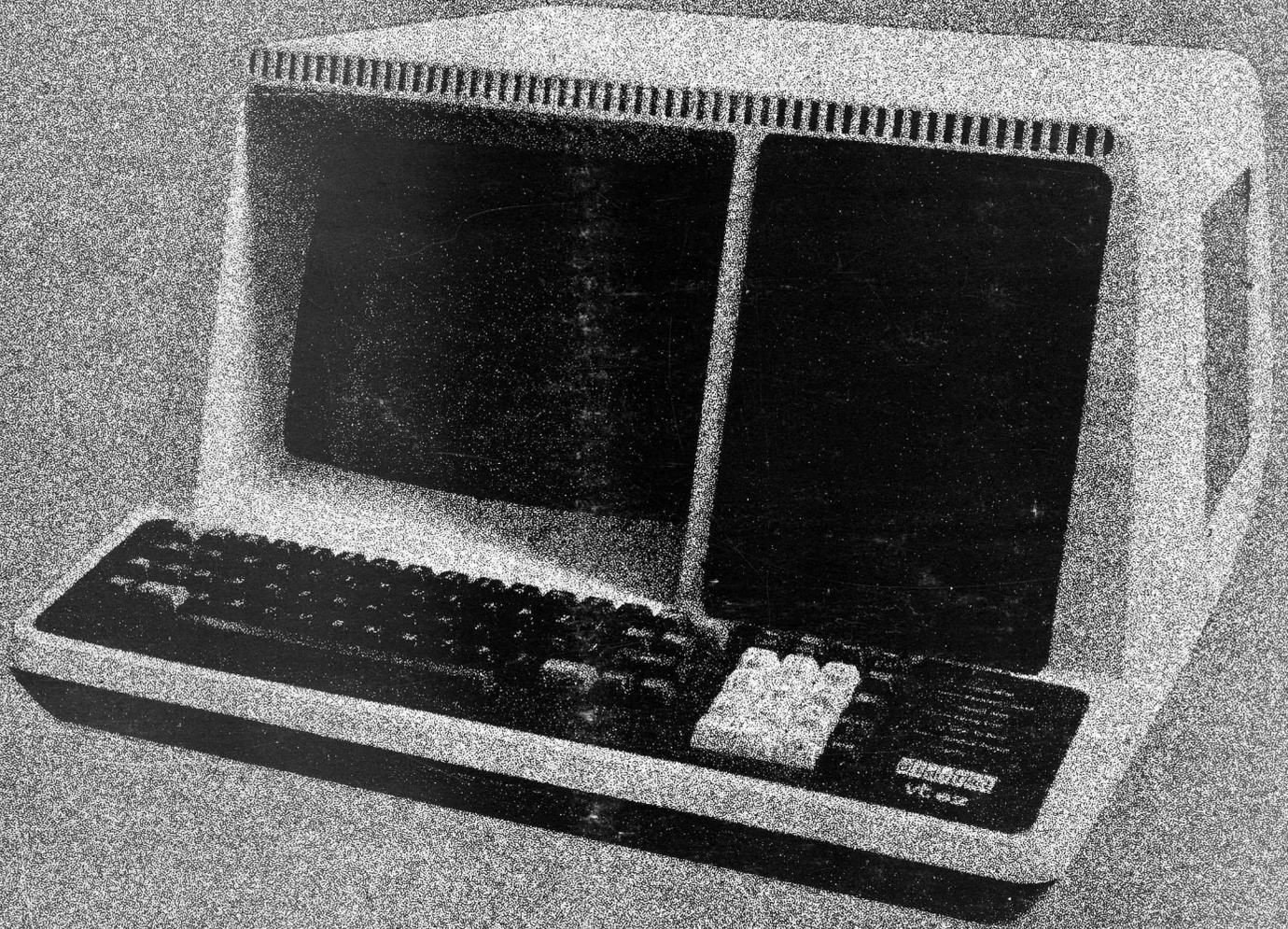


# VT62 DECscope User's Guide



digital

# **VT62 DECscope User's Guide**

Copyright © 1978 by Digital Equipment Corporation

The material in this manual is for informational purposes and is subject to change without notice.

Digital Equipment Corporation assumes no responsibility for any errors which may appear in this manual.

Printed in U.S.A.

**This document was set on DIGITAL's DECset-8000 computerized typesetting system.**

The following are trademarks of Digital Equipment Corporation, Maynard, Massachusetts:

DIGITAL	DECsystem-10	MASSBUS
DEC	DECSYSTEM-20	OMNIBUS
PDP	DIBOL	OS/8
DECUS	EDUSYSTEM	RSTS
UNIBUS	VAX	RSX
	VMS	IAS

# CONTENTS

Page

## CHAPTER 1 INTRODUCTION

1.1	GENERAL DESCRIPTION.....	1-1
1.1.1	VT62 System Description .....	1-1
1.2	FEATURES .....	1-5
1.2.1	Control .....	1-5
1.2.2	Formatted Screen .....	1-5
1.2.3	Input Fields.....	1-6
1.2.4	Automatic Field Checking.....	1-6
1.2.5	Error Messages.....	1-6
1.2.6	Program – Controlled Screen Layout.....	1-7
1.2.6.1	Display Area .....	1-7
1.2.6.2	Display Area Menu Selection .....	1-7
1.2.6.3	Dynamic Displays .....	1-8
1.2.6.4	Form Area .....	1-8
1.2.7	System/Terminal Configuration.....	1-9
1.2.8	Power-Up State .....	1-10
1.2.9	Data Transmission .....	1-10
1.2.10	Printer Option .....	1-10
1.3	SPECIFICATIONS .....	1-10
1.3.1	VT62 Overall Specifications.....	1-10
1.3.2	20 mA Current Loop Specifications .....	1-14
1.3.2.1	Baud Rates vs. Distance for Integral 20 mA Loop .....	1-15
1.3.2.2	20 mA Adapter Card with Cable (BN52B) Specifications .....	1-15
1.3.3	EIA (With Control) Card Specifications (BN61B).....	1-16
1.4	Documentation Ordering Information.....	1-17

## CHAPTER 2 INSTALLATION

2.1	SITE PLANNING.....	2-1
2.2	UNPACKING AND INSPECTION.....	2-1
2.3	INSTALLATION AND SET-UP.....	2-4
2.4	ACCEPTANCE TEST PROCEDURES.....	2-11
2.4.1	VT62 OFF-LINE ACCEPTANCE TESTS.....	2-11
2.4.1.1	Set-Up Procedure .....	2-11
2.4.1.2	Self-Test Procedure.....	2-13
2.4.1.3	Flashing Cursor Check .....	2-15
2.4.1.4	Display All Check.....	2-16
2.4.1.5	CAPS LOCK Check .....	2-16

## CONTENTS (Cont)

	<b>Page</b>
2.4.1.6	OFF-LINE Operating Mode Check.....2-16
2.4.1.7	Escape, Control, and Function Commands.....2-18
2.4.1.8	Escape, Control, and Function Command Checks .....2-18
 <b>CHAPTER 3 OPERATION, PROGRAMMING, AND OPERATOR MAINTENANCE</b>	
3.1	<b>CONTROLS AND INDICATORS.....3-1</b>
3.1.1	Operator Controls.....3-1
3.1.2	Keyboard.....3-3
3.1.2.1	Alphabetic Keys.....3-5
3.1.2.2	Numeric/Symbol Keys.....3-5
3.1.2.3	Function Keys.....3-6
3.1.2.4	Auxiliary Keyboard.....3-7
3.1.2.5	The REPEAT Key.....3-9
3.1.3	Indicators.....3-10
3.2	<b>OPERATING PROCEDURE.....3-10</b>
3.2.1	OFF-LINE.....3-10
3.2.2	ON-LINE.....3-12
3.3	<b>USER PROGRAMMING.....3-12</b>
3.4	<b>USER MAINTENANCE.....3-13</b>
3.4.1	Irregularities in the Appearance of the Screen .....3-13
3.5	<b>DEC MAINTENANCE ARRANGEMENTS.....3-13</b>
 <b>APPENDIX A VT62 FUNCTIONAL SPECIFICATION</b>	
A.1	<b>GENERAL.....A-1</b>
A.2	<b>SCREEN FORMAT.....A-2</b>
A.3	<b>CURSOR.....A-2</b>
A.4	<b>MENU DEFINITIONS.....A-2</b>
A.5	<b>FORM DEFINITIONS.....A-3</b>
A.5.1	Form Field Attributes.....A-3
A.6	<b>DISPLAY AREA USAGE.....A-4</b>
A.6.1	Menu Mode.....A-4
A.6.1.1	Menu Attributes.....A-4
A.6.2	Non-Menu Mode.....A-4
A.7	<b>FORM AREA USAGE.....A-5</b>
A.7.1	Form Mode.....A-5
A.7.2	Packed and Unpacked Transmissions.....A-5
A.7.3	Auto-Transmit Mode.....A-6
A.8	<b>COMBINED DISPLAY/FORM AREA USAGE.....A-6</b>
A.9	<b>HOST/TERMINAL/HOST MESSAGES.....A-6</b>
A.9.1	Command.....A-6
A.9.2	Command Packet.....A-7
A.9.3	Response.....A-8

## CONTENTS (Cont)

		<b>Page</b>
A.9.4	Response Packet.....	A-9
A.9.5	Command Packet.....	A-9
A.9.6	Response Packet.....	A-10
A.9.7	Summary of Commands.....	A-11
A.9.7.1	Definitions.....	A-11
A.9.7.2	Area Manipulation.....	A-11
A.9.7.3	Error Line Manipulation.....	A-12
A.9.7.4	Field Manipulation.....	A-12
A.9.7.5	Input Requests.....	A-12
A.9.7.6	Kills.....	A-12
A.9.7.7	Hard Copy Control.....	A-12
A.9.7.8	Diagnostics.....	A-13
A.9.7.9	Control.....	A-13
A.9.8	Summary of Responses.....	A-13
A.9.9	Common Notations.....	A-13
A.9.10	Description of Commands.....	A-16
A.9.10.1	Definitions.....	A-16
A.9.10.2	Area Manipulation.....	A-19
A.9.10.3	Error Line Manipulation.....	A-22
A.9.10.4	Field Manipulation.....	A-23
A.9.10.5	Input Requests.....	A-26
A.9.10.6	Kills.....	A-31
A.9.10.7	Hard Copy Control.....	A-32
A.9.10.8	Diagnostics.....	A-33
A.9.10.9	Control.....	A-33
A.9.11	Description of Responses.....	A-35
A.9.11.1	Form Responses.....	A-35
A.9.11.2	Menu Responses.....	A-37
A.9.11.3	Control Responses.....	A-39
A.9.11.4	Function Key response.....	A-40
A.9.11.5	Diagnostic Response.....	A-41
A.10	<b>KEYBOARD</b> .....	A-41
A.10.1	Main Keyboard.....	A-41
A.10.2	Keypad.....	A-43
A.10.3	Repeating Keys.....	A-44
A.11	<b>LEDs</b> .....	A-44
A.12	<b>COMMUNICATIONS PROTOCOLS</b> .....	A-45
A.12.1	Logical Communications Protocol.....	A-45
A.12.2	Physical Communications Protocols.....	A-46
A.12.2.1	Asynchronous.....	A-46
A.12.2.2	Synchronous.....	A-46

## CONTENTS (Cont)

	<b>Page</b>
<b>APPENDIX B</b>	<b>INSTALLATION PROCEDURES FOR COMMUNICATIONS OPTIONS</b>
B.1	REMOVING THE COMMUNICATIONS OPTION .....B-1
B.2	INSTALLING THE COMMUNICATIONS OPTION .....B-1

## FIGURES

<b>Figure No.</b>	<b>Title</b>	<b>Page</b>
1-1	VT62 Data Entry and Retrieval Terminal.....	1-2
1-2	VT62 Main Keyboard.....	1-3
1-3	VT62 Auxiliary Keyboard and LED Display.....	1-4
1-4	Screen Layout.....	1-7
1-5	Menu Selections.....	1-8
1-6	VT62 Configurations.....	1-9
1-7	VT62 Switch Setting Bit Configurations (Asynchronous Communication).....	1-14
2-1	Site Considerations.....	2-2
2-2	Terminal, Rear View.....	2-3
2-3	Terminal, Bottom View, Base Front Screen Installed.....	2-4
2-4	Terminal, Bottom View (Partial), Base Front Screen removed.....	2-6
2-5	Terminal, Bottom View (Partial), Base Removed.....	2-7
2-6	20 mA Communications Option Card Installation.....	2-8
2-7	EIA Conversion Board (BN61B) Installation.....	2-9
2-8	Limited Access Operator Control Switches.....	2-10
2-9	Terminal Louver Plate Removal.....	2-12
2-10	VT62 Terminal Interface.....	2-13
2-11	Operator Controls Access Position.....	2-14
2-12	Terminal, Bottom View, Base Front Screen Installed.....	2-14
2-13	Terminal, Rear View.....	2-15
2-14	VT62 Main Keyboard.....	2-17
3-1	Terminal, Rear View.....	3-2
3-2	Terminal, Bottom View.....	3-3
3-3	VT62 Main Keyboard.....	3-4
3-4	VT62 Auxiliary Keypad and LED Display.....	3-8
3-5	Terminal, Operator Control Switch Access Position.....	3-11
3-6	Terminal, Bottom View, Base Front Screen Installed.....	3-11
B-1	Terminal Servicing Position.....	B-2
B-2	Switch Panel Board Connector Removal.....	B-3
B-3	20 mA Communications Option Card Removal.....	B-4
B-4	EIA Communications Option Card (BN61B) Removal.....	B-5

## TABLES

<b>Table No.</b>	<b>Title</b>	<b>Page</b>
1-1	20 mA Signals.....	1-16
2-1	VT62 Identification Tag Data.....	2-3
3-1	Operator Controls.....	3-1
3-2	Alphabetic Key Codes.....	3-5
3-3	Numeric/Symbol Key Codes.....	3-6



# CHAPTER 1 INTRODUCTION

## 1.1 GENERAL DESCRIPTION

The VT62 video display terminal is a 24-line, 80-character display terminal that serves as an input/output (I/O) device to a host processor. This terminal transmits and receives data at speeds up to 9600 baud. The terminal features a main keyboard which uses a standard typewriter layout. This keyboard supplies uppercase and lowercase alphabetic characters as well as numeric, symbol, and control characters to a host. The video display of the terminal uses a 7 × 8 dot matrix to form the displayable characters.

