**

The terminal supports the XON and XOFF signals from the host computer. When the terminal receives a XOFF (DC3) from the host computer, it stops further transmission. When the terminal receives a XON (DC1) from the host computer, it resumes transmission.

#### **ENQ/ACK**

The terminal supports an internal receiver buffer to prevent the loss of data in communicating with the host computer. In some cases, the terminal may be performing time consuming tasks such as clearing the entire text segment. But instead of requiring the host computer to insert NULLs or timeouts (as a lot of conventional terminals do), the terminal provides a mechanism by which the host computer can inquire the busy/not busy status of the terminal.

The host computer can probe the terminal at any time to see if the terminal is busy or if its receiver buffer is empty by sending an ENQ to the terminal. If the terminal is not busy or if its receiver buffer is empty, the terminal returns an ACK to the host computer.

#### **ACK AFTER DISPLAY MODE INITIALIZATION**

While changing display modes using the ESC, x, MODE command, the terminal has to perform several time consuming operations (such as clearing the entire data storage). But instead of requiring the host computer to send an ENQ, the terminal transmits an ACK to the host computer upon completion of the display mode initialization.

#### **ACK AFTER LOCAL PRINT COMPLETION**

The terminal transmits an ACK to the host computer upon completion of the local print operation if it is initiated from the host computer.

**SECTION 6 TERMINAL SELF-TEST FACILITIES**

The terminal has two internal self-test features and display/keyboard verification capability to ensure proper terminal operation.

**POWER-UP SELF-TEST:**

Upon power up of the terminal, an internal self-test sequence is automatically initiated. The terminal halts all further operation if a faulty module is detected. An error message is displayed near the lower right hand corner of the screen as follows:

- |                           |                            |
|---------------------------|----------------------------|
| Z - faulty microprocessor | 9 - faulty memory chip A9  |
| 4 - faulty memory chip A4 | 0 - faulty memory chip A10 |
| 5 - faulty memory chip A5 | 1 - faulty memory chip A11 |
| 6 - faulty memory chip A6 | 2 - faulty memory chip A12 |
| 7 - faulty memory chip A7 | 3 - faulty memory chip A13 |
| 8 - faulty memory chip A8 |                            |

**DIAGNOSTIC SELF-TEST:**

The diagnostic self-test is a repetitive self-test sequence provided to assist the field engineer in isolating a faulty module in the terminal. In addition to performing all the test sequences in the power-up self-test, the diagnostic self-test also verifies the proper operation of the communication and printer port electronics. It requires two communication 'loop-back' connectors, one for the primary communication port and the other one for the local printer output port:

Primary port loop back pins connection: (RS-232C) 2 to 3, 4 to 5, 8 to 20; (Current loop) 7 to 13, 9 to 10, 11 to 12, 4 to 5, 8 to 20

Printer port loop back pins connection: 3 to 20

The diagnostic self-test can be initiated by setting all the switches in SW 1 to OFF prior to the power up of the terminal. If no error is detected, the terminal will repeat the self-test. If a fault is detected, the terminal terminates the diagnostic test and displays a message near the lower right hand corner of the screen:

- X - faulty communication port
- Y - faulty printer port

**DISPLAY/KEYBOARD VERIFICATION:**

The terminal provides the field engineer with the capability to verify proper keyboard connection and display adjustment. When the 'SPACE BAR' of the keyboard is depressed during the diagnostic self-test, the self-test is terminated and a pattern of all displayable characters and the screen attribute settings required for proper display adjustments appear on the screen.

Depressing any key at this point displays the associated key symbol at the 10th and 11th column position of the bottom row. An '!' symbol is displayed at column 9 if the depressed key is a function key or editing key. The following describes these key symbols:

1. All alphanumeric keys including special symbol keys such as \*, -, and keys on the numeric key pad are displayed with their associated symbols at columns 10 and 11 (see Table 2).
2. Keys generating ASCII control codes are displayed with their associated ASCII symbol (see Tables 1 and 10). An exception is the RETURN key which is displayed with the ! and the double vertical line graphic symbols at columns 9 and 10.
3. Keys generating escape sequences are displayed with an ! at column 9 and their associated escape sequence code at columns 10 and 11. For example, the SCRL up key is displayed with !ww at columns 9, 10, and 11.
4. The RESET and BREAK keys are not displayed. They can be verified by the key-click feed back.
5. The CAPS LOCK, SHIFT, AND CTRL are not displayed. They can be verified in conjunction with another alphanumeric key. For example, depressing the CTRL and A keys simultaneously generates the 'SH' symbols at columns 10 and 11.

## TERMINAL SELF-TEST FACILITIES

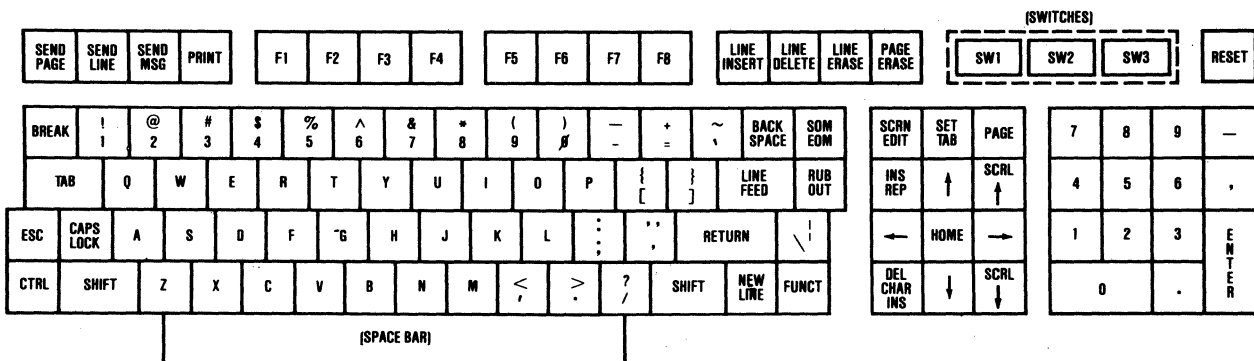
6. The function keys are displayed with the following symbols at columns 9 and 10:

	UNSHIFTED	SHIFTED		UNSHIFTED	SHIFTED
F1	!(blank)	!BS	F5	!EΓ	!FF
F2	!SH	!HT	F6	!EQ	!CR
F3	!SX	!LF	F7	!AK	!S0
F4	!EX	!VT	F8	!BL	!SI

7. The FUNCT key generates the ! and the control \_ symbols (see Table 1).

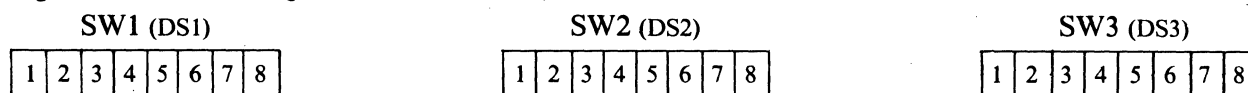
User programmable switch settings are displayed on the bottom row at columns 2 and 3 for SW1, 4 and 5 for SW2, and 6 and 7 for SW3. The following patterns can be used to verify the switch settings:

	SWITCH POSITION								DISPLAY POSITION ON BOTTOM ROW						
	1	2	3	4	5	6	7	8	2nd	3rd	4th	5th	6th	7th	
<b>PATTERN I:</b>															
SW1	on	off	on	off	on	off	on	off	#	:					
SW2	on	off	on	off	on	off	on	off			#	:			
SW3	on	off	on	off	on	off	on	off					%	Z	
<b>PATTERN II:</b>															
SW1	off	on	off	on	off	on	off	on	\	E					
SW2	off	on	off	on	off	on	off	on			\	E			
SW3	off	on	off	on	off	on	off	on					Z	%	
<b>PATTERN III:</b>															
SW1					All on				Locations 2, 3, 4, 5, 6, 7 are displayed with solid 'blocks' or the rubout symbols.						
SW2					All on										
SW3					All on										
<b>PATTERN IV:</b>															
SW1					All off				Locations 2, 3, 4, 5, 6, 7 are all blanks.						
SW2					All off										
SW3					All off										



**APPENDIX A SETTINGS OF USER PROGRAMMABLE SWITCHES**

There are three banks of Dip switches each with 8 switch positions located on the keyboard. Facing the front of the keyboard, the following diagram corresponds to the exact arrangement of the switches (also see the keyboard layout in Figure 4-1 and the drawing at the end of Section 6):



**SW1 – OPERATING SWITCHES AND PRINTER BAUD RATE SETTINGS**

Switches 1, 2, 3, 4 of SW1 are the switches which define the four operating options. These options can be selected by the operator at any time by simply depressing the RESET key, as long as the terminal is in an idle mode (i.e. no ongoing operation or communication).