### 1.1.1 VT62 System Description

The VT62 (Figure 1-1) is a buffered CRT terminal designed for data entry and retrieval applications.

Like the VT61 and VT52, it is an uppercase and lowercase ASCII video terminal whose display holds 24 lines of 80 characters.

The VT62 is unique, however, because it is an application terminal for use in the commercial markets. It is specifically designed for commercial data capture and retrieval applications. It is made to be operated by personnel not specially trained in the use of “computers.” The host computer defines the areas on the screen which the operator is allowed to enter, to fill in, or to select. The terminal provides most of the real-time detailed checking of operator responses, but the ability to extend some of the more complicated checking, such as file searches, to the host computer is provided. Operator errors are signaled by an audible tone, an error light, and a short, but descriptive message of what went wrong. Because the host computer loads the message array, it is not necessary to restrict the messages to the English language.

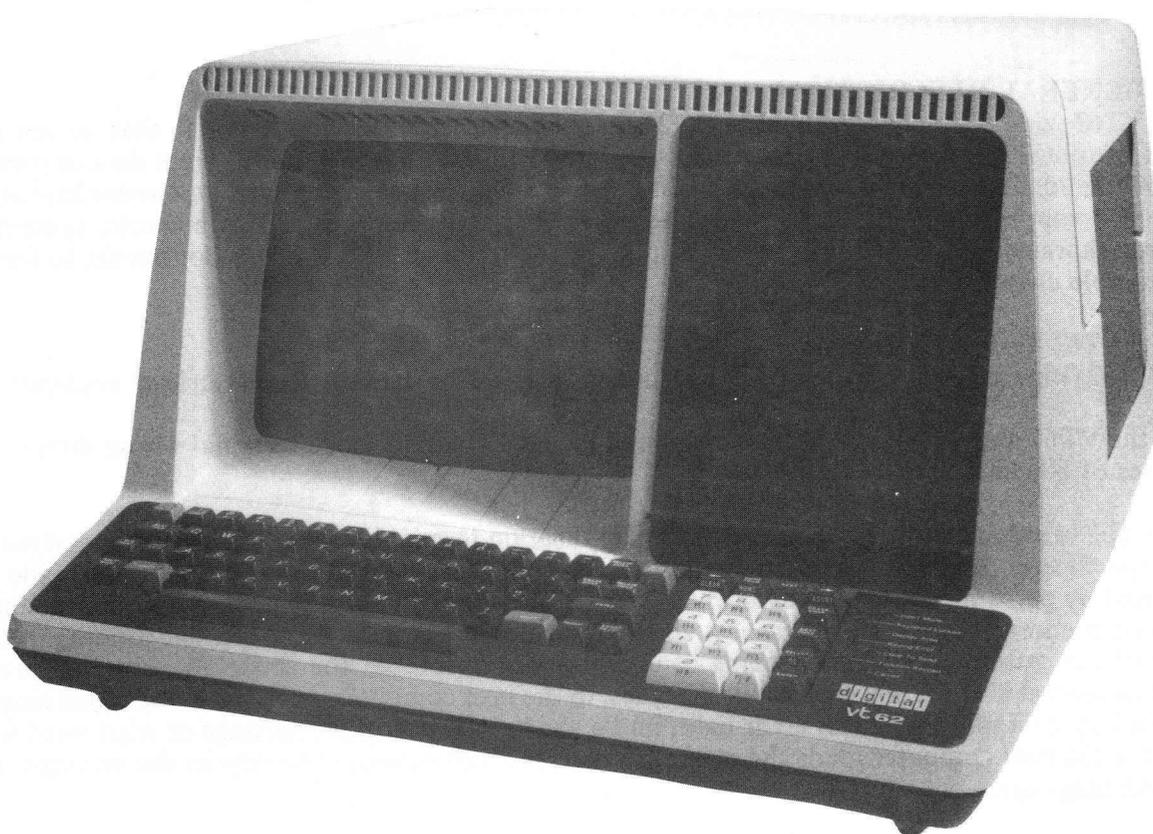
The VT62 uses a standard typewriter keyboard (Figure 1-2) except that the shift lock key is labeled CAPS LOCK, and affects only the alphabetic keys. Operator-oriented features include an adjustable clicking sound which provides feedback as the operator strikes the keys and multiple key rollover to eliminate scrambling of characters, even if several keys are hit simultaneously. The VT62 also has a numeric/special function auxiliary keypad (Figure 1-3).

The 10 numeric keys and the “.” key on the 19-key auxiliary pad serve an additional function as program function keys. The program function is indicated by a label on the front of the key cap and is obtained by striking the key while the SHIFT key is pressed.

The terminal can be placed in a special mode, selectable by the host software, in which the PROGRAM FUNCTION keys operate without the use of the shift key. In this case, the program foregoes the normal function of the numeric keypad.

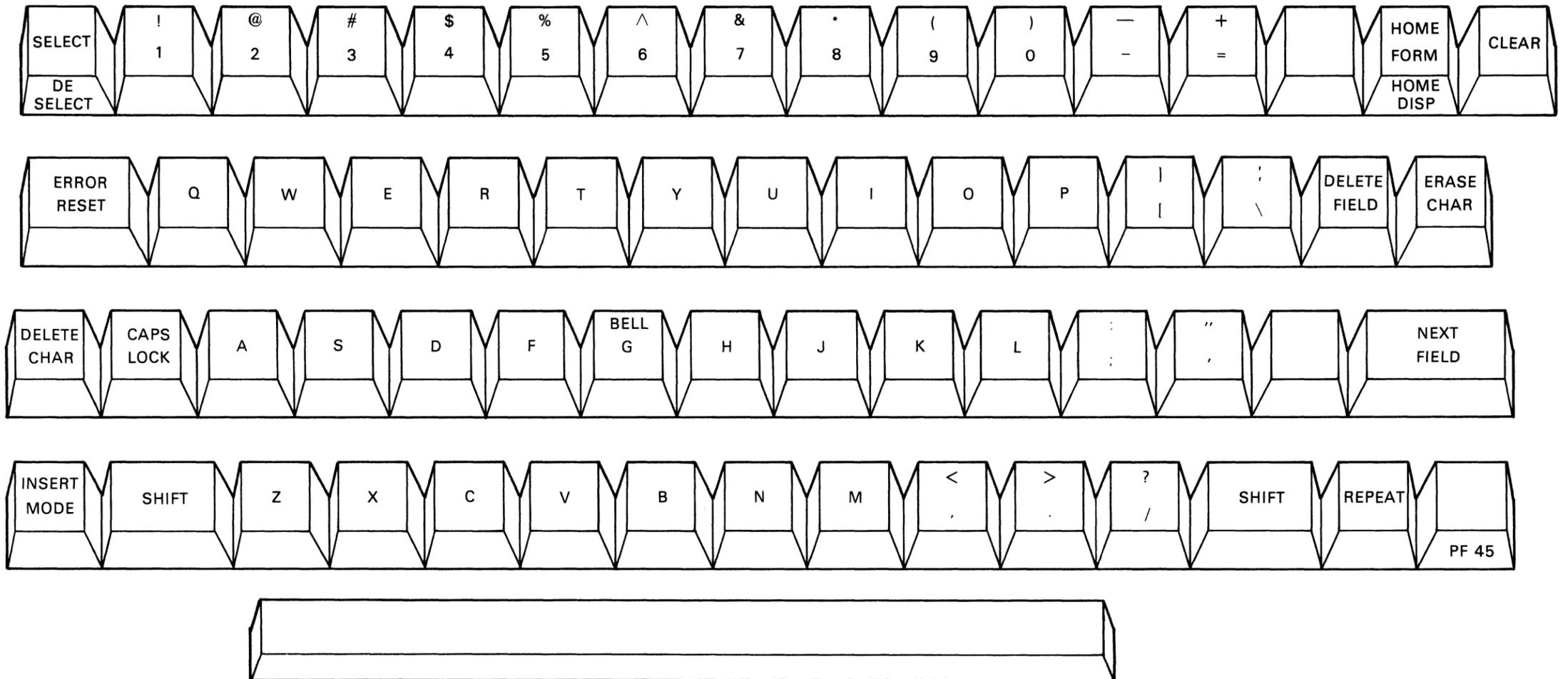
Note that the numeric keypad is functionally self-contained. If all replies in the form are numeric, then the entire form can be entered and sent with one hand which need never move from the pad.

The VT62 provides eight light-emitting diodes (LEDs) (located to the right of the 19-key auxiliary pad) that indicate to the operator when a particular operating mode or condition of the terminal is in effect.



8595-14-A0029

Figure 1-1 VT62 Data Entry and Retrieval Terminal



MA-0930

Figure 1-2 VT62 Main Keyboard

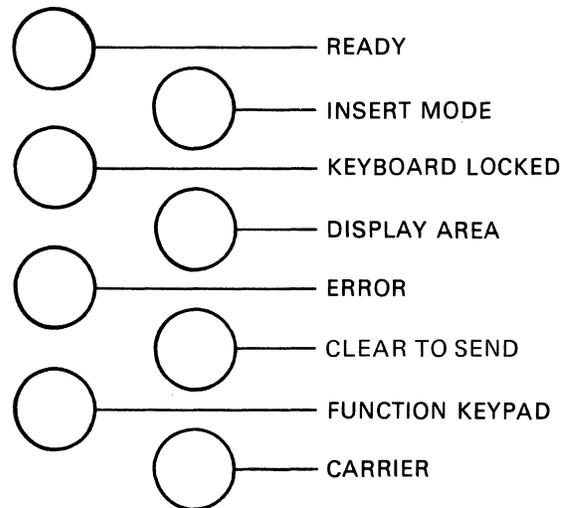
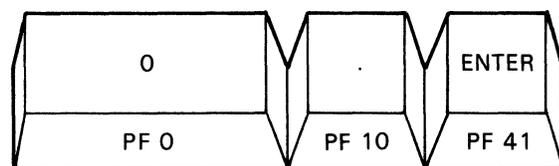
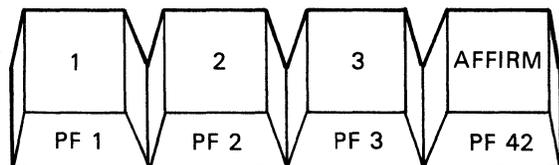
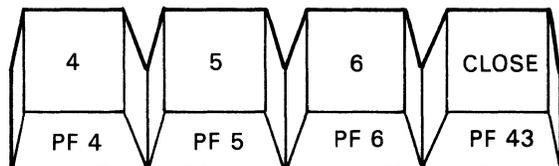
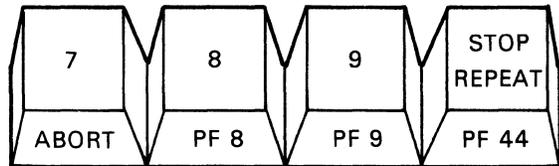
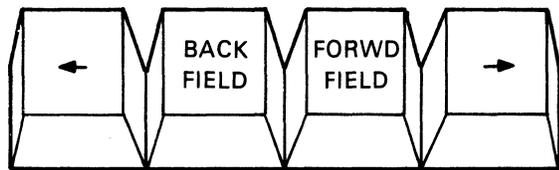


Figure 1-3 VT62 Auxiliary Keyboard and LED Display

MA-0931

## **1.2 FEATURES**

The VT62 contains many special features to enable foolproof operation by relatively unsophisticated users. It also provides reliable, cost-effective communication connections to DIGITAL computer systems for both local and distributed terminal networks.

Important features of the VT62 are:

1. Multipoint DDCMP block-oriented communication
2. Half or full duplex and asynchronous or synchronous communication for application flexibility
3. Formatted split screen display with protected fields, menu selection, invisible password field, and reverse video
4. Local data checking and error notification for many operator errors
5. DEC standard typing keyboard augmented by many special functions for operator convenience
6. High-level software support.

### **1.2.1 Control**

The operation of the VT62 terminal is under the control of the host computer, and not of the terminal operator. However, many data input, editing, and correction functions are at the disposal of the terminal operator. Software in the host or front end computer controls the format of the terminal's display.

For foolproof, protected operation by unsophisticated terminal users, the terminal has no direct access to the operating system of the host machine: the analyst configuring a system with VT62 terminals can grant to the terminal user only those features needed to get the job done.

### **1.2.2 Formatted Screen**

The visible screen of the VT62 is the same as that of the VT61: 24 lines of 80 characters each. The screen displays both uppercase and lowercase characters. The visible cursor alternates between normal and reverse video at approximately 2 Hz.

The VT62 display is "formatted," that is, it consists of protected fields:

1. DISPLAY AREA for operator prompting and information – provided in a rigid format described by the software which the operator is unable to alter from the keyboard, and
2. FORM AREA for data entry – also established by the software with certain syntactic characteristics which are enforced by the terminal.

All text on the screen (except, to some extent, the input fields defined by the host software) is protected from the operator. A program in the host computer defines the fields for operator input by giving tables to the terminal.

### **1.2.3 Input Fields**

The host software's field definitions instruct the system:

1. How to display the field
2. What checks to perform on the data entered in the field (if any)
3. Which format to return the entered data to the application program.

Field definitions supplied by the host software can specify the following attributes:

1. Field number
2. Screen position (line and column)
3. Length
4. Character type, for example:
  - a. Numeric
  - b. Numeric with decimal point (.) and sign (-) allowed
  - c. Alphanumeric with no special characters
  - d. Any displayable characters
5. Field must be entered
6. Field, if entered, must be full length
7. Right/left justified
8. Field may not be altered by operator.

### **1.2.4 Automatic Field Checking**

Character-type restrictions are enforced by the terminal as the operator is entering the content of the field. For example, the terminal does not allow an operator to enter any other characters than 0 through 9 in the field defined as numeric. Some other examples of how the terminal enforces field definitions are:

The "alteration" restriction is checked when the cursor attempts to enter the field: if the field may not be altered, then the cursor moves to the next field (in the current direction, forward or back).

The "non-empty" and "full" restrictions are checked when the operator attempts to move the cursor out of the field: until certain conditions are met, the operator may not go on to the next field. For example, if a field must be entered, the operator must place something non-blank in the field before continuing.

### **1.2.5 Error Messages**

When an operator error is detected by the terminal, one or two messages appropriate to the field type are displayed for the operator on the bottom line of the screen, and the terminal rings the "bell." The error messages are removed from the error line (the bottom line of the screen) when the operator completes the field correctly.

A table of error messages is maintained by the terminal in an "off-line" mode to give guidance to the operator filling in the form. The error messages are loaded into the terminal by the host computer during the initial definition period. These error messages are maintained by the terminal in an "off-line" mode, that is, without intervention of the host computer during the transaction process.

The bottom line of the screen is reserved for the use of the error messages, if they are present. If no messages are sent to the terminal for off-line cueing, then the bottom line of the screen can be used by the program.

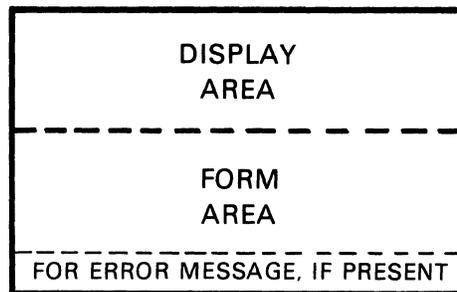
Error messages may be given under the following conditions:

1. Field length exceeded
2. Non-digit in numeric field
3. Illegal character in amount field
4. Special character in alpha field
5. Field entered less than full length
6. Required field not entered
7. Attempt to alter protected text.

### 1.2.6 Program – Controlled Screen Layout

The screen of the VT62 can be divided conceptually into two parts, for the convenience of the operator and the programs. The top portion of the screen is called the DISPLAY AREA, and the bottom the FORM AREA (Figure 1-4).

The dividing line between the two areas is conceptual only: there is no visible indication. The location of the “division” is under program control. A program can choose to devote from 0 to 24 lines of the screen to the display area, leaving the remainder of the screen for the form area.



MA-0925

Figure 1-4 Screen Layout

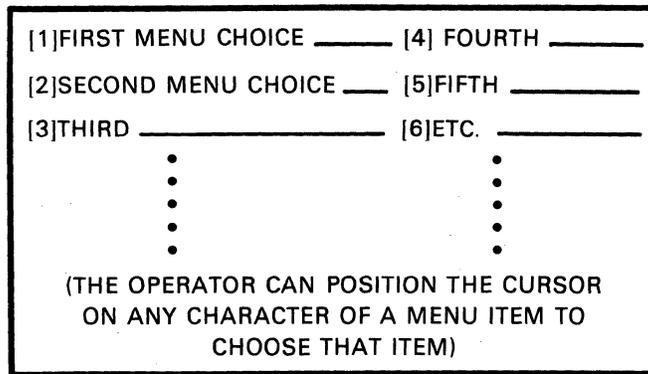
**1.2.6.1 Display Area** – The display area is intended for two uses: to supply dynamic information to the operator, and to allow the operator to choose easily from structured menus.

The display area can be cleared, filled, altered, or changed in size under program control. Operators may not alter its contents in any way, but they can move the visible cursor into the area to select an item from a menu displayed there.

**1.2.6.2 Display Area Menu Selection** – Within the display area of the screen, the operator’s cursor moves freely throughout the text presented there.

To choose a menu item (Figure 1-5), the operator merely positions the cursor anywhere within the text of the item, then presses either the ENTER key or one of the PROGRAM FUNCTION keys.

If the operator’s cursor is within the display area of the screen when the ENTER key or one of the PROGRAM FUNCTION keys is pressed, then the terminal sends the cursor position to the host as an indication of menu item choice.



MA-0926

Figure 1-5 Menu Selections

**1.2.6.3 Dynamic Displays** – The display area can also be used to supply the operator with dynamic information of interest, or to provide responses to previous operator input.

The VT62 can receive unsolicited messages from the host machine at any time, without disturbing operator keyboard activity. Thus, the display area can be updated at any time by the host, to provide current reports in static format; or it can be scrolled upward (independent of activity in the form area), to provide running information.

Examples of these uses are:

1. A fixed-format display of a stock portfolio, with values computed on most recent market transactions.
2. A running stock ticker, giving all market activity, one or several transactions per line, with old transactions scrolling off the screen.

To respond to previous operator input, a program can clear the entire screen, establish the size of the display area needed, and send the reply followed by the next form.

**1.2.6.4 Form Area** – The form area is used to hold a protected form and fields for operator data entry. Text layout in the form and fields within the form is arbitrary, as determined by the program.

The order of processing of fields is determined by the host software. For example, fields can be defined so that they appear to be ordered in columns, top to bottom, rather than across the screen, left to right.

Within the form area, the operator can move his cursor from field to field, and among the characters within a field. The cursor cannot be placed on protected text inside the form area. The repeating FORWARD FIELD and BACK FIELD keys are provided for rapid access to fields. The repeating cursor keys are provided for rapid positioning within a selected field.

Keyboard text entry is always done in overstrike mode: rekeying is generally faster and more accurate than editing the contents of individual fields. However, special keys are provided to insert or delete characters from any location with a defined field.

### 1.2.7 System/Terminal Configurations

The VT62 can be connected to a system in multipoint and single-point configurations using a multiplexer (Figure 1-6). Local single-point connections can be accomplished without modems.

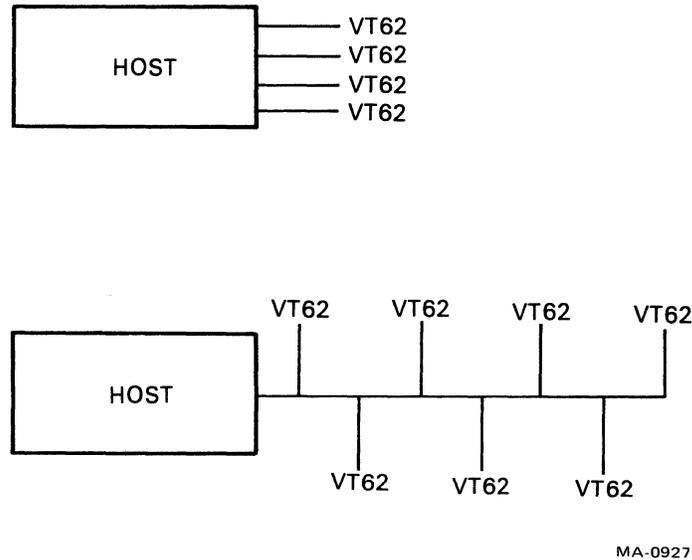


Figure 1-6 VT62 Configurations

In either case, the VT62 acts as a DDCMP multipoint tributary (multidrop slave, circumspect mode), transmitting only when polled by the host or front end computer. In multipoint configurations, therefore, the host computer has complete control over all the terminals connected by multidrop lines – and many terminals can be serviced concurrently. In single-point configurations, the host computer has complete control over the terminal's demands on the system's resources.

All communication between the terminal and the computer is in blocks of data containing both commands and text.

The VT62 may be configured three ways:

1. EIA synchronous, up to 4800 baud: for use with a synchronous modem as a stand-alone multidrop terminal.
2. EIA asynchronous, up to 9600 baud: for direct radial (single-drop) connection to a host or concentrator, as a high-speed local terminal.
3. 20 mA asynchronous, up to 9600 baud: for direct radial (single-drop) connection to a host or concentrator, as a high-speed local terminal (with the optional 20 mA Communications Interface Card).

### **1.2.8 Power-Up State**

When powered up in the ON-LINE mode, the terminal immediately enters Short Self-Test. At the end of the Self-Test (approximately ten seconds), when polled, the terminal sends a DDCMP START or START ACKNOWLEDGE message to the host. The host software can then take action to set up the terminal for the operator. It is possible, in case of temporary loss of power, that the host may have to take special action to continue transactions that were in process.

To the operator, the terminal screen is blank, and the keyboard is locked.

### **1.2.9 Data Transmission**

If the cursor is within the form area of the screen when the ENTER key is pressed, the contents of all fields entered by the user is sent to the host.

Transmission of data from the input fields can be in packed or unpacked format, at the discretion of the software. In packed format, only fields entered or modified by the terminal operator are transmitted, and these fields are null-compressed. In unpacked format, all characters of all fields are transmitted, with no data compression, and no field separators.

### **1.2.10 Printer Option**

On command from the host, the terminal will send information to the printer interface. Three types of printing are possible:

1. Dump the display portion of the screen
2. Dump one specific line from the screen
3. Send text directly to the printer without altering the content of the screen.

No additional formatting is implied. The screen is not cleared or altered.

## **1.3 SPECIFICATIONS**

### **1.3.1 VT62 Overall Specifications**

#### **Dimensions**

Height	360 mm (14.1 in)
Width	530 mm (20.9 in)
Depth	690 mm (27.2 in)
Minimum Table Depth	450 mm (17.2 in)

Weight 18.86 kg (44 lb)

Plastic Case Material Injection-molded Noryl thermoplastic

Operating Environment	DEC Standard 102 Class B environment
Temperature	10° C to 40° C (50° F to 104° F) 10% to 90% relative humidity with: Maximum wet bulb: 28° C (82° F) Minimum dew point: 2° C (36° F)
Line Voltage	90–127 V (115 V nominal) 180–254 V (230 V nominal) (115/230 V nominal is switch selectable)
Line Frequency	60 Hz $\pm$ 1 Hz; 50 Hz $\pm$ 1 Hz, switch selectable
Power Line Filter	RFI filter
Leakage Current	Less than 3.5 mA at 250 V 50 Hz
Ride-Through Capability	Provides a minimum of 22 ms ride-through power during a power outage condition from low line as specified (90 V to 180 V)
Overload Protection	Fast blow fuse
Power Consumption	93 W (318 btu hours)
Internal Power Supply	Overcurrent protected +5 V $\pm$ 5% at 8.5 A, maximum +12 V $\pm$ 5% at 1.605 A, maximum -12 V $\pm$ 5% at 0.6 A, maximum +90, 140 V at 0.2, 0.03 A (not used)
UL Approval	Yes
Internal Timing Source	Crystal oscillator 15.360 MHz $\pm$ 0.01%
Monitor Electronics	100% solid state
Phosphor	P4, unilluminated color may vary
Geometric Distortion	The parameter of a full field of characters will approach an ideal rectangle to within $\pm$ 4% of the rectangle height.
Video	Raster scan: roll free: no need for horizontal hold and vertical hold adjustments

## TV Picture Quality

The character resolution will not vary markedly across the screen nor will there be excessive pincushion nor excessive barrel distortion. What is acceptable will be that which is acceptable to the untrained observer such as clerk-typist. The picture must be clear, easy to read, and free of visual defects that impair readability.

The horizontal centering will be such that the distance of the left-hand character on the 12th line from the left screen boundary is within 13 mm (1/2 inch) of the distance from the rightmost character on the 12th line to the right screen boundary.

The vertical centering will be such that the distance of the top screen edge to the 40th character on line 1 is within 13 mm (1/2 inch) of the distance of the bottom screen edge to the 40th character on the 24th line.

The screen image will appear parallel to the table surface.