**SW1-1:**

- OFF- All editing keys are set up as local keys and are executable locally. Their ESC sequences are not sent to the computer.
- ON - All editing keys are set up as duplex keys (host controllable) and operate as follows:  
**BLOCK mode:** all editing keys operable locally, their corresponding ESC sequences are not sent to the computer.  
**HALF-DUPLEX CONVERSATION MODE:** All editing keys are executed locally and their corresponding ESC sequences are not sent to the computer (see Table 10).  
**FULL-DUPLEX CONVERSATION MODE:** Editing keys are not executed locally. When a local editing key is depressed, it corresponding ESC sequence is sent to the computer. The terminal executes the function if the computer 'echoes' the ESC sequence.

**SW1 - 2:**

- OFF- Selects the following block transmission sequences:  
 End of line = US (X'1F')  
 End of Transmission = CR (X'0D')
- ON- Selects the following block transmission sequences:  
 End of line = CR, LF, NULL (X'0D', X'0A', X'00')  
 End of Transmission = ETX (X'03')

**SW1-3:**

- OFF- Enables the auto-new-line (NL) operation. When data is entered at the last position of the row, the cursor is moved to the beginning of the next row automatically.
- ON- Disables the auto-new-line (NL) operation. The cursor does not advance to the next row if data is entered at the last position of the row.

This switch does not affect any other cursor movement which is activated by a cursor movement command such as cursor right or left.

**SW1-4:**

- OFF- Selects the ENTER key to operate as the tab (TAB or HT) function and the RETURN key to operate as the combination of the carriage return (CR) and line feed (LF) functions.
- ON - Selects the ENTER key to operate as the carriage return (CR) function and the RETURN key to operate as the carriage return (CR) function only.

**SW1-5 to 8:**

**Printer Port Baud Rate Setting:**

5	6	7	8	Baud Rate
off	off	off	on	9600
off	off	on	off	7200
off	off	on	on	4800
off	on	off	off	3600
off	on	off	on	2400
off	on	on	off	2000
off	on	on	on	1800
on	off	off	off	1200
on	off	off	on	600
on	off	on	off	300
on	off	on	on	150
on	on	off	off	134.5
on	on	off	on	110
on	on	on	off	75
on	on	on	on	50

NOTE: Data size, parity and stop bits selections are the same as that of the primary communication switches

**APPENDIX A**

**SW2 – COMMUNICATION SWITCHES**

SW2-1:

- OFF – 7 bits data
- ON – 8 bits data

SW2-2:

- If parity is enabled (SW2-4 ON):
    - OFF – Odd parity
    - ON – Even parity
  - If parity is disabled (SW2-4 OFF):
    - OFF – Space parity (most significant bit = 0)
    - ON – Mark parity (most significant bit = 1)
- Requires 8 bit data setting.

SW2-3:

- OFF – 1 stop bit
- ON – 2 stop bits

SW2-4:

- OFF – parity disabled
- ON – parity enabled

SW2-5 to 8:

RS232C Communication port baud rate setting:

5	6	7	8	Baud Rate
off	off	off	on	9600
off	off	on	off	7200
off	off	on	on	4800
off	on	off	off	3600
off	on	off	on	2400
off	on	on	off	2000
off	on	on	on	1800
on	off	off	off	1200
on	off	off	on	600
on	off	on	off	300
on	off	on	on	150
on	on	off	off	134.5
on	on	off	on	110
on	on	on	off	75
on	on	on	on	50

**SW3 – CONFIGURATION SWITCHES**

This set of switches initiates the start-up mode of the terminal at power-up configuration.

SW3-1:

- OFF – supports a 1 page memory (or a 1920 characters data storage)
- ON – supports a 2 page memory (or a 3840 characters data storage)

\*SW3-2:

- OFF – 60 Hz screen refresh rate
- ON – 50 Hz screen refresh rate

SW3-3 and 4:

Reserved. Set to OFF position.

\*SW3-5:

- OFF – starts with black screen background
- ON – starts with green screen background

\*SW3-6:

- OFF – Blinking cursor
- ON – Non-blinking cursor

SW3-7:

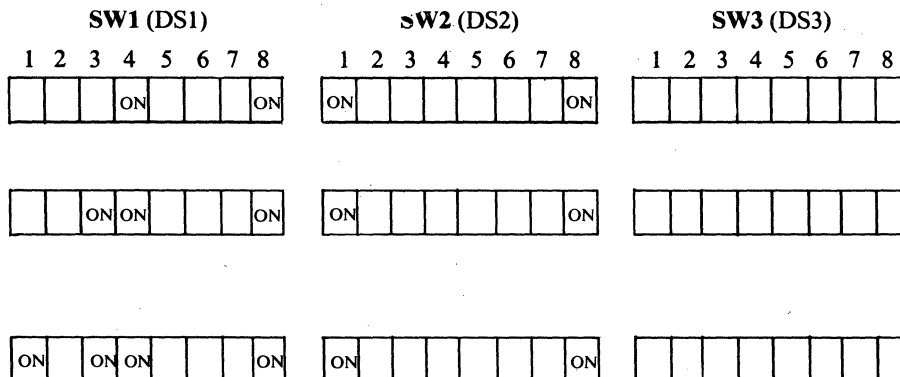
- OFF – starts with Full Duplex
- ON – starts with Half Duplex

SW3-8:

- OFF – starts with Conversation mode
- ON – starts with Block mode

\*These switches are also read when the display mode is being changed (see Section 5.2.12).

**EXAMPLES:** The following are three typical switch settings which operate in most micro-computer and minicomputer environments under full duplex mode. Communication speed is assumed to be 9600 baud with SPACE parity.



SW1-4 selects the RETURN key to generate a CR code.

SW1-4 selects the RETURN key to generate a CR code and SW1-3 disables the auto-new-line operation. (When data entered passes the end of the line, the cursor does not move).

In addition to the above, SW1-1 selects the ESC sequences generated by all editing keys to be sent to the host computer. This is used in an environment where the host computer is given full control of the terminal operations.

---



---

**APPENDIX B: TOP MESSAGE FIELDS LINE  
AND FUNCTION KEY LABELING LINE**

**TOP MESSAGE FIELDS LINE**

The top message field is arranged as follows (see Section 5.2.11):

<u>Column</u>	<u>Description/Message</u>
1	(Attribute for local message field)
2-5	CAPS - Keyboard CAPS mode on LOCK - Keyboard locked
9	* - monitor mode on
10-12	BLK - Block Mode FDX - Full Duplex Conversation Mode HDX - Half Duplex Conversation Mode
14-17	EDIT - Edit Mode on
19-22	PROT - Protect Mode on
24-27	WPRT - Write Protect Mode on
29-32	INS - Insert Mode on
33	(Attribute for Host Message Field)
34-79	(Host Message Field - 46 bytes)
80	(Attribute for data/text entry area)

Columns 2-32 are for local messages

Columns 34-79 are for host messages

**FUNCTION KEY LABELING LINE**

The bottom function key labeling line is arranged as follows (see Section 5.2.11):

<u>Column</u>	<u>Description</u>
1	**
2-9	Labeling field 0
10-11	**
12-19	Labeling field 1
20-21	**
22-29	Labeling field 2
30-31	**
32-39	Labeling field 3
40-41	**
42-49	Labeling field 4
50-51	**
52-59	Labeling field 5
60-61	**
62-69	Labeling field 6
70-71	**
72-79	Labeling field 7
80	**

Column 2-79 can be used continuously as one message field using the ESC z ( aaaa CR command (see Section 5.2.11).

\*\* = Attribute for labeling fields separation and enhancement.



TABLE 1

**TABLE 1**  
**TERMINAL CONTROL CODES**

ASCII codes (in hex.)	Display Symbol	Control Key	Function/Operation
00	(blank)	@,/	NULL
01	SH	A,a	SOH
02	SX	B,b	STX
03	EX	C,c	ETX - Cursor up
04	ET	D,d	EOT
05	EQ	E,e	ENQ - Host check busy, Terminal returns ACK if not busy
06	AK	F,f	ACK - Terminal sends ACK to Host if it is not busy
07	BL	G,g	Sound beeper
08	BS	H,h	Back Space
09	HT	I,i	Tab Cursor
0A	LF	J,j	Line Feed (cursor down)
0B	VT	K,k	Cursor Up
0C	FF	L,l	Cursor Right
0D	CR	M,m	Carriage Return
0E	SO	N,n	Unlock Keyboard
0F	SI	O,o	Lock Keyboard
10	┐	P,p	DLE
11	┌	Q,q	DC1 - XON
12	└	R,r	DC2
13	┘	S,s	DC3 - XOFF
14	├	T,t	DC4
15	┤	U,u	NAK
16	┆	V,v	SYN
17	■	W,w	ETB
18	+	X,x	CAN
19	+	Y,y	EM
1A	—	Z,z	SUB - Clears all with spaces, see ESC +
1B	▨	[ {	ESCape sequence start
1C	▬	▮	FS
1D	┴	] }	GS
1E	┴	^ ~	Cursor Home
1F	▣	→ RUBOUT	New Line