### NOTE

**VT62 terminals use unburned-in cathode-ray tubes and after a period of use will show dark areas across the face of the CRT where characters are displayed. The selective darkening of the CRT in no way impairs proper operation nor shortens the life of the CRT.**

Display Format  
Character Type  
Character Size  
Screen Size  
Active Screen Size

24 lines of 80 characters  
7 × 8 dot matrix  
2.3 ± 0.4 × 3.5 ± 0.4 mm  
205 mm (12 in) diagonal rectangular  
218 × 114 mm (8.6 × 4.5 in)

Character Set

96 characters uppercase and lowercase ASCII, plus 32 special graphic characters

Key Layout

Typewriter rather than keypunch format; main keypad (63-key array); alternate keypad (19-key array)

Keyclick

Audible sound simulates typewriter feel

Operator Controls  
(Direct Access)

Power ON/OFF  
Intensity Control  
BAUD RATE switches  
ON-LINE/OFF-LINE switch

Display control character switch  
Keyclick volume control

**Internal Setup Controls  
(Limited Access)**

**PARITY/NO PARITY  
EVEN/ODD PARITY  
MAINTENANCE MODE  
HALF DUPLEX/FULL DUPLEX  
7 bits/8 bits (ASYNC MODE only)  
60/50 Hz (line frequency matching for correct screen size)  
115/230 V Switch  
Vertical Height  
Vertical Linearity  
Horizontal Size  
Focus  
Reverse Video Intensity**

**Operator Signals**

There are 8 light-emitting diodes to the right of the 19-key array on the keyboard labeled as follows (from top to bottom):

**READY  
INSERT MODE  
KEYBOARD LOCKED  
DISPLAY AREA  
KEYING ERROR  
CLEAR TO SEND  
FUNCTION KEYPAD  
CARRIER**

**Bell**

An audible alarm sounds when CTRL G (007) is received and in other situations as described in the VT62 Functional Specification (Appendix F).

**Cursor Type**

**Blinking reverse-video**

**Communications**

**Synchronous  
Asynchronous  
Transmission Code**

**Choice of 20 mA current loop, EIA (with control)  
See Dec Standard 121**

**8-bit ASCII (U.S.) extended through escape sequences**

**Transmission Code Length**

**10 bit (asynchronous)  
11 bit**

**75, 150, 300, 600, 1200, 1800, 2400, 4800, 9600 baud  
110 baud only**

**Transmission Character  
Bit Configuration**

The VT62 has an internal set of switches that are used to select parity or no parity, even or odd parity if parity is enabled, and the number of data bits per character (7 bits or 8 bits). The bit configuration of each switch setting is shown in Figure 1-7.

WITH NO PARITY SELECTED (PARITY SWITCH OFF)

START	DATA 1 LSB	2	3	4	5	6	7	8* MSB	IDLE OR STOP – NEXT CHARACTER
-------	------------------	---	---	---	---	---	---	-----------	-------------------------------------

WITH PARITY (ODD OR EVEN; PARITY SWITCH ON)

START	DATA 1 LSB	2	3	4	5	6	7	8*	PARITY	IDLE OR STOP-NEXT CHARACTER
-------	------------------	---	---	---	---	---	---	----	--------	-----------------------------------

\*TO SUPPORT DDCMP OPERATION THE 7/8 BIT (LAS 3) SWITCH MUST BE SET TO 8 AND THE PARITY/NO PARITY (LAS 1) SWITCH MUST BE SET TO NO PARITY.

MA-0933

Figure 1-7 VT62 Switch Setting Bit Configurations (Asynchronous Communication)

Speed

Switch selectable; switch settings:

- Receive at 600 baud
- Receive at 1200 baud
- Receive at 1800 baud
- Receive at 2400 baud
- Receive at 4800 baud
- Receive at 9600 baud

Receive speed matches transmitter speed.

- Transmit at 100 baud
- Transmit at 75 baud
- Transmit at 150 baud
- Transmit at 300 baud
- Transmit at 600 baud

Transmitter speed matches receiver speed.

Parity

Generated on transmission as even or odd or a mark (switch selectable) required on received data as a function of switch settings.

**1.3.2 20 mA Current Loop Specifications**

Transmitter

Passive, isolated goes to the mark state when power is turned off

Open Circuit Voltage  
(of the current being driven)

5.0 V, minimum  
50 V, maximum

Voltage Drop Marking	0.5 V, minimum 4.0 V, maximum
Spacing Current	0.4 mA, minimum 2.0 mA, maximum
Marking Current	20 mA, minimum 50 mA, maximum
Spacing Current	0.0 mA, minimum 3.0 mA, maximum
Marking Current	15 mA, minimum 50 mA, maximum

### 1.3.2.1 Baud Rates vs. Distance for Integral 20 mA Loop

- Greater than 60.96 m (200 ft) at any baud rate when connected to an M596
- Greater than 304.8 m (1000 ft) at any baud rate when connected to an M5960.

**1.3.2.2 20 mA Adapter Card with Cable (BN52B) Specifications** – The 20 mA adapter card is supplied with a cable that ends with an 8-pin Mate-N-Lok connector for connection to DEC computers. The plug may be cut off and changed as desired. Table 1-1 describes the 20 mA signals.

Used On	VT52, VT61, VT62
Card Size	5 × 15.2 cm (1.97 × 5.98 in)
Cable Length	4 m (13.12 ft)
Cable Type	Four wires No. 22 AWG standard
Connector Housing Type	DEC Part No. 12-09340-01 AMP 1-480460
Connector Pin Types (four required)	DEC Part No. 12-09378-03 AMP 350079-4
Connector Mates With	DEC Part No. 12-09340-00 AMP 1-480459
Female Pins (four required)	DEC Part No. 12-09379-03 AMP 350078-4

**Table 1-1 20 mA Signals**

<b>Signals from 20 mA Adapter Card</b>	<b>Description</b>	<b>Computer Signal</b>
Transmit Positive	Wire color = green Mate-N-Lok pin = No. 7 Connects to	Receive Positive
Transmit Negative	Wire color = red Mate-N-Lok pin = No. 3 Connects to	Receive Negative
Receive Positive	Wire color = white Mate-N-Lok pin = No. 5 Connects to	Transmit Positive
Receive Negative	Wire color = black Mate-N-Lok pin = No. 2 Connects to	Transmit Negative

**1.3.3 EIA (With Control) Card Specifications (BN61B)**

The EIA card is supplied with a VT62, and consists of a cable and a card containing five chips that convert to and from EIA levels.

Length of Cable	5 m (16.4 ft)
Connector	Cannon DB 19604-432 or equivalent Male 25-position connector
EIA Pin No. 1	Signal: Protective ground Comment: Logic 1 = Off = -10 V
EIA Pin No. 2	Signal: Transmitted data Comment: Logic 0 = On = +10 V
EIA Pin No. 3	Signal: Received data Comment: Logic 1 = Off = -5 to 25 V Comment: Logic 0 = On = +5 to 25 V
EIA Pin No. 4	Signal: Request to send Comment: Wired true (+10 V)
EIA Pin No. 7	Signal ground
EIA Pin No. 8	Signal: EIA carrier detect
EIA Pin No. 14	Signal: EIA secondary transmitted data
EIA Pin No. 15	Signal: EIA serial clock transmit

EIA Pin No. 16	Signal: EIA secondary received data
EIA Pin No. 17	Signal: EIA serial clock receive
EIA Pin No. 20	Signal: Data terminal ready Comment: Wired true (+10 V)

The VT62 meets RS-232C by using the Motorola MC 1488 RS-232C line driver and MC 1489 RS0232C line receiver.

Appendix A contains the functional specification for the VT62.

#### **1.4 Documentation Ordering Information**

The following VT62 hardware manuals are available from DIGITAL's Accessory and Supplies Group.

<b>Title</b>	<b>Index No.</b>
VT62 DECscope User's Guide	EK-VT62-UG-001
VT62 DECscope Technical Manual	EK-VT62-TM-001

All purchase orders and inquiries for Hardware Documentation should be forwarded to:

Digital Equipment Corporation  
Accessory and Supplies Group (PO86)  
Cotton Road, Nashua, N.H. 03060



## **CHAPTER 2 INSTALLATION**

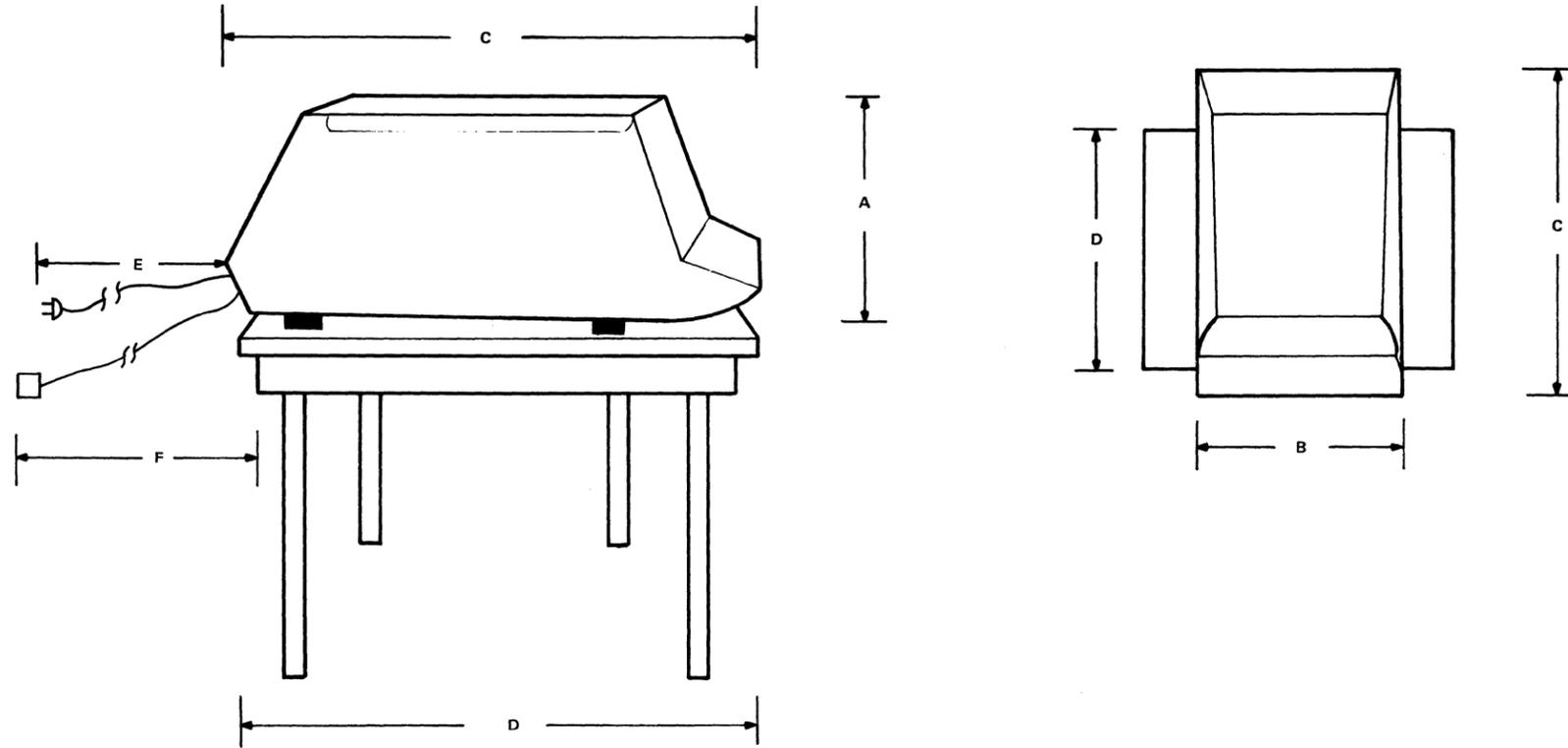
### **2.1 SITE PLANNING**

The design of the VT62 will normally pose few constraints on the selection of a place to install the terminal (Figure 2-1). In most cases, any environment which is suitable to the operator of the terminal will be a satisfactory environment in which to operate the terminal. Extremes of temperature and humidity should be avoided. A summary of the guaranteed operating conditions of the VT62 is found in Chapter 1.

### **2.2 UNPACKING AND INSPECTION**

1. Unpack the terminal and place it on the desk or the surface where it will be operated.
2. Inspect the terminal for physical damage.
3. Make certain that site power meets the requirements of the terminal. Information concerning the electrical requirements of the terminal and the communications interface supplied with the unit is provided by the identification tag located on the rear of the terminal (Figure 2-2). This tag also contains the serial number and the model number of the terminal. Use Table 2-1 to translate the identification tag data into usable information.

All VT62 models come equipped with the parallel interface for the printer (LA180 or equivalent). However, the printer is supplied separately and is not specified by the model number of the VT62.

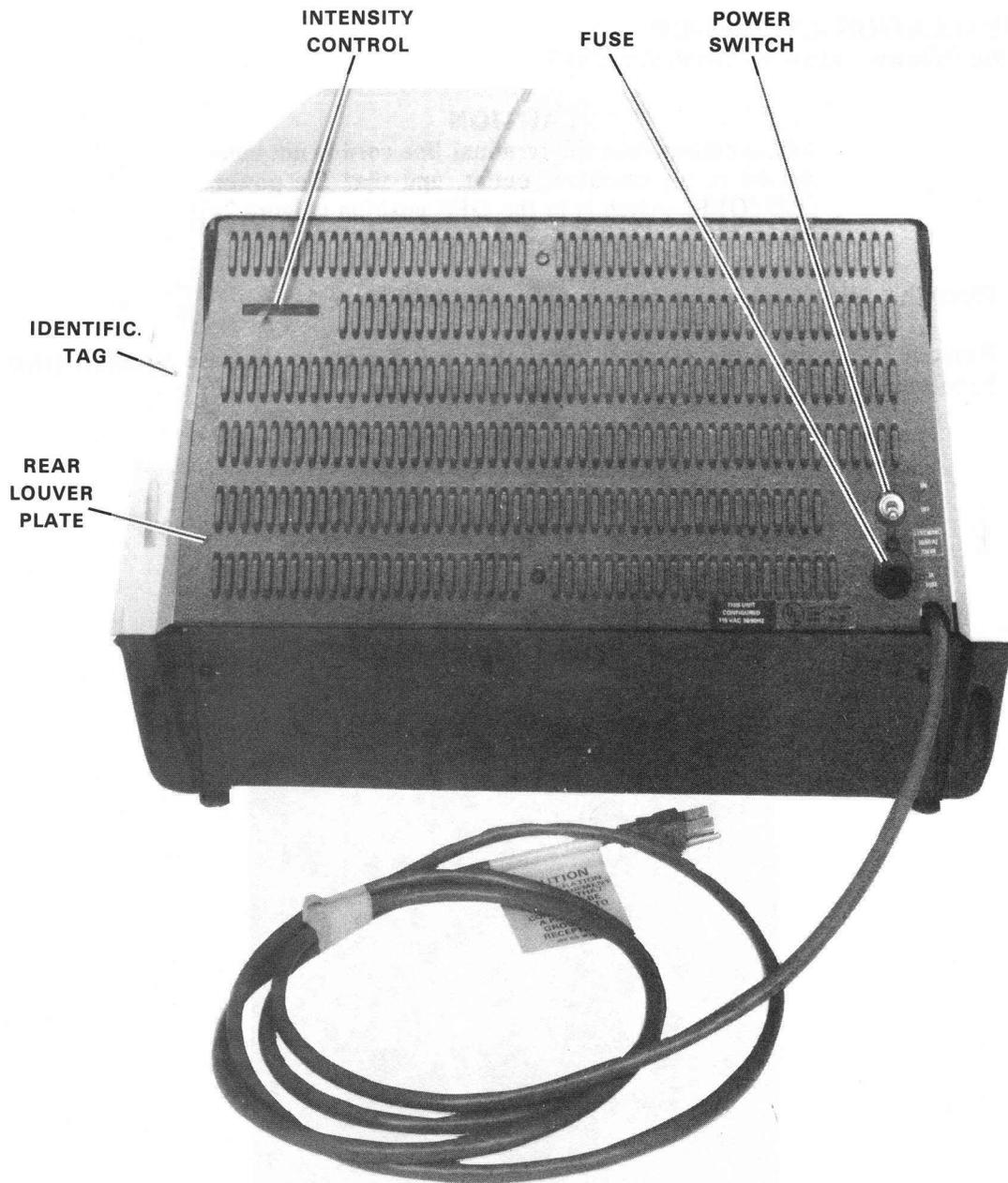


DIMENSIONS	MILLIMETERS	INCHES
A. Height	360	14.1
B. Width	530	20.9
C. Depth	690	27.2
D. Minimum Table Depth	450	17.2
E. AC Line Cord Length	2438	96.0
F. Interface Cable	*	*

\*Total cable length (including the 7,620 meter (25 foot) cable supplied) must not exceed 304.8 meters (1000 feet).

CP-2819

Figure 2-1 Site Considerations



8595-5-A0005

Figure 2-2 Terminal, Rear View

Table 2-1 VT62 Identification Tag Data

Communications Option	115 V, 60 Hz	230 V, 50 Hz
20 mA loop, Mate-N-Lok EIA with control	VT62-AA VT62-AC	VT62-AB VT62-AD

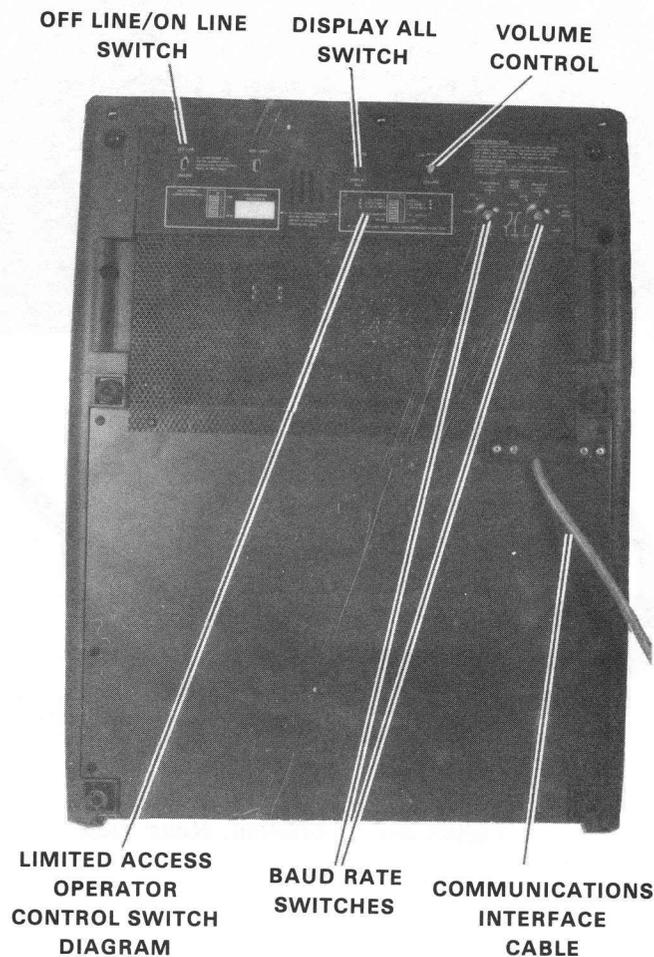
## 2.3 INSTALLATION AND SET-UP

Perform the following steps to install the VT62:

### CAUTION

Make certain that the terminal line cord is not connected to an electrical outlet, and that the power (ON/OFF) switch is in the OFF position (Figure 2-2).

1. Place the unit upside down on a bench or table (Figure 2-3).
2. Remove the upper cable clamp securing the option cable to the base assembly (two Phillips head screws 8-32 X .38) using a Phillips screwdriver.



8595-9-A0003

Figure 2-3 Terminal, Bottom View, Base Front Screen Installed

3. The forward portion of the terminal base (the base front screen) can be removed to gain access to part of the terminal's electronics.

This allows the operator to:

- a. Remove the communications interface card and replace it with another communications option.
- b. Access the Terminal Address Switch and/or the HALF/FULL DUPLEX, PARITY ON/OFF, PARITY EVEN/ODD, 7/8 DATA BIT, and 50/60 Hz switches located inside the unit.
- c. Plug in the cable to the printer.

The base front screen is held down at the rear by five hex-head screws and at the front by a lip in the shell material. By unscrewing the hex-head screws, this section of the base can be lifted out. It is not necessary to remove the entire base.

Use care when removing this section of the base, as the switches and speaker which are attached to the screen will still be connected to the internal electronic by a cable (Figure 2-4).

4. Remove the switch panel board connector P1 from J4 on the I/O board (Figure 2-4).
5. A 40-pin male Berg connector is visible at the right in Figure 2-4. This is the connector for the general output register (printer). The cable from the LA180 or equivalent printer should be attached to this plug so that the lettering on the cable faces you and can be read.
6. There are two 11-pin connector rows in the center of the printed circuit board (Figure 2-5). The communications interface card (20 mA or EIA) is plugged into one or both of these connectors depending on its size. A single-width card that has only one connector on its underside (i.e., the 20 mA card) will plug into the connector closest to the front of the unit (Figure 2-6). The BN61B EIA connector, which provides control as well as data signals, will use both rows of pins (Figures 2-7 and 2-5). Appendix B contains complete communications interface installation instructions.
7. The Limited Access Operator Control Switches (Figure 2-8), are also found under the base screen. The switches are shown by a diagram mounted on the portion of the base screen that was removed (Figure 2-3).

Make certain that Limited Access Switch 4 (LAS 4) is in the ON position.

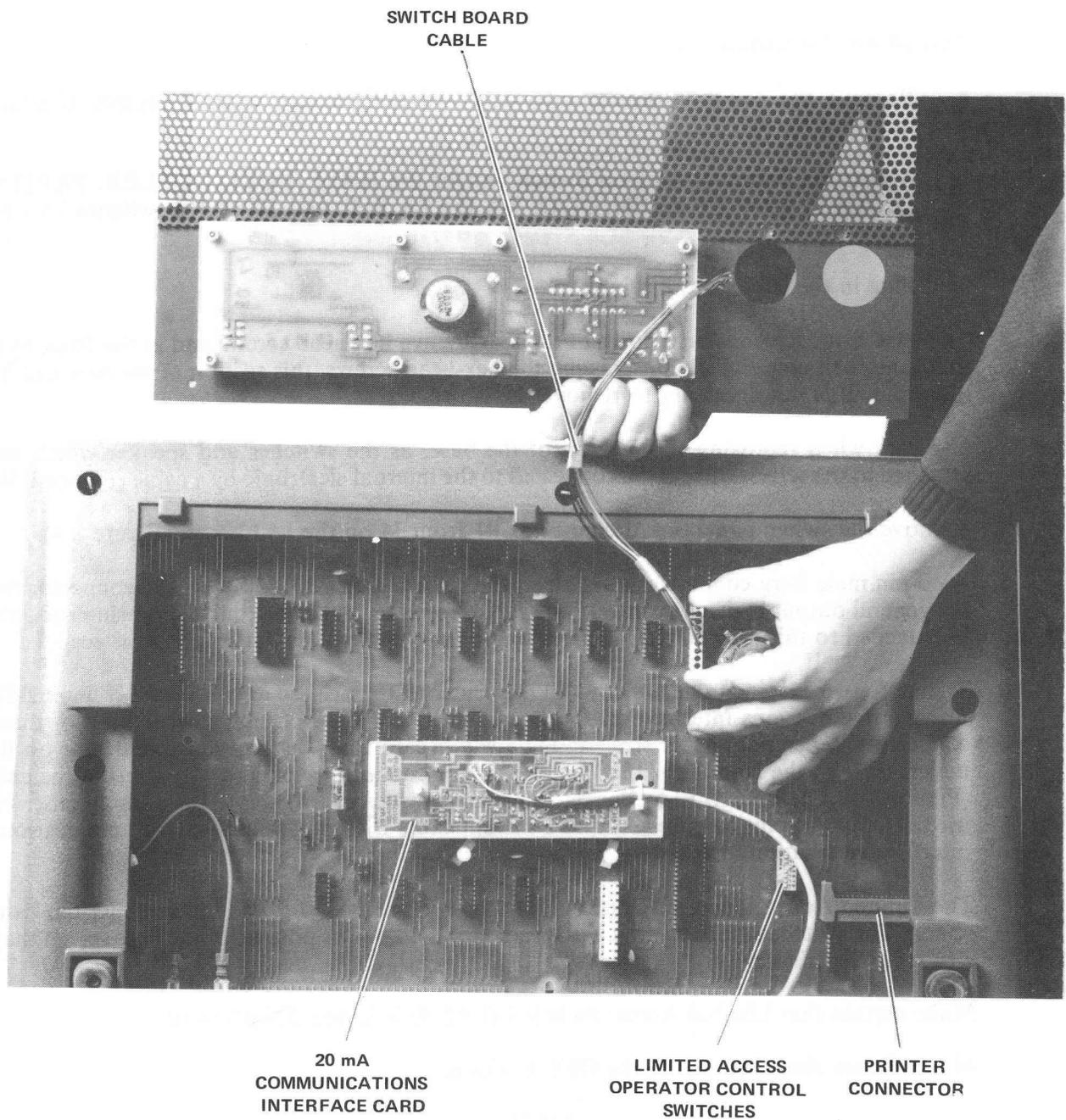
Make certain that LAS 7 is in the OFF position.