**TABLE 2**  
**ASCII CODES (7 BITS)**

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

**TABLE 3**  
**DEFAULT FUNCTION KEY CODE**

Function Key Code Sequence: *SOH, Function Key Code, CR.*

Function Key	Unshifted Code	Shifted Code
F1	@	H
F2	A	I
F3	B	J
F4	C	K
F5	D	L
F6	E	M
F7	F	N
F8	G	O

**TABLE 4  
GRAPHIC CODES**

Sequence to enter Graphic code: *ESC H Graphic Code.*

Graphic Code	Display Symbol	Display Symbol (in ASCII HEX)
0	T	10
1	L	11
2	T	12
3	T	13
4	T	14
5	T	15
6	T	16
7	■	17
8	+	18
9	T	19
:	—	1A
;	▨	1B
=	—	1C
?	T	1D
	■	1E
	■	1F

**TABLE 5  
CURSOR ROW OR COLUMN CODE**

* r or c	ASCII Character	* r or c	ASCII Character	* r or c	ASCII Character
1	(space)	28	;	55	V
2	!	29	<	56	W
3	"	30	=	57	X
4	#	31	>	58	Y
5	\$	32	?	59	Z
6	%	33	@	60	[
7	&	34	A	61	/
8	'	35	B	62	]
9	(	36	C	63	^
10	)	37	D	64	-
11	*	38	E	65	\
12	+	39	F	66	a
13	,	40	G	67	b
14	-	41	H	68	c
15	.	42	I	69	d
16	/	43	J	70	e
17	0	44	K	71	f
18	1	45	L	72	g
19	2	46	M	73	h
20	3	47	N	74	i
21	4	48	O	75	j
22	5	49	P	76	k
23	6	50	Q	77	l
24	7	51	R	78	m
25	8	52	S	79	n
26	9	53	T	80	o
27	:	54	U		

\* r = row number -  
c = column number

**TABLE 6**  
**SCREEN ATTRIBUTE CODES**

The sequence to enter screen attributes is: *ESC, G, Attribute Code*

<b>Attribute Code</b>	<b>Screen Attribute Effect</b>
(space)	Enter space code (X'20')
0	Normal (green character on black background)
1	Blank (no show)
2	Blink
3	Blank
4	Reverse
5	Reverse and Blank
6	Reverse and Blink
7	Reverse and Blank
8	Underscore
9	Underscore and Blank
:	Underscore and Blink
;	Underscore and Blank
<	Underscore and Reverse
=	Underscore, Reverse and Blank
>	Underscore, Reverse and Blink
?	Underscore, Reverse, Blink and Blank
p	Dim
q	Blank
r	Dim and Blink
s	Blank
t	Dim and Reverse
u	Reverse and Blank
v	Dim, Reverse and Blink
w	Reverse and Blank
x	Dim and Underscore
y	Underscore and Blank
z	Dim, Underscore and Blink
}	Dim, Underscore, Blink, and Blank
	Dim, Underscore and Reverse
}	Underscore, Reverse and Blank
~	Dim, Underscore, Reverse and Blink

TABLE 7

TABLE 7  
VERTICAL SPLIT COLUMN 40 SEPARATION CODE

Sequence: ESC, n, Column Code.

NOTE: This sequence is provided to set at column 40 the separation column for the vertical-split (with 39 column text segments) display window. However, the sequence is recognized at all times and can be executed under other display modes.

Column Code	Display Symbol	Column Code	Display Symbol	Column Code	Display Symbol
(space)	(space)	@	(BLANK)		
!	!	A	SH	a	a
"	"	B	SX	b	b
#	#	C	EX	c	c
\$	\$	D	ET	d	d
%	%	E	EQ	e	e
&	&	F	AK	f	f
'	'	G	BL	g	g
(	(	H	BS	h	h
)	)	I	HT	i	i
*	*	J	LF	j	j
+	+	K	VT	k	k
,	,	L	FF	l	l
-	-	M	CR	m	m
.	.	N	SO	n	n
/	/	O	SI	o	o
0		P	T	p	
1		Q	L	q	
2		R	T	r	
3		S	T	s	
4		T	T	t	
5		U	T	u	
6		V		v	
7	Enter	W	■	w	Enter
8	Attribute	X	+	x	Attribute
9	Code	Y	+	y	Code
:	See Table 6	Z	-	z	See Table 6
;		[	▣	{	
<		\	=	}	
=		]	⊥	~	
>		^	⊥		
?		-	■	RUBOUT	

**TABLE 8  
DISPLAY MODE SELECTION**

*Sequence: ESC, x, Display Mode Code, Horizontal split row (option)\**

Display Mode Code	Display Window	Internal Memory Storage	Display Mode Code	Display Window	Internal Memory Storage
0	Normal display: 24 rows X 80 char.	1920 characters	4	Normal display: 24 rows X 80 char.	3840 characters
1	Horizontal Split: M rows X 80 char., and N rows X 80 char.	1920 characters	5	Horizontal Split: M rows X 80 char., and N rows X 80 char.	3840 characters
2	Vertical Split: 24 rows X 39 char., and 24 rows X 39 char.	1920 characters	6	Vertical Split: 24 rows X 39 char., and 24 rows X 39 char.	3840 characters
3	Vertical Split: 24 rows X 40 char., and 24 rows X 40 char.	1920 characters	7	Vertical Split: 24 rows X 40 char., and 24 rows X 40 char.	3840 characters

\*Horizontal Split Row Number Definition:  
It defines the physical row number on the data/text area of the screen where the lower window starts, i.e. a number = 12 means that the lower split starts at row 12. The limit of the horizontal split number is from 2 to 25. If the horizontal split row number equals 25, the terminal internal memory is split into two 24 rows x 80 character text segments. This is a special mode where the screen data text area can be selected to display one text segment at a time. To select this mode, the second page option is required. Horizontal split row number follows the rule outlined on TABLE 5.

**TABLE 9  
TERMINAL ESCAPE SEQUENCES**

HEX	2	3	4	5	6	7
0		0 clear all tab	@ print unprotected	P print all	,	p print unformatted
1	! write unprotected with attribute*	1 set tab at cursor column	A set special attribute*	Q insert one character space	a	q on insert mode
2	" enable keyboard	2 clear tab at cursor column	B set block mode	R delete line	b	r off insert mode
3	# disable keyboard	3	C set conv. or TTY mode	S send MSG. unprotected	c	s send MSG. all
4	\$ enable CR in funct. key seq.	4 send line unprotected	D set HDX/FDX *	T erase to end of line to space	d	t erase to end-of-line with null
5	% disable CR in funct. key seq.	5 send page unprotected	E insert line	U on monitor mode	e	u off monitor mode
6	& on protect mode	6 send line all	F message to host message field*	V enter protect space column	f	v internal scroll up
7	' off protect mode	7 send page all	G enter attribute *	W delete character	g	w internal scroll down
8	( off write protect mode	8 write SOM (STX)	H enter graphic*	X off monitor mode	h internal paging	x select display mode*
9	) on write protect mode	9 write EOM (ETX)	I back tab cursor	Y erase to end-of-text to space	i tab cursor	y erase to end-of-text with null
A	* clear all to null	: clear unprotected to null	J activate alternate split	Z	j	z message to funct. key or label line*
B	+ clear all to space	; clear unprotected to space	K activate alternate split	[	k	} cursor to SOTX (true home)
C	' clear all to protected space	< on data validation	L print unformatted	\	l	
D	- set cursor to selected window*	= set cursor to active window*	M Read cursor byte	] activate split segment 0	m	} activate split segment 1
E	write unprotected with code *	> off data validation	N on EDIT	^	n enter column 40 code*	~
F	/ read cur. and active seq. #	? read cur. from active window	O off EDIT	-	o	(DEL or Rubout)

\*Multiple character sequence, requires additional characters or codes

TABLE 10

**TABLE 10**  
**KEYBOARD COMMAND CODE SEQUENCES**

<b>KEYS GENERATING SINGLE ASCII CONTROL CODE:</b>		
<b>KEY</b>	<b>CODE SEQUENCE</b>	<b>OPERATION</b>
BACK SPACE	BS (X'08')	Back space cursor
Cursor Left	BS (X'08')	Back space cursor
Cursor Right	FF (X'0C')	Cursor right
Cursor Up (unshifted)	VT (X'0B')	Cursor up
Cursor Down (unshifted)	ETX (X'03')	Cursor up (see Section 5.2.10)
LINE FEED	LF (X'0A')	Cursor down
RETURN (SW1-4 on)	CR, (X'0D')	Cursor to start-of-row
RETURN (SW1-4 off)	CR, LF (X'0D', X'0A')	Cursor to start-of-next row
NEW LINE	US (X'1F')	Cursor to start-of-next row
HOME (unshifted)	RS (X'1E')	Cursor to HOME of window
TAB (unshifted)	HT (X'09')	Tab cursor
ESC	ESC (X'1B')	Generate a ESC code
RUB OUT	DEL (X'7F')	Generate a DEL code
ENTER (SW1-4 on)	CR (X'0D')	Same as RETURN
ENTER (SW1-4 off)	VT (X'09')	Same as TAB
<b>KEYS GENERATING ESCAPE SEQUENCES:</b>		
<b>KEY</b>	<b>CODE SEQUENCE</b>	<b>OPERATION</b>
RESET (shifted)	ESC B	Set block mode
SCRL Up	ESC w	Scroll towards the start-of-text
SCRL Down	ESC v	Scroll towards the end-of-text
HOME (shifted)	ESC {	Cursor to start-of-text
LINE INSERT	ESC E	Insert line
LINE DELETE	ESC R	Delete line
LINE ERASE (unshifted)	ESC T	Erase to end-of-line with spaces
LINE ERASE (shifted)	ESC t	Erase to end-of-line with nulls
PAGE ERASE (unshifted)	ESC Y	Erase to end-of-text with spaces
PAGE ERASE (shifted)	ESC y	Erase to end-of-text with nulls
SEND PAGE (unshifted)	ESC 5	Send unprotected data from
SEND PAGE (shifted)	ESC 7	Send all data from start-of-text to cursor
SEND LINE (unshifted)	ESC 4	Send unprotected data from start-of-line to cursor
SEND LINE (shifted)	ESC 6	Send all data from start-of-text to cursor
SEND MSG (unshifted)	ESC S	Send unprotected data from SOM to EOM labels
SEND MSG (shifted)	ESC s	Send all data from SOM to EOM labels
PRINT (unshifted)	ESC @	Print unprotected data from start-of-text to cursor
PRINT (shifted)	ESC P	Print all data from start-of-text to cursor
TAB (shifted)	ESC I	Back tab cursor
SCRN/EDIT (unshifted)	ESC N	On EDIT mode
SCRN/EDIT (shifted)	ESC O	Off EDIT mode
PAGE (unshifted)	ESC h	Internal paging
PAGE (shifted)	ESC K	Activate alternate split
INS/REP (unshifted)	ESC r	Off insert mode
INS/REP (shifted)	ESC q	On insert mode
SET TAB (unshifted)	ESC l	Set tab at cursor
SET TAB (shifted)	ESC 2	Clear tab at cursor
SOM/EOM (unshifted)	ESC 9	Enter EOM (X'03') mark
SOM/EOM (shifted)	ESC 8	Enter SOM (X'02') mark
DEL/INS CHAR (unshifted)	ESC Q	Insert a space character
DEL/INS CHAR (shifted)	ESC W	Delete a character

Following are some notes on the WY-100 keyboard tables for firmware revision 23-004-01 REV. D.

- 1) The WY-100 keyboard tables are located between 0730H and 07FFH.
- 2) There are two keyboard tables, namely the unshifted and shifted key code tables. Each table contains 104 one-byte entries corresponding to the 104 keys on the keyboard. The key codes are arranged in the two tables in the same order, basically according to the matrix of the keyboard layout. For example, the first entry of the unshifted table contains the lower case "s" and the first entry of the shifted table contains the upper case "S".
- 3) Keys associated with "ESC" sequences (see Table 10 in the Reference Manual) are coded with their associated "ESC" sequence code with the highest order bit set to "1". For example, the DELETE LINE key generates the ESC R sequence and its key code is 0D2H which is the ASCII code 52H (upper case R) with bit 7 set to 1.
- 4) Keys associated with a control code are coded with the same control code, i.e. TAB=09H; LINE FEED = 0AH.
- 5) Function Keys: Unshifted F1-F8 = 80H-87H  
Shifted F1-F8 = 88H-8FH
- 6) Special Key Codes (00H to 07H & 90H are reserved):  

SHIFT	=	00	RESET	=	01
CNTRL	=	00	CAPSLOCK	=	02
BREAK	=	06	ENTER (shifted)	=	04
RETURN	=	9EH	ENTER (unshifted)	=	05
FUNCT	=	9FH			

Following are some examples of key code table alterations:

- 1) Any keys on the keyboard can be disabled by replacing the associated key code entry with 00H.
- 2) Functions of keys can be "exchanged" by swapping key caps and their associated key code positions.
- 3) SHIFT and CTRL keys can not be moved.
- 4) "ESC" sequences also may be used as key functions which can be implemented by entering the key code (ESC sequence + bit 7 set) into the desirable location in the key code table. Locations such as the shifted positions of the numeric pad can be used.

I hope this answers your questions. Please call if you have any additional questions.



THIS KEYBOARD TABLE IS FOR USE IN THE WY-100 TERMINAL FIRMWARE  
RELEASE 23-004-01 REV D. WYSE TECHNOLOGY RESERVE THE RIGHT TO  
MODIFY THIS TABLE WITHOUT PRIOR NOTIFICATION FOR FUTURE RELEASE.

THE WY-100 KEYBOARD TABLE IS COMPOSED OF TWO TABLES LOCATED AT:

UNSHIFTED KEYCODE TABLE : 0730H TO 0797H  
SHIFTED KEYCODE TABLE : 0798H TO 07FFH

\*\*\*\*\* FOR FIRMWARE 23-004-01 REV D \*\*\*\*\*

\*\*\* KEYTAB - KEYCODE TABLE

0730

KEYTAB:

S, C, NEW LINE, LINE FEED, 3, SHIFT, F4, W

0730 73631F0A  
0734 33008377

DB 0073H,0063H,001FH,000AH  
DB 0033H,0000H,0083H,0077H

F7, ERASE LINE, 8, DELETE LINE, =, 3, 9, INSERT LINE

0738 86D438D2  
073C 3D3339C5

DB 0086H,00D4H,0038H,00D2H  
DB 003DH,0033H,0039H,00C5H

J, ., INS/REP, SCRN EDIT, 9, CURSOR LEFT, INS CHAR, I

0740 6A2EF2CE  
0744 3908D169

DB 006AH,002EH,00F2H,00CEH  
DB 0039H,0008H,00D1H,0069H

K, /, \, RUBOUT, 0, EOM, FUNCT, O

0748 6B2F5C7F  
074C 30B99F6F

DB 006BH,002FH,005CH,007FH  
DB 0030H,00B9H,009FH,006FH

H, ., CURSOR DOWN, SET TAB, 8, HOME, CURSOR UP, U

0750 682C0AB1  
0754 381E0375

DB 0068H,002CH,000AH,00B1H  
DB 0038H,001EH,0003H,0075H

G, M, SCRL DOWN, PAGE, 7, CRIGHT, SCRL UP, Y

0758 676DF7E8  
075C 370CF679

DB 0067H,006DH,00F7H,00E8H  
DB 0037H,000CH,00F6H,0079H

F, N, 0, 7, 6, 1, 4, T

0760 666E3037  
0764 36313474

DB 0066H,006EH,0030H,0037H  
DB 0036H,0031H,0034H,0074H

ESC, Z, SEND PAGE, ERASE PAGE, 1, BACKSPACE, SPACE,

0768 1B7AB5D9  
076C 31082009

DB 001BH,007AH,00B5H,00D9H  
DB 0031H,0008H,0020H,0009H

F2, F3, SEND MSG, F5, PRINT, F8, F6, F1

0770 8182D384  
0774 C0878580

DB 0081H,0082H,00D3H,0084H  
DB 00C0H,0087H,0085H,0080H

L, B, 2, 6, 5, 5, ., R

0778 6C623236  
077C 35352E72

DB 006CH,0062H,0032H,0036H  
DB 0035H,0035H,002EH,0072H

D, V, ENTER, RESET, 4, ., -, E

0780 64760501  
0784 342C2D65

DB 0064H,0076H,0005H,0001H  
DB 0034H,002CH,002DH,0065H

A, X, SEND LINE, BREAK, 2, CAPS, CTRL, Q

0788 6178B406  
078C 32020071

DB 0061H,0078H,00B4H,0006H  
DB 0032H,0002H,0000H,0071H

;, ', ], ` , -, [, RETURN, P

0790 3B275D60  
0794 2D5B9E70

DB 003BH,0027H,005DH,0060H  
DB 002DH,005BH,009EH,0070H

#### SHIFT KEY TABLE

S, C, NEW LINE, LINE FEED, #, SHIFT, F4, W

0798 53431F0A  
079C 23008B57

DB 0053H,0043H,001FH,000AH  
DB 0023H,0000H,008BH,0057H

F7, ERASE LINE, 8, DELETE LINE, +, 3, 9, INSERT LINE

07A0 8EF438D2  
07A4 2B3339C5

DB 008EH,00F4H,0038H,00D2H  
DB 002BH,0033H,0039H,00C5H

J, >, INS/REP, SCRN EDIT, (, CURSOR LEFT, DEL CHAR,

07A8 4A3EF1CF  
07AC 2808D749

DB 004AH,003EH,00F1H,00CFH  
DB 0028H,0008H,00D7H,0049H

K, ?, |, RUBOUT, ), SOM, FUNCT, O

07B0 4B3F7C7F  
07B4 29B89F4F

DB 004BH,003FH,007CH,007FH  
DB 0029H,00B8H,009FH,004FH

H, <, CURSOR DOWN, SET TAB, \*, HOME, CURSOR UP, U

```

07B8 483C0AB2      DB  0048H,003CH,000AH,00B2H
07BC 2AFB0355      DB  002AH,00FBH,0003H,0055H
;
; G, M, SCRL DOWN, PAGE, &, CURSOR RIGHT, SCRL DOWN, Y
;
07C0 474DF7CB      DB  0047H,004DH,00F7H,00CBH
07C4 260CF659      DB  0026H,000CH,00F6H,0059H
;
; F, N, 0, 7, ^, 1, 4, T
;
07C8 464E3037      DB  0046H,004EH,0030H,0037H
07CC 5E313454      DB  005EH,0031H,0034H,0054H
;
; ESC, Z, SEND PAGE, ERASE PAGE, !, BACKSPACE, SPACE, TA
;
07D0 1B5AB7F9      DB  001BH,005AH,00B7H,00F9H
07D4 210820C9      DB  0021H,0008H,0020H,00C9H
;
; F2, F3, SEND MSG, F5, PRINT, F8, F6, F1
;
07D8 898AF38C      DB  0089H,008AH,00F3H,008CH
07DC D08F8D88      DB  00D0H,008FH,008DH,0088H
;
; L, B, 2, 6, 8, 5, .. R
;
07E0 4C423236      DB  004CH,0042H,0032H,0036H
07E4 25352E52      DB  0025H,0035H,002EH,0052H
;
; D, V, ENTER ( KEY CLICK ), RESET ( BLOCK ), $, .. -, E
;
07E8 445604C2      DB  0044H,0056H,0004H,00C2H
07EC 242C2D45      DB  0024H,002CH,002DH,0045H
;
; A, X, SEND LINE, BREAK, @, CAPS, CTRL, Q
;
07F0 4158B606      DB  0041H,0058H,00B6H,0006H
07F4 40020051      DB  0040H,0002H,0000H,0051H
;
; :. ", }, ~, _, {, RETURN, P
;
07F8 3A227D7E      DB  003AH,0022H,007DH,007EH
07FC 5F7B9E50      DB  005FH,007BH,009EH,0050H
;
0000      END

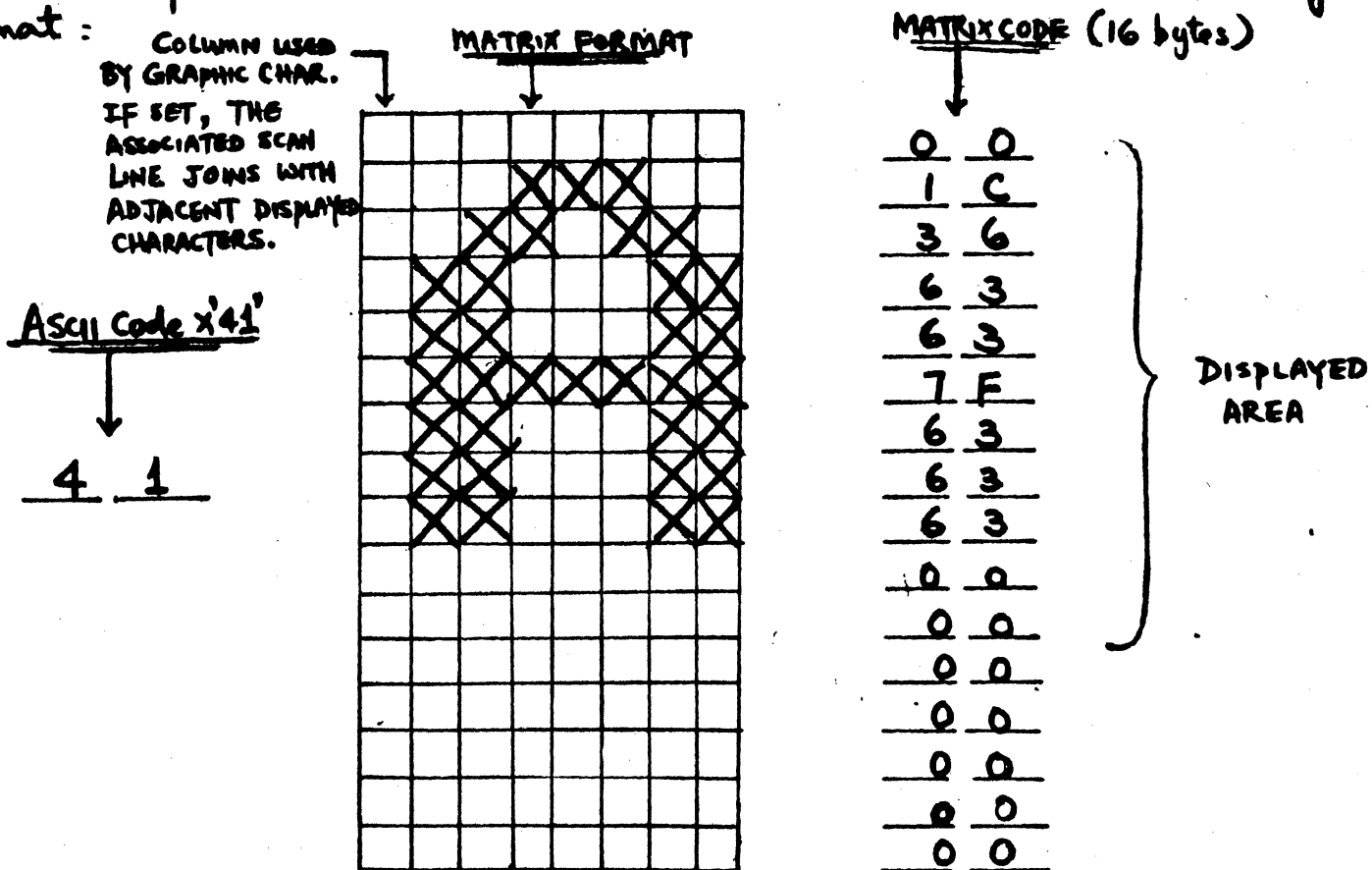
```

↑  
TAB

1. Character matrix of each ASCII Code occupies 16 locations in the character generator prom in ascending order, i.e. ASCII code '00' occupies the '0000' to '000F' locations of the prom. ASCII code '01' occupies the '0010' to '001F' locations of the prom.

2. The 128 character matrix of ASCII code '00' to '7F' are documented in the following pages with the format:

3. PROM #  
23-002-01



APPLICATION		UNLESS OTHERWISE NOTED				DWN		ENGR CHANGES				<b>WYSE TECHNOLOGY</b> TITLE <b>CHARACTER GENERATOR            PROM MATRIX</b>			
TEXT ASSY	USED ON	DIMENSIONS ARE IN				CHK	REV	CHG NO.	DATE	APPD					
		ANG	2 PLC	3 PLC		ENGR	B		12/1/77						
		±	±	±	±	APPD									
		SCALE:				APPD									
		MATERIAL				FINISH									
						DO NOT SCALE DRAWING				SIZE	SHT	DRAWING NO.	REV		
										A	1 OF 65	23-002-1	B		

TEM/ FIND NO.	QTY PER ASSY				UNIT OF MEAS	KEY (1)	KEY (2)	PART NUMBER/ REFERENCE DOCUMENT	TITLE/DESCRIPTION	REMARKS

NOTES: REFER TO THE APPLICATION NOTE ATTACHED.

<b>APPLICATION</b>		<b>UNIT OF MEASURE:</b>		<b>DWN</b>		<b>ENGR CHANGES</b>				<b>WYSE TECHNOLOGY</b>			
NEXT ASSY	USED ON	01-EACH	04-BULK	CHK	ENGR	REV	CHG NO.	DATE	APPD	TITLE WY-100/ ADDS EMULATION FIRMWARE PACKAGE Part # 23-018-01			
		02-INCH	05-AS REQ'D	APPD		A	38	1/23/83					
		03-FEET	09-OTHER (see notes)	APPD									
		<b>KEY 1:</b>		<b>KEY 2:</b>						SIZE	SHT. /	BILL OF MAT'L NO.	REV
		A-WITH B/M								A	OF 16	88-008-01	A
		D-WITHOUT B/M											
		R-REFERENCE											
		S-SPECIFICATION											
		O-OTHER (see notes)											

RELEASE

## APPLICATION NOTE - WY-100/ADDS EMULATION FIRMWARE

( REV. A 1/21/83 )

### PURPOSE

This application note documents an ADDS terminal compatible firmware prom which operates in the WY-100 terminal.