#### NOTE

**The VT62 can be set up with Limited Access Control Switch 7 in the ON position. This causes the VT62 to always remain ON-LINE, regardless of the setting of the ON-LINE/OFF-LINE Control Switch.**

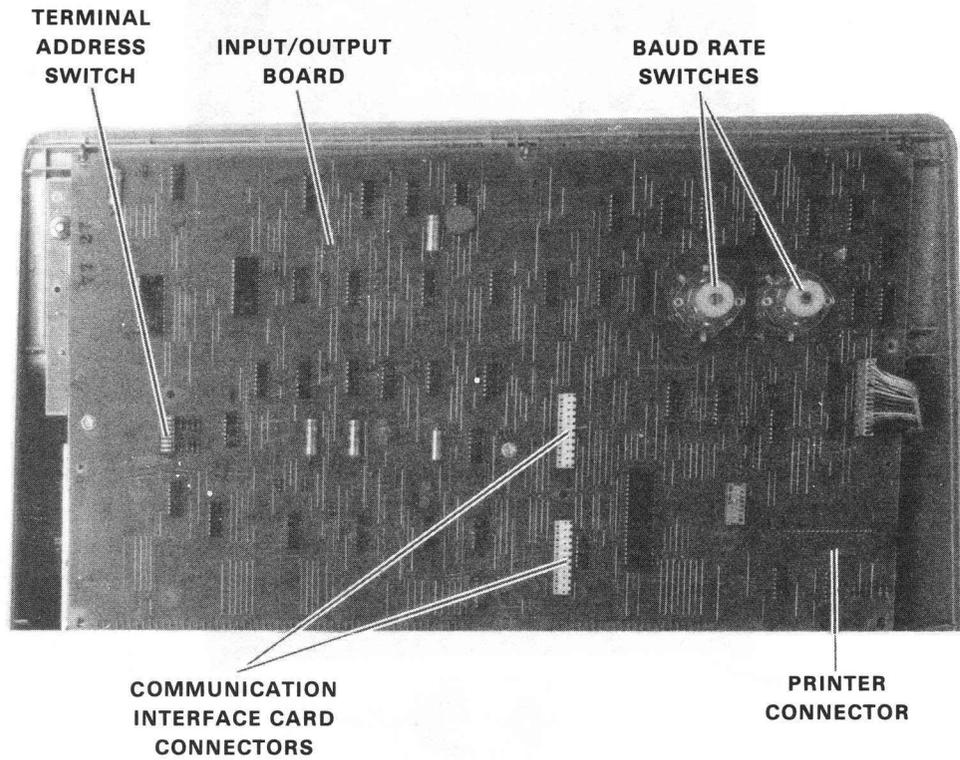
Set LAS 8 (the 50/60 Hz switch) to match the line frequency (within 1 Hz) to ensure that the video display is easy to read.

At this time Parity (LAS 1), Even/Odd Parity (LAS 2), 7/8 Bits (LAS 3), and Half/Full Duplex (LAS 6) selections should be made.



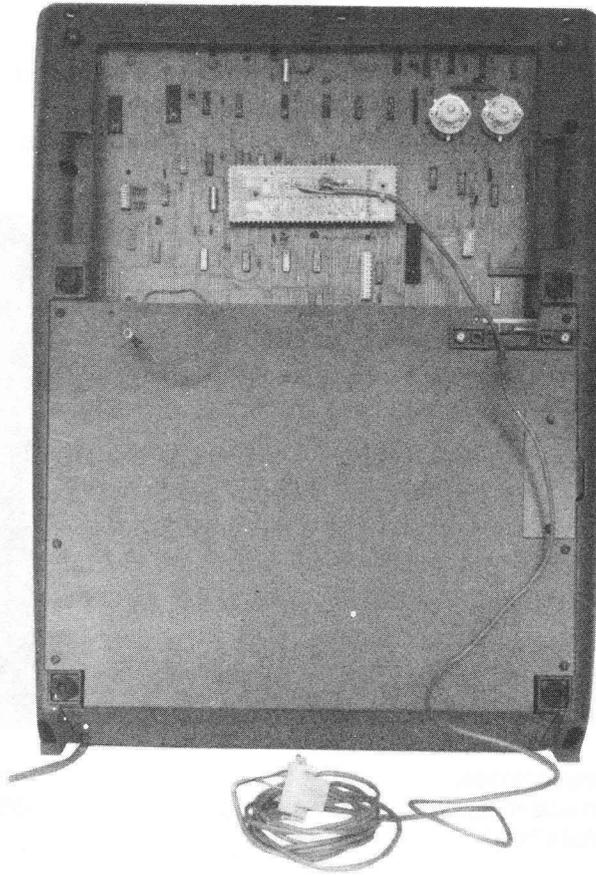
8284-14-A0007

Figure 2-4 Terminal, Bottom View (Partial), Base Front Screen Removed



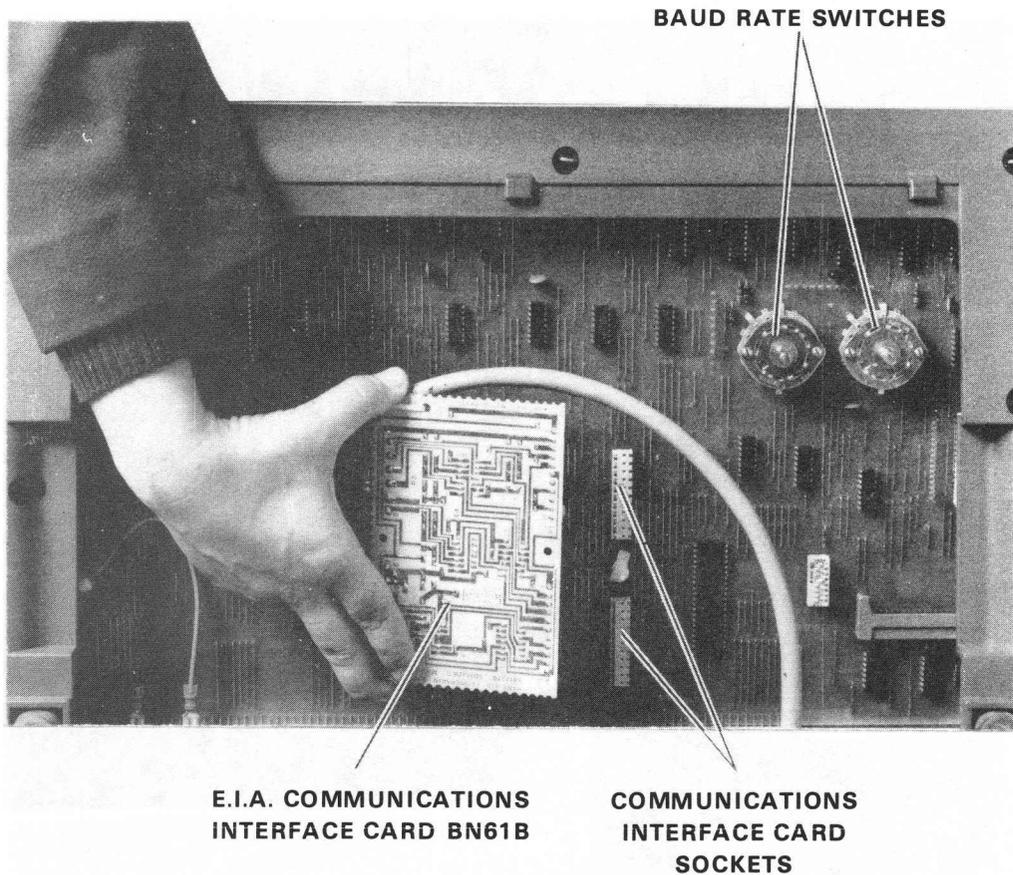
8595-4-A0015

Figure 2-5 Terminal, Bottom View (Partial), Base Removed



8595-15-A0030

Figure 2-6 20 mA Communications Option Card Installation

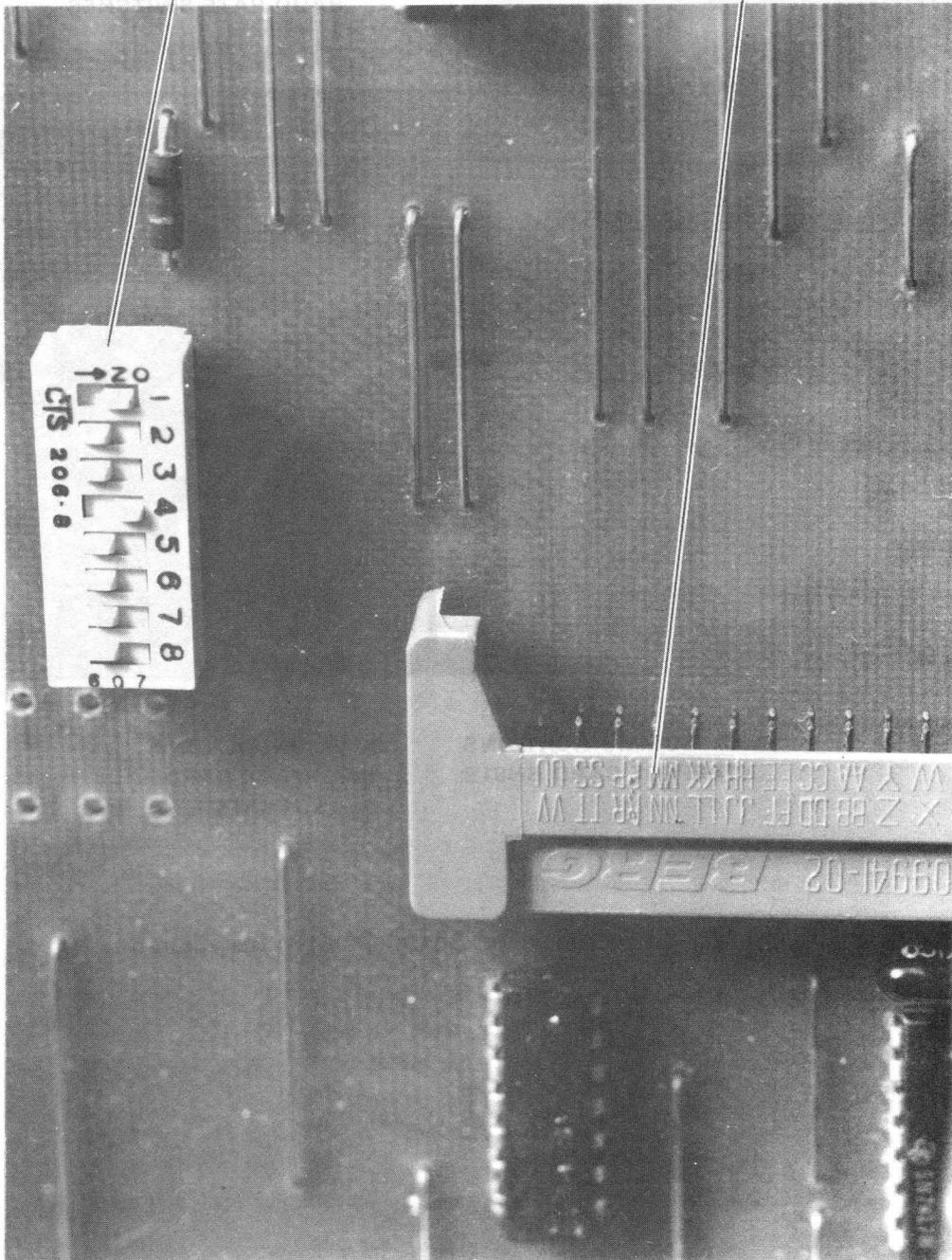


8284-12-A0026

Figure 2-7 EIA Conversion Board (BN61B) Installation

LIMITED ACCESS  
OPERATOR CONTROL SWITCHES

PRINTER  
CONNECTOR



8284-25-A0031

Figure 2-8 Limited Access Operator Control Switches

8. If a terminal address is required, set the Terminal Address Switch at this time (Figure 2-5).

There is an adjustment by which the user can adapt the VT62 to the higher voltage ranges (180–254 V). This switch is found on the power supply module (Figure 2-9) and is clearly marked “115” (for the 90–127 V range) and “230” (for the 180–254 V range). To gain access to the power supply, the rear louver plate must be removed. In addition to the steps covered so far, this will involve removing the screws in the rear louver plate with a 1/4 inch nut driver and removing the louver plate. Reverse these steps before applying power.

9. Replace the switch panel board connector P1 to J4 on the I/O board.

#### **CAUTION**

**Make certain that P1 is replaced correctly on J4, with the black wire of the cable toward the front of the unit. P1 incorrectly replaced on J4, will damage the I/O board upon power-up, necessitating its replacement.**

10. Replace the base front screen by reversing Step 3.
11. Secure the option cable to the base assembly by reversing Step 2.
12. Figure 2-3 shows another cable coming from the unit. This cable will have a Mate-N-Lok connector or an EIA connector, depending on which option you have ordered. Plug this connector into the I/O part of the computer or modem with which you wish to communicate.
13. Make certain that the rear louver plate is back in place and then return the VT62 to its operating position.
14. Plug the 3-pronged line cord of the unit into an electrical outlet that meets the requirements of the terminal.
15. The power (ON/OFF) switch is at the rear of the unit (Figure 2-2). Make sure it is in the upper (ON) position.