### INSTALLATION

The ADDS terminal compatible firmware replaces the standard WY-100 firmware prom, serial no. 23-004-01. To install the ADDS terminal compatible firmware, follow these steps:

1. Turn off the WY-100 terminal and remove the top cover of the terminal.
2. Remove the firmware prom at location 2B of the terminal logic board.  
The standard WY-100 firmware prom is labeled with serial no. 23-004-01.
3. Insert the ADDS terminal compatible prom into location 2B.
4. Set up the switch settings which are outlined in page 11 of this application note and the standard WY-100 reference manual.
5. Replace the top cover and you are now have an ADDS compatible terminal.

If you has further questions on the ADDS terminal compatible firmware, please contact the WYSE local distributor or the WYSE Technical Support, telephone number 408-946-3075.

APPLICATION NOTE: WY-100/ADDS TERMINAL LINE EMULATION FIRMWARE

TABLE OF CONTENT

	PAGE
PURPOSE	2
COMPACTIBILITY	2
ENHANCED FEATURES	2-6
TERMINAL COMMANDS	3-6
EXCEPTIONS AND CONVENTIONS	6-7
KEYBOARD COMMANDS	7-10
SWITCH SETTINGS	11

TABLES AND ILLUSTRATIONS

	PAGE
KEYBOARD LAYOUT	12
SCREEN ATTRIBUTE CODE TABLE	13
GRAPHIC CODE TABLE	14

## PURPOSE

This application note documents the special WY-100/ADDS terminal firmware which emulates the ADDS terminal line.

## COMPACTIBILITY

The WY-100/ADDS firmware is estimated to provide the following result with respect to the terminal being emulated:

ADDS Regent 60 - 70% Compatible  
ADDS Regent 40 - 90% Compatible  
ADDS Regent 25 - 95% Compatible  
ADDS Regent 20 - 95% Compatible  
Viewpoint 30 - 95% Compatible

Although this piece of firmware is designed to emulate an ADDS terminal in most operating environments, the user should study and compare this firmware with the particular ADDS terminal to be replaced.

Wyse Technology makes no representation or warranties with respect to the contents hereof and disclaims any implied warranties of fitness for any particular application. Wyse Technology reserves the right to make changes to this piece of firmware without prior notice.

## ENHANCED FEATURES

In addition to the ADDS terminal emulation functions, almost all the enhanced features which are standard in the WY-100 terminal remain the same except the command sequences (escape sequences) to activate them has been changed to avoid conflict with those in the ADDS terminals. These enhanced features are:

1. Horizontal and vertical split screen modes and the abilities to operate all terminal function within a split window.
2. 26th row function key labelling fields.
3. Top row host message field.
4. Special screen enhancement attributes (using ESC A).
5. Protect form data entry.
6. Soft function keys (totally 16 software programmable function key-positions).
7. Automatic data validation capability (supported only with 2nd page option).

Although the command sequences are different, these features are functionally described in the Wyse Terminal Reference Manual.



APPLICATION NOTE: WY-100/ADDS TERMINAL LINE EMULATION FIRMWARE

TERMINAL COMMANDS

This section addresses the WY-100/ADDS firmware command sequences comparing to the ADDS terminal line. The standard WY-100 terminal command sequences are listed for reference purposes in the standard WY-100 Terminal Reference Manual. Compatible commands in the ADDS terminals are checked with an "X" and deviations are listed in the following section.

COMMAND	WY-100/ADDS	VIEWPOINT	ADDS20/25	ADDS 40	ADDS 60	STANDARD WY-100
Abort Transmit/						
Print Local	Reset Key					Reset Key
Address, Absolute	ESC Y r C	X	X	X	X	ESC=rC
Address, Horiz'l	DLE C	X	X	X	X	
Address, Verti'l	VT r	X	X	X	X	
Attribute, Set						
Video	ESC Ø a*	X*		X	X	ESC G a
Bell	BEL	X	X	X	X	BEL
Carriage Return/						
New Line	CR	X	X	X	X	CR
Cursor Back	NAK or BS	X	X	X	X	BS
Cursor Down	LF	X	X	X	X	LF
Cursor Forward	ACK	X	X	X	X	FF
Cursor Home	SoH	X	X	X	X	RS
Cursor Up	SUB	X	X	X	X	VT
Delete Character						
(In Line/Field)	ESC E				X	ESC W
Delete Line	ESC l			X	X	ESC R
Erase All	FF	X	X	X	X	ESC*
Erase To End Of						
Field/Line	ESC K	X	X	X	X	ESC T
Erase To End of						
Page	ESC k	X	X	X	X	ESC Y
Erase Variable/						
Unprotected Data	ESC G				X*	ESC ;
Insert Character						
(In Line/Field)	ESC F				X*	ESC Q
Insert Line	ESC M			X	X	ESC E
Keyboard Lock	ESC 5	X	X	X	X	ESC #
Keyboard Unlock	ESC 6	X	X	X	X	ESC "
Line Drawing						
Mode OFF	ESC 2				X*	
Line Drawing						
Mode ON	ESC l				X*	
Line Feed	LF	X	X	X	X	LF
Mode,						
Conversational	ESC V				X	ESC C
Mode, Local						
(Block)*	ESC t			X	X	ESC B
Mode, Monitor						
OFF	Shifted					
	Reset Key	X*	X*	X*	X*	ESC u

APPLICATION NOTE: WY-100/ADDS TERMINAL LINE EMULATION FIRMWARE

COMMAND	WY-100/ADDS	VIEWPOINT	ADDS 20/25	ADDS 40	ADDS 60	STANDARD WY-100
Mode, Monitor ON	Shifted Reset Key	X*	X*	X*	X*	ESC U
Mode, ON LINE*	ESC T			X	X	ESC C
Print All (Local)	ESC x				X*	ESC P
Print Variable/Unprotected	ESC X				X*	ESC@
Printer ON (Aux. ON)	DC 2		X*	X*	X*	
Printer OFF (Aux. OFF)	DC 4		X*	X*	X*	
Status Line Disable	ESC B			X	X	ESC Aa
Status Line Enable	ESC b			X	X	ESC Aa
Display Control Code	ESC Z Code		X	X	X	ESC Code
Trasmit Page Transparent	ESC DC1				X*	ESC 7
Print OFF Transparent	ESC 4	X	X	X	X	
Print ON	ESC 3	X	X	X	X	
Tag Bit Reset	SI					ESC (
Tag Bit Set	SO	X				ESC )
TAB	HT				X*	
BACK TAB	ESC O				X*	

WY-100 FEATURES:

ON Protect Mode	ESC &					ESC &
OFF Protect Mode	ESC '					ESC '
ON Write Protect/Tag Bit Set	ESC )					ESC )
OFF Write Protect/Tag Bit ReSet	ESC (					ESC (
Erase All to Protected Spaces	ESC ,					ESC ,
Set Cursor to Active Windows	ESC-nrc					ESC-nrc
Write Unprotected with Attribute	ESC ! a					ESC ! a
Write Unprotected with Code	ESC . code					ESC . code
Read Cursor and Active Seg. #	ESC /					ESC /
Write SOM (STX)	ESC 8					ESC 8

APPLICATION NOTE: WY-100/ADDS TERMINAL LINE EMULATION FIRMWARE

COMMAND	WY-100/ADDS VIEWPOINT	ADDS 20/25	ADDS 40	ADDS 60	STANDARD WY-100
Write EOM (ETX)	ESC 9				ESC 9
ON Data Validation Mode	ESC <				ESC <
OFF Data Validation Mode	ESC >				ESC >
Read Cursor Position	ESC ?				ESC ?
Erase Unprotected to Space	ESC G				ESC ;
Erase to End of Line with Null	ESC I				ESC t
Erase to End of Page with Null	ESC i				ESC y
Erase Unprotected to Null	ESC g				ESC :
Set "typewriter" Column Tab	ESC a				ESC 1
Clear "typewriter" Column Tab	ESC' (060H)				ESC 2
Clear All "typewriter" Tabs	ESC P				ESC Ø
Set Special Screen Attributes	ESC Qna				ESC Ana
Enter a Column of Protected Space	ESC W				ESC V
Send Message, Unprotected	ESC S				ESC S
Send Message, All	ESC s				ESC s
Send Line, Unprotected	ESC \				ESC \
Send Line, All	ESC ^				ESC ^
Send Page, Unprotected	ESC ]				ESC 5
Send Page, All	ESC -				ESC 7
Internal Paging (2 page mode)	ESC j				ESC h
Read Cursor Byte	ESC m				ESC M
Enter Column 40 with Code	ESC n a				ESC n a
Erase All to Null	ESC O				ESC *
Print Unformatted	ESC p				ESC p
ON Insert Mode	ESC q				ESC q
OFF Insert Mode	ESC r				ESC r
Internal Scroll Up (2 page)	ESC v				ESC v
Internal Scroll Down (2 page)	ESC w				ESC w

COMMAND	WY-100/ADDS	VIEWPOINT	ADDS 20/25	ADDS 40	ADDS 60	STANDARD WY-0100
Select Display						
Mode	ESC y n					ESC x n
Function Key						
Set Up	ESC z					ESC z
Activate Split						
Segment 0 or 1	ESC[ or ESC}					ESC] or ESC}
Cursor to Start						
of Text	ESC {					ESC {

## EXCEPTIONS AND CONVENTIONS

### VIEWPOINT 30

This emulation firmware contains all the functions of those in the Viewpoint 30 except the ESC 0 function. The ESC 0 function in the Viewpoint 30 terminal selects the attribute enhancement for the "tag" bit of each character. This conflicts with ESC 0 sequence in the ADDS 40 terminal. The WY-100/ADDS firmware select to use the ESC 0 sequence to be compatible to those in the ADDS40. Characters with the "tag" bit on appear in half intensity in the Wyse Terminal and is not alterable by the ESC 0 sequence.

### MONITOR MODE

The WY-100/ADDS firmware uses the shifted RESET key to toggle the monitor mode on and off.

### AUX/TRANSPARENT PRINT MODE

There is only one transparent print mode supported by the WY-100/ADDS firmware. It can be enabled either by the DC2 or ESC3 sequence and disabled either by the DC4 or ESC4 sequence.

Once the transparent mode is enabled, all subsequently received data will be retransmitted to the printer or AUX port. No data will be entered onto the screen and the keyboard is locked except the RESET key.

Pressing the RESET key during the transparent print operation terminates the transparent print mode. The terminal is back to the normal mode again.






The transparent print mode can be activated only the by host computer. X-ON/X-OFF are sent from the terminal if the printer is detected to be busy/not busy.

### SCREEN ATTRIBUTES;

The screen attribute codes follow the same convention as those in the ADDS 40 and 60.

GRAPHIC LINE DRAWING MODE

The graphic line drawing mode follows the same convention as those in the ADDS 40 and 60 except that all line graphic characters appear in NORMAL intensity only. Also, the WY-100/ADDS firmware supports five additional symbols for bar charts and enhanced line graphics:

GRAPHIC CODE	VISUAL EFFECT
l	 (solid block)
p	 (quarter tone block)
t	 (double horizontal line)
x	 (double vertical line)
	 (half tone block)

ADDS 60

The FORM, PAGE, MESSAGE and MODIFY modes of the ADDS Regent 60 Terminal are not supported. The PAGE mode and LOCAL mode sequences are supported to enable the BLOCK mode operation which is standard in the WY-100 terminal. (See the WY-100 Reference Manual)

ENHANCED FEATURES OF THE WY-100

Command sequences which activate other standard WY-100 features are modified to avoid conflict with sequences in the ADDS terminals. For example, the split screen mode can be selected with the ESC y sequence instead of the ESC x sequence. Also, the ESC a sequence is used to set column tab position instead of the ESC l sequence.

**KEYBOARD COMMANDS**

All the keys on the WY-100/ADDS keyboard are set up as "duplex" keys; their key-codes or command sequences are transmitted to the best computer. Under the full duplex conversational or on-line mode, key functions are not executed unless they are "echoed" from the host computer. Under the half duplex conversational mode, key functions are executed and their associated keycode and command sequences are transmitted to the host computer.

The following table lists the keycode or key command sequences:

WY100/ADDS	ACTION/KEYCODE/COMMAND SEQUENCES	STANDARD WY100
Alphanumeric Keys and Unshifted	Send corresponding key code to the host Computer; i.e. the "A" key, (unshifted)	Same
Numeric Pad Keys	sends out a hex. 61 code.	
CTRL	Generate control code if pressed simultaneously with certain alphanumeric key.	Same
Caps Lock	All alpha keys generate upper case alpha characters.	Caps Only
Shift	Invert the polarity of the alpha key code. If the Caps mode is on, pressing the Shift key simultaneously with any alpha key generates the lower case alpha key code. If the Caps mode is off, the result is inverted.	

WY100/ADDS	ACTION/KEYCODE/COMMAND SEQUENCES	STANDARD WY100
ESC	ESC CODE	ESC
TAB	HT CODE	HT
TAB (SHIFTED)	ESC Ø (BACK TAB FUNCTION)	ESC I
BREAK	SEND 250 MS. BREAK SIGNAL	SAME
BACK SPACE	BS CODE	SAME
LINE FEED	LF CODE	SAME
RUBOUT	RUBOUT OR DEL CODE	SAME
RETURN	CR CODE OR CR, LF CODE SEQUENCE (SEE SWITCH SETTINGS)	SAME
NEW LINE	CR CODE	US
CURSOR ↑	SUB CODE	VT
CURSOR ↓	LF CODE	SAME
CURSOR →	ACK CODE	FF
CURSOR ←	<del>NAK</del> BS CODE	<del>SAME</del> BS
HOME	SOH CODE	RS
SCRN EDIT	ESC T (ON LINE)	ESC Ø
SCRN EDIT (SHIFTED)	ESC t (LOCAL)	ESC N
SET TAB	ESC a	ESC l
SET TAB (SHIFTED)	ESC ` (HEX 60H)	ESC 2
PAGE	ESC j (INTERNAL PAGING)	ESC h
PAGE (SHIFTED)	ESC J (SELECT ALTERNATE SPLIT WINDOW)	ESC K
INS/REP	ESC r (OFF INSERT MODE)	SAME
INS/REP (SHIFTED)	ESC q (ON INSERT MODE)	SAME
DEL/INS. CHAR	ESC F (INSERT CHARACTER, IN LINE/FIELD)	ESC Q
DEL/INS CHAR (SHIFTED)	ESC E (DELETE CHARACTER, IN LINE/FIELD)	ESC W
SCRL ↑	ESC W (INTERNAL SCROLLING UP)	SAME
SCRL ↓	ESC V (INTERNAL SCROLLING DOWN)	SAME
SEND PAGE	ESC ] (SEND PAGE, UNPROTECTED)	ESC 5
SEND PAGE (SHIFTED)	ESC _ (SEND PAGE, ALL)	ESC 7
SEND LINE	ESC \ (SEND LINE, UNPROTECTED)	ESC 4
SEND LINE (SHIFTED)	ESC ^ (SEND LINE, ALL)	ESC 6
SEND MSG	ESC S (SEND MESSAGE, UNPROTECTED)	SAME
SEND MSG (SHIFTED)	ESC s (SEND MESSAGE, ALL)	SAME
LINE INSERT	ESC M (INSERT LINE)	ESC E
LINE DELETE	ESC l (DELETE LINE)	ESC R
LINE ERASE	ESC K (ERASE TO END OF LINE WITH SPACES)	ESC T
LINE ERASE (SHIFTED)	ESC I (ERASE TO END OF LINE WITH NULLS)	ESC t
PAGE ERASE	ESC k (ERASE TO END OF PAGE WITH SPACES)	ESC Y
PAGE ERASE (SHIFTED)	ESC i (ERASE TO END OF PAGE WITH NULLS)	ESC y
ENTER	CR or HT CODE (SEE SWITCH SETTINGS)	SAME
ENTER (SHIFTED)	TOGGLE THE KEY CLICK OPTION ON/OFF	SAME

---

WY100/ADDS                      ACTION/KEY CODES/COMMAND SEQUENCES                      STANDARD WY100

---

<u>FUNCTION KEYS:</u>	GENERATE THE FOLLOWING SEQUENCES TO THE HOST COMPUTER.		
F1	STX 1	CR* (SEE SWITCH SETTING)	USE ADM-31
F2	STX 2	CR*	CONVENTION
F3	STX 3	CR*	↓
F4	STX 4	CR*	
F5	STX 5	CR*	
F6	STX 6	CR*	
F7	STX 7	CR*	
F8	STX 8	CR*	
F1 (SHIFTED)	STX ↓	CR*	
F2 (SHIFTED)	STX "	CR*	
F3 (SHIFTED)	STX #	CR*	
F4 (SHIFTED)	STX \$	CR*	
F5 (SHIFTED)	STX %	CR*	
F6 (SHIFTED)	STX &	CR*	
F7 (SHIFTED)	STX '	CR*	
F8 (SHIFTED)	STX (	CR*	

FUNCT KEY                      GENERATES STX X CR\* sequence if press preceding alpha numeric key X.

NOTE: F1 - F8 keys can be soft programmable; see WY-100 Reference Manual.

\* Set switch SW1-2 to the ON position to delete the CR Code; to be compatible to the viewpoint 30 terminal.