#### **NOTE**

**Run the interface cabling to the host according to the appropriate cable configuration diagram located in Figure 2-10.**

## **2.4 ACCEPTANCE TEST PROCEDURES**

### **2.4.1 VT62 OFF-LINE ACCEPTANCE TESTS**

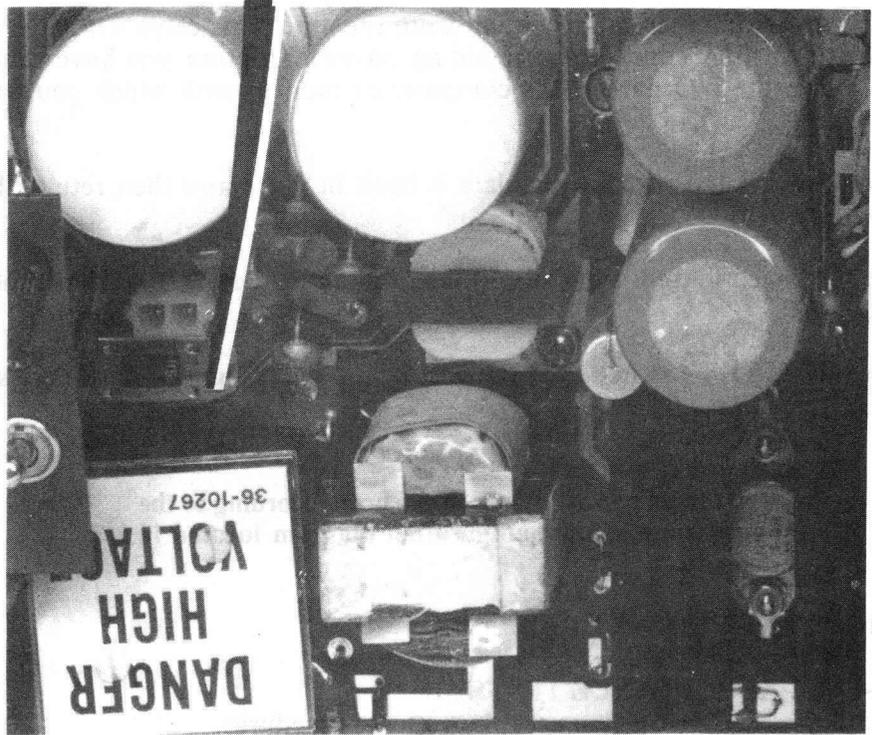
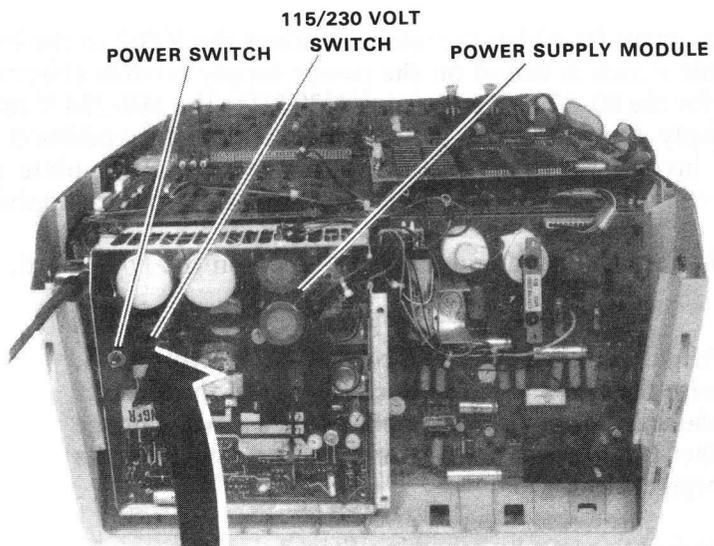
The following paragraphs list the off-line acceptance test procedures.

#### **2.4.1.1 Set-Up Procedure**

#### **NOTE**

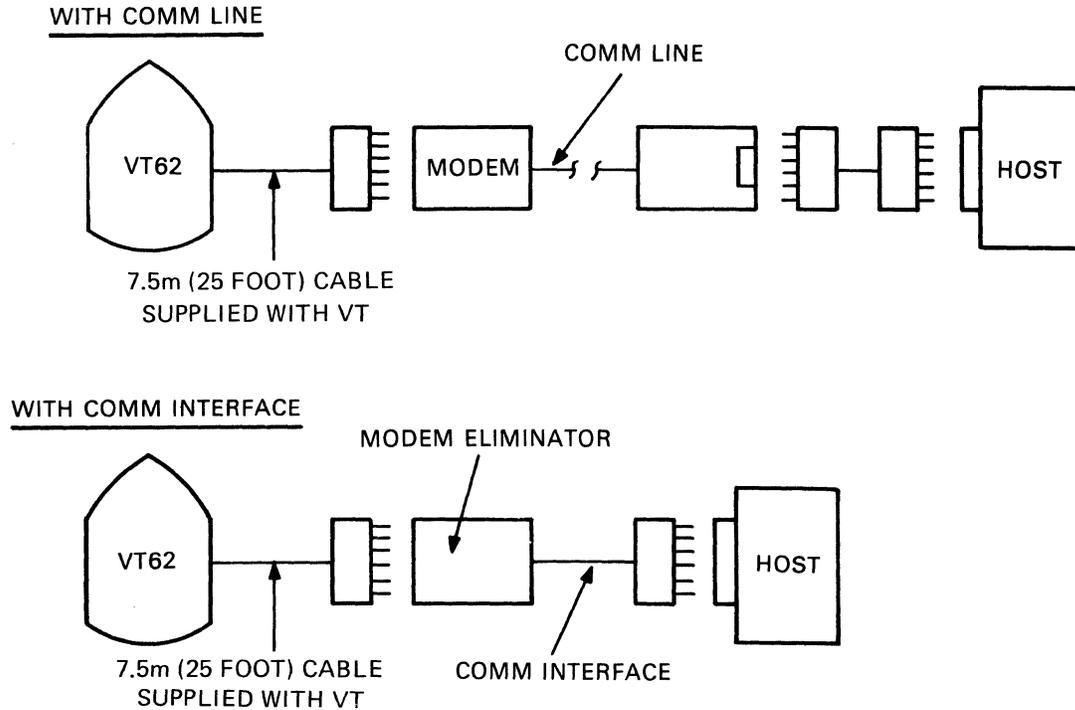
**Perform the initial set-up procedure in Paragraph 2.3 before continuing with this procedure.**

1. Make certain that the terminal line cord is not connected to an electrical outlet and that the power (ON/OFF) switch is in the OFF position.



8595-11-A0020

Figure 2-9 Terminal Louver Plate Removal



MA-0928

Figure 2-10 VT62 Terminal Interface

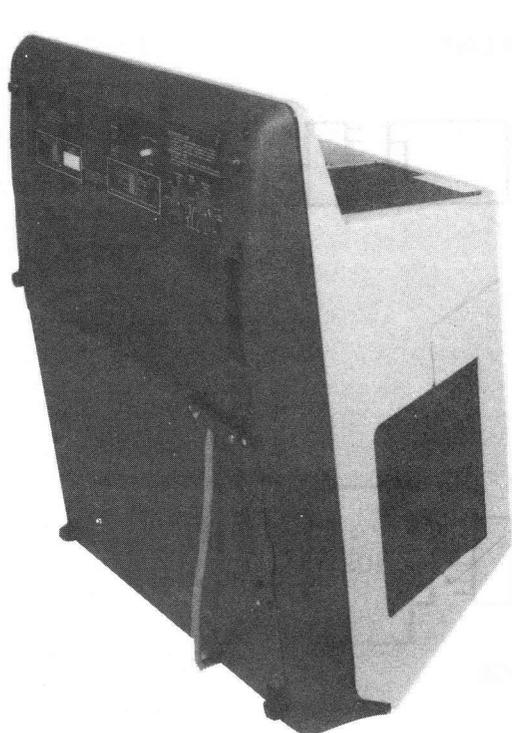
2. Tip the unit back until it is stable. This is done to gain access to the operator control switches (Figure 2-11).
3. Set the ON-LINE/OFF-LINE switch to the OFF-LINE position (Figure 2-12).
4. Return the VT62 to its proper operating position.
5. Plug the 3-pronged line cord of the unit into an electrical outlet that meets the requirements of the terminal.

#### 2.4.1.2 Self-Test Procedure

1. Make certain that the CAPS LOCK key on the main keyboard is not in the pressed position.
2. Apply power to the terminal by setting the ON/OFF switch to ON. This switch is located on the rear of the terminal.

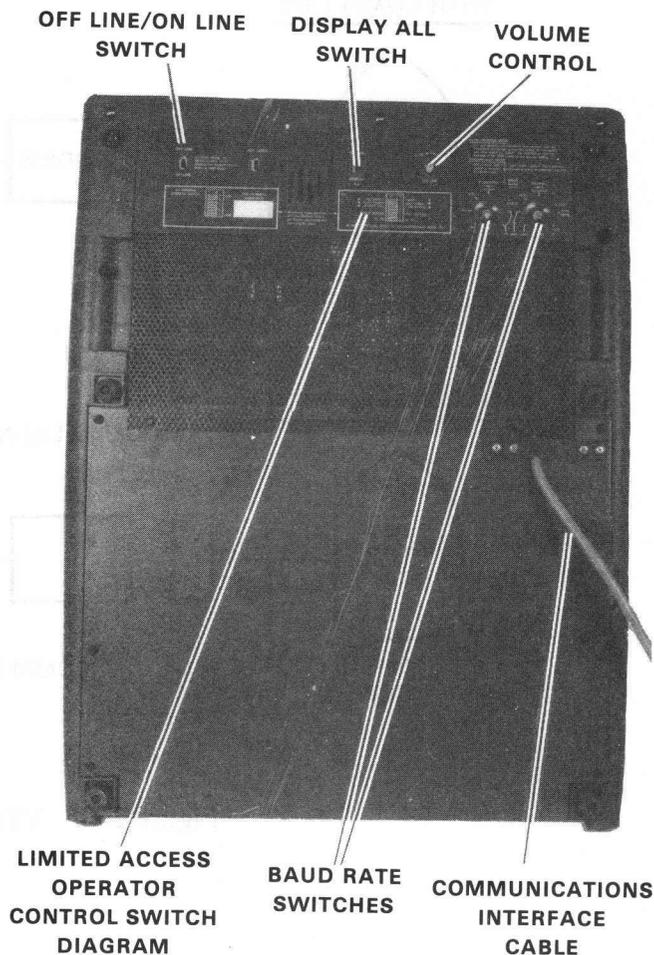
#### NOTE

The terminal enters a self-test mode immediately upon power-up. If the terminal CAPS LOCK key is in the released position, the VT62 enters Long Self-Test mode. The VT62 will remain in this mode until the CAPS LOCK key is pressed.



8595-10-A0028

Figure 2-11 Operator Controls Access Position



8595-9-A0003

Figure 2-12 Terminal, Bottom View,  
Base Front Screen Installed

3. Press the CAPS LOCK key.

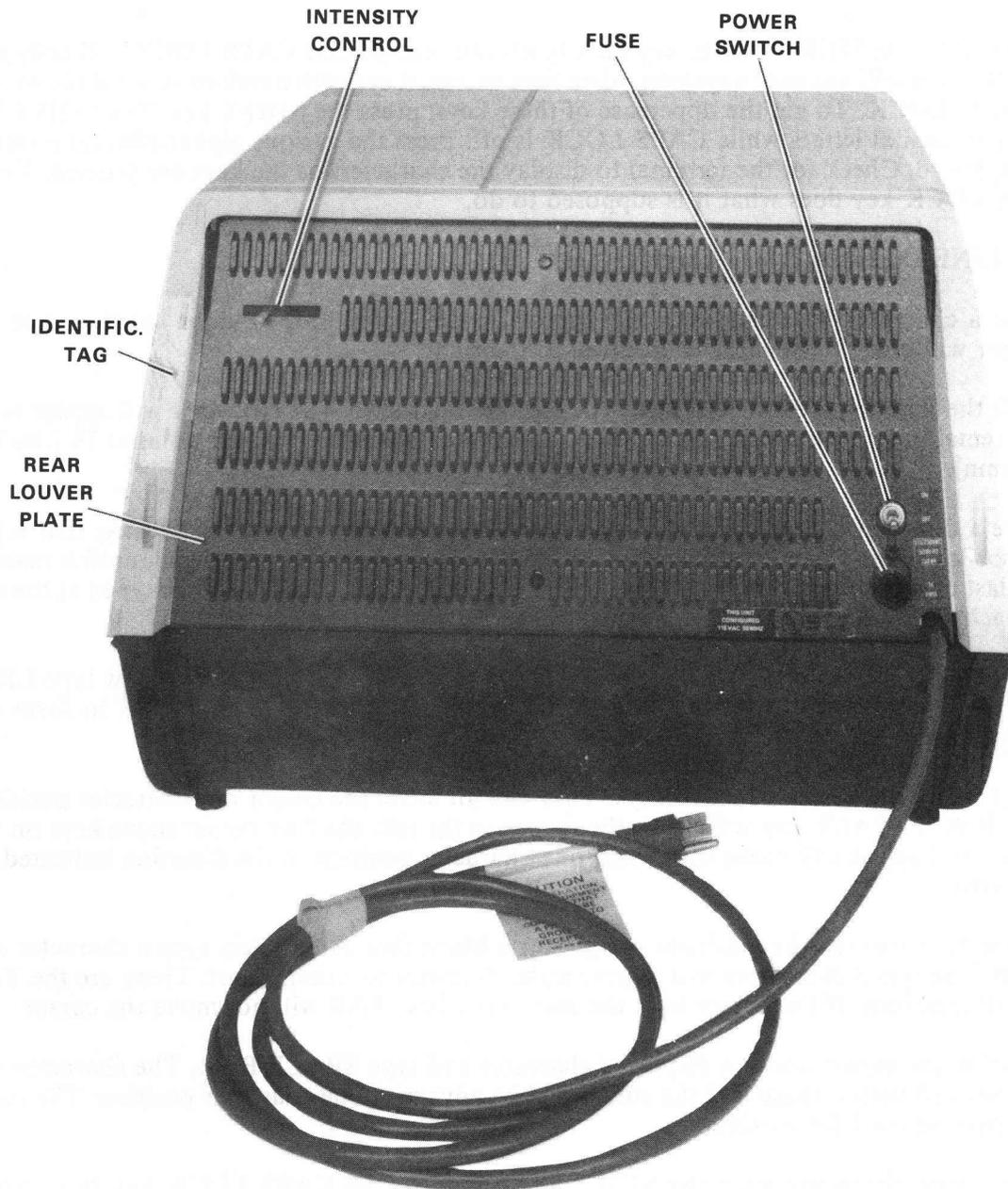
**NOTE**

When the CAPS LOCK key is pressed, the terminal enters Short Self-Test mode. When the Long Self-Test reaches the bottom of the current screen, it then executes the Short Self-Test one time (approximately ten seconds). At the end of the Short Self-Test, the terminal exits the Short Self-Test mode, the cursor goes to the HOME position (at the top left corner of the display), and the keyboard is enabled.