WY100/ADDS	ACTION/KEYCODES/COMMAND SEQUENCES	STANDARD WY-100
------------	-----------------------------------	--------------------

NUMERIC PAD:

UNSHIFTED 0, 1, 2,  
3, 4, 5, 6, 7, 8,  
9, ,

THE UNSHIFTED KEYS ON THE NUMERIC PAD GENERATE SINGLE CODE ASSOCIATED WITH THEIR KEY LEGEND. IF DEPRESSED WITH THE SHIFTED KEY, IT GENERATES FUNCTION KEY SEQUENCES SIMILAR TO THOSE IN THE ADDS25.

SINGLE NUMERIC KEY CODE ONLY

0 (SHIFTED)

1  
2  
3  
4  
5  
6  
7  
8  
9  
.  
,



STX	0	CR*	OR	ESC	0	(SEE ** SWITCH SETTING)
STX	1	CR*	OR	ESC	1	
STX	2	CR*	OR	ESC	2	
STX	3	CR*	OR	ESC	3	
STX	4	CR*	OR	ESC	4	
STX	5	CR*	OR	ESC	5	
STX	6	CR*	OR	ESC	6	
STX	7	CR*	OR	ESC	7	
STX	8	CR*	OR	ESC	8	
STX	9	CR*	OR	ESC	9	
STX	.	CR*	OR	ESC	.	
STX	,	CR*	OR	ESC	,	



\* IF SW1-2 is ON, the CR Code will be deleted to be compatible to the Viewpoint 3 Terminal.

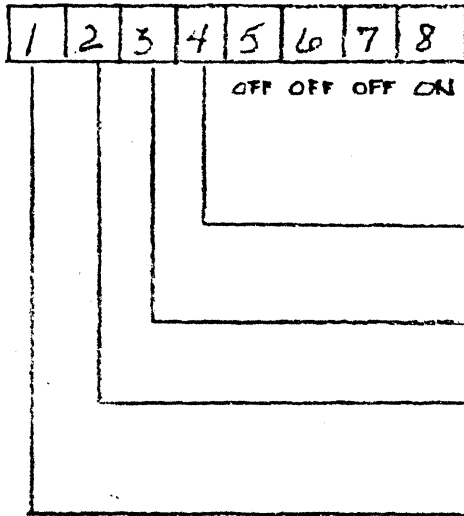
\*\* SW3-4 = ON, Numeric pad generates STX, Code, CR Sequence.  
SW3-4 = OFF, Numeric pad generates ESC, Code Sequence.



SWITCH SETTINGS

The Dip Switch located at the upper right hand cover of the keyboard are assigned as follows:

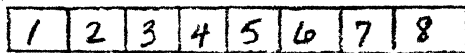
SW1 (DS1)



Local  
Print Baud Rate  
9600 Baud (See WY-100 Reference Manual for additional rate settings.)

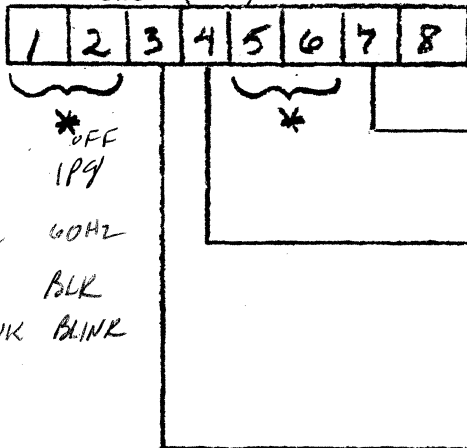
- ON - ENTER = CR ; RETURN = CR
- OFF - ENTER = TAB or HT CODE; RETURN = CR, LF
- ON - Disable AUTO CR LF for CR Code
- OFF - Enable AUTO CR LF for CR Code
- ON - Delete CR Code for Function Key Sequence
- OFF - CR as Terminator for Function Key Sequence
- ON - Disable AUTO Scroll
- OFF - Enable AUTO Scroll

SW2 (DS2)



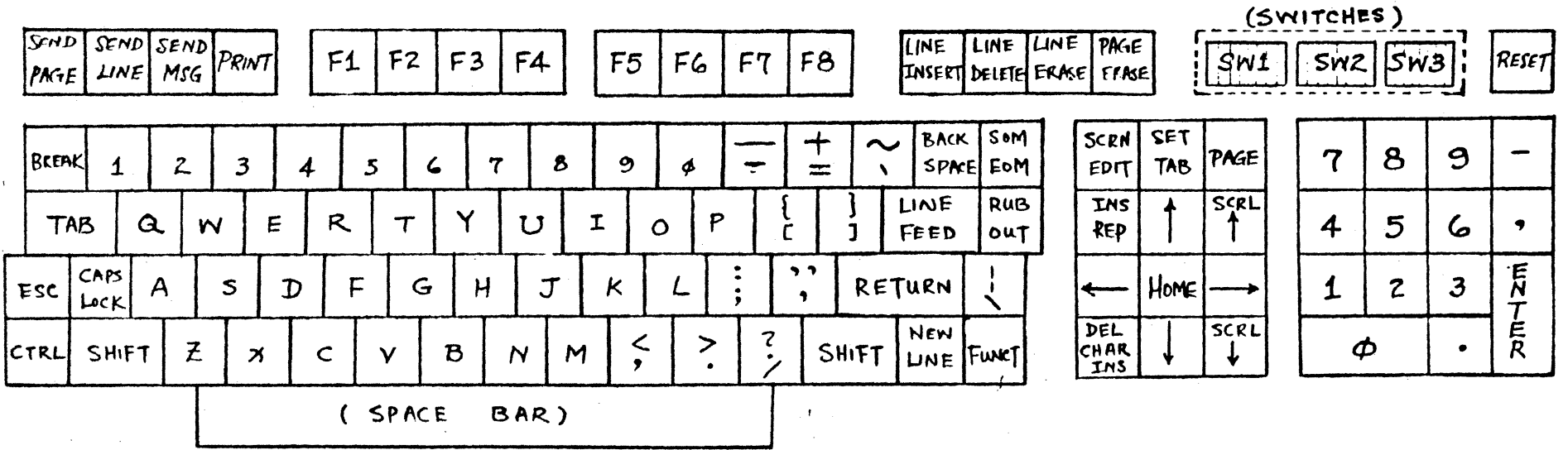
SAME AS STANDARD WY-100 FIRMWARE (BAUD RATES)  
(See WY-100 Reference Manual) PG 2-3

SW3 (DS3)



ON \* OFF  
1 = 2PG 1PG  
2 = 50HZ 60HZ  
5 = GRN BLK  
6 = NOBLINK BLINK

- ON - Block or Local Mode
- OFF - Conversational (CONV) or On-line Mode
- ON - Half Duplex
- OFF - Full Duplex
- ON - Shifted Numeric Keys generate STX, Code, CR
- OFF - Shifted Numeric Keys generate ESC, Code
- ON - Compatible to ADDS Console 580 E0t and STX are used to lock/unlock KB.
- OFF - No ADDS Console 580 Compatibility.



WYSE TERMINAL KEYBOARD LAYOUT

## SCREEN ATTRIBUTE CODE TABLE\*

ATTRIBUTE CODE	IN HEX	SCREEN EFFECT
@	40	Normal (green character on block)
`	60	Normal, underlined
P	50	Reverse Video
P	70	Reverse Video, underlined
R	52	Reverse Video, blink
r	72	Reverse, blink, underline
B	42	Blink
b	62	Blink, underline
A	41	Half intensity
C	43	Half intensity, blink
Q	51	Half intensity
S	53	Half intensity, reverse, blink
q	71	Half intensity, reverse, underline
s	73	Half intensity, reverse, blink, underline
a	61	Half intensity, underlined
c	63	Half intensity, underline, blink
D	44	Zero intensity (blank)
T	54	Zero intensity (reverse video)

\* Following the ADDS 40/60 convention sequence used: ESC 0 attribute code

GRAPHIC CODE TABLE\*

GRAPHIC CODE	HEX CODE	SCREEN SYMBOL
@	40	┌
D	44	┐
H	48	┌┐
L	4C	└└
P	50	┌┐└└
T	54	┌┐└└└└
X	58	┌┐└└└└└└
\	5C	┌┐└└└└└└└└
.	60	┌┐└└└└└└└└└└
d	64	┌┐└└└└└└└└└└└└
h	68	┌┐└└└└└└└└└└└└└└
l	6C	┌┐└└└└└└└└└└└└└└└└
p	70	┌┐└└└└└└└└└└└└└└└└└└
t	74	┌┐└└└└└└└└└└└└└└└└└└└└
x	78	┌┐└└└└└└└└└└└└└└└└└└└└└└
	7C	┌┐└└└└└└└└└└└└└└└└└└└└└└└└

\* First 12 Symbols follows ADDS 40/60 convention.  
 ESC 1 enables line graphic mode.  
 ESC 2 enables line graphic mode.