4. Remove power from the terminal by setting the VT62 ON/OFF power switch to OFF.
5. With the CAPS LOCK key still in the pressed position, reapply power to the terminal by setting the ON/OFF switch to ON.
6. Make certain that the terminal enters Short Self-Test mode and executes the Short Self-Test one time (approximately ten seconds).

**2.4.1.3 Flashing Cursor Check** – At the end of the Short Self-Test, the terminal cursor will go to the HOME position and the keyboard will be enabled. Verify that the flashing cursor appears at the upper left corner of the screen (the HOME position).

Using the intensity control located on the rear of the terminal (Figure 2-13) adjust the brightness of the cursor to the desired level. For comfortable viewing, set the brightness to the lowest possible level while still maintaining good character definition. High screen intensity causes characters to become fuzzy and prolonged viewing of the bright display is very tiresome.



8595-5-A0005

Figure 2-13 Terminal, Rear View

**2.4.1.4 Display All Check** – The terminal may display a dot in each character position. This is caused by the DISPLAY ALL switch being in the DISPLAY ALL position.

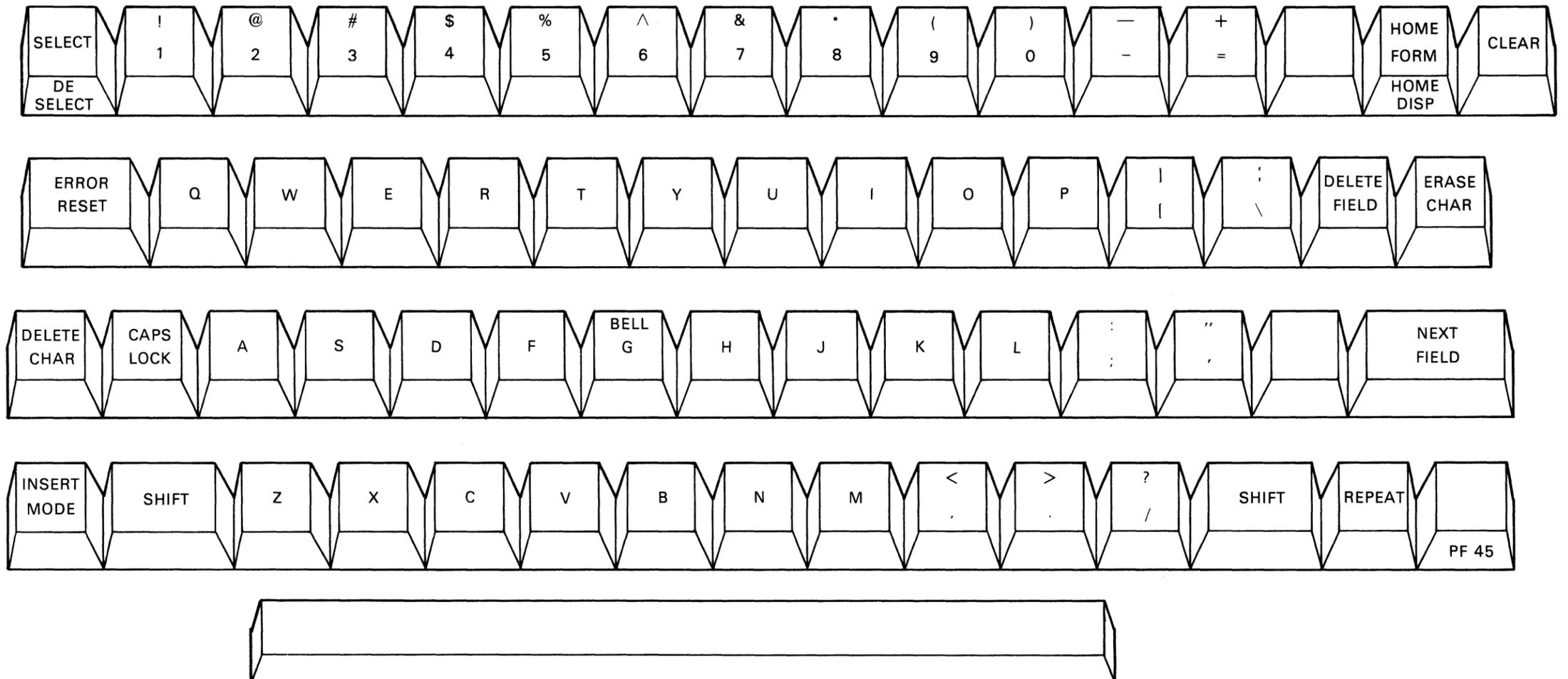
Move the DISPLAY ALL switch (Figure 2-12) several times between the DISPLAY ALL and normal positions and verify that these dots can be made to appear and disappear. Place the DISPLAY ALL switch in the normal position. (The normal position of the DISPLAY ALL switch is toward the front of the terminal.)

**2.4.1.5 CAPS LOCK Check** – Note the key labeled CAPS LOCK on the main keyboard (Figure 2-14).

This key is similar to the SHIFT LOCK key on a typewriter except that CAPS LOCK will only give you capital letters and will not shift anything other than letters. Keys with numbers or symbols are not affected by CAPS LOCK. To get the uppercase of these keys, press the SHIFT key. The SHIFT key will also give you capital letters while CAPS LOCK is off. Press the various alphabetic and numeric keys on the keyboard. Check for the terminal to display the characters as the keys are pressed. Verify that the CAPS LOCK key does what it is supposed to do.

**2.4.1.6 OFF-LINE Operating Mode Check**

1. Type a character. The character will appear on the screen at the cursor location, and the cursor will move one character location to the right.
2. With the REPEAT key held down, type another character. The character will appear in all the remaining locations of the top line, and the cursor will be under column 79 (the last column).
3. Type a different character. It will replace the last character in the line because that is the cursor location. The cursor advances automatically as characters are typed, until it reaches the last character position in the line. RETURN and LINE FEED must be typed at the end of each line.
4. Type the RETURN key. The cursor will move to the left end of the line. Now type LINE FEED. The cursor will move down to the next line. Type a few characters to form the second line.
5. Try the rest of the cursor move keys. They will all move the cursor one character position. The BACK SPACE key will move the cursor to the left; the four cursor move keys on the auxiliary keypad will move the cursor one character position in the direction indicated by the arrow.
6. Move the cursor to the left-hand margin on a blank line. Alternately type a character and TAB. The typed characters will appear eight character locations apart. These are the TAB STOP locations. If the cursor is at the end of the line, TAB will not move the cursor.
7. Position the cursor under a displayed character and type SPACE BAR. The character will be replaced with a space and the cursor will be advanced one character position. The space bar may be used for erasing.
8. Type some characters with the SHIFT key and then the CAPS LOCK key held down. Alphabetic characters will appear in uppercase when typed with either SHIFT or CAPS LOCK pressed. Numeric keys are not affected when CAPS LOCK is down. Symbols associated with numeric keys are displayed only when the numeric keys are typed with the SHIFT key held down.



MA-0930

Figure 2-14 VT62 Main Keyboard

9. Move the cursor to the left-hand margin of the bottom line. With REPEAT held down, type a character key and fill the bottom line.

Type RETURN and LINE FEED. The terminal will move all lines up by one position erasing the bottom line for new entries. The top line is lost in this operation. This is called scrolling and will occur whenever a LINE FEED is typed while the cursor is on the bottom line.

#### **2.4.1.7 Escape, Control, and Function Commands** – Escape and control commands are invoked by:

1. Pressing the ESC (escape) key (Figure 2-14) and then, following with a certain character to identify which escape function is to be performed.
2. Holding down the CTRL (control) key and simultaneously typing a character to identify which control function is to be performed.

Note that the ESC key and the FCN key need only be typed once and then followed by the identifying character (or characters), while the CTRL key must be held down while the identifying character is being typed.

#### **2.4.1.8 Escape, Control, and Function Command Checks**

1. Audible Alarm Check
  - a. Hold down the CTRL key and simultaneously type the letter G.

Verify that the audible alarm sounds. If the audible alarm does not sound, adjust the VOLUME potentiometer on the bottom of the terminal and repeat the procedure.

2. Terminal Self-Test
  - a. Press the CAPS LOCK key.
  - b. Press the ESC key.
  - c. Press the S key. The terminal should execute a self-test.

During the test, the screen will be filled with characters and the audible alarm will sound. Upon completion of the test (approximately 5 seconds), the cursor will appear at home position and the screen will reset.

- d. Release the CAPS LOCK key.

## CHAPTER 3

# OPERATION, PROGRAMMING, AND OPERATOR MAINTENANCE

### 3.1 CONTROLS AND INDICATORS

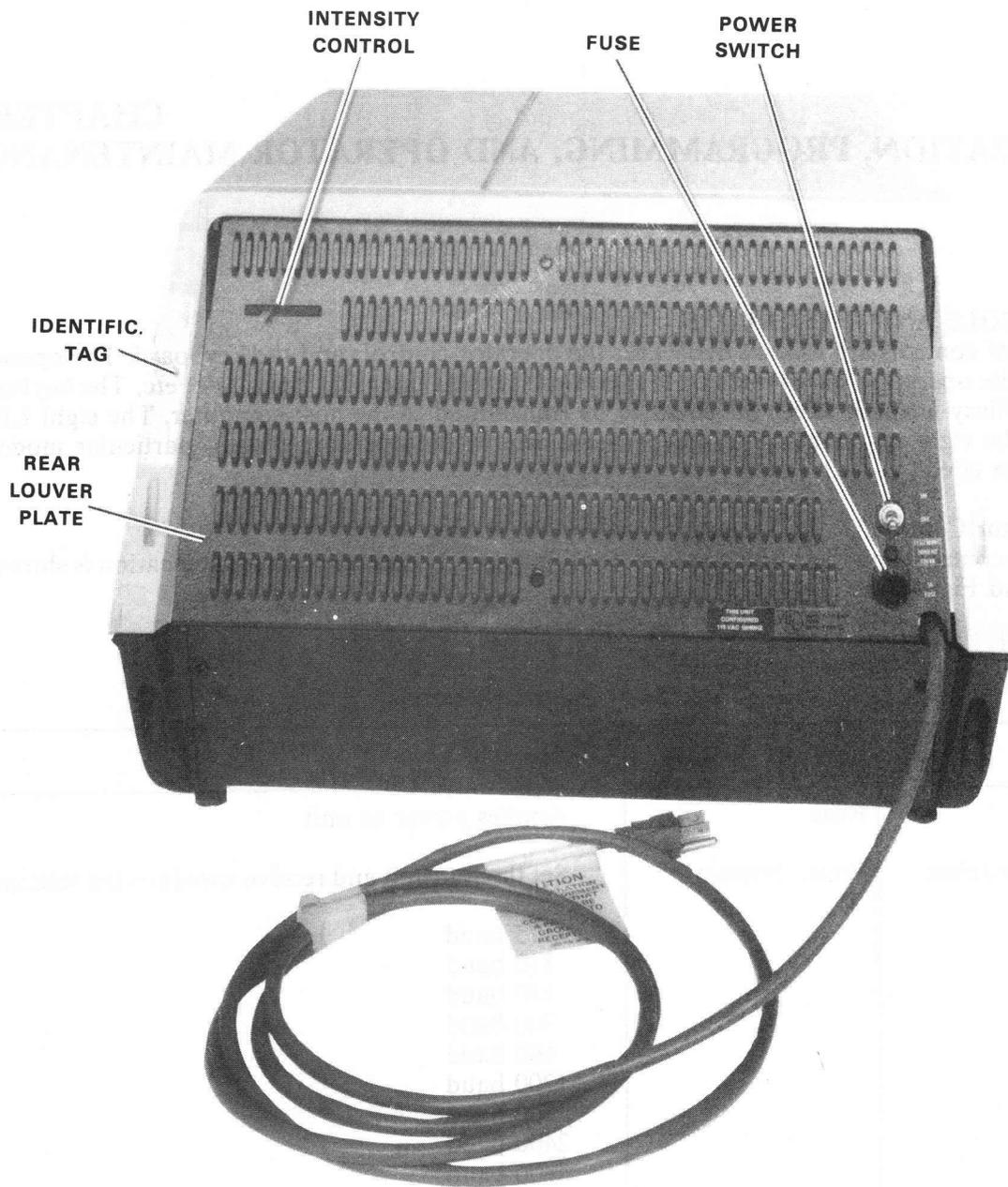
Two groups of controls are used in the VT62: the operator controls and the keyboard. The operator controls set the operating parameters of the terminal such as intensity, baud rate, etc. The keyboard (and the auxiliary keypad) select the codes to be transmitted to the host computer. The eight LEDs (located to the right of the auxiliary keypad) indicate to the operator when a particular mode or condition is in effect.

#### 3.1.1 Operator Controls

The operator controls and their functions are listed in Table 3-1 and their physical location is shown in Figure 3-1 and Figure 3-2.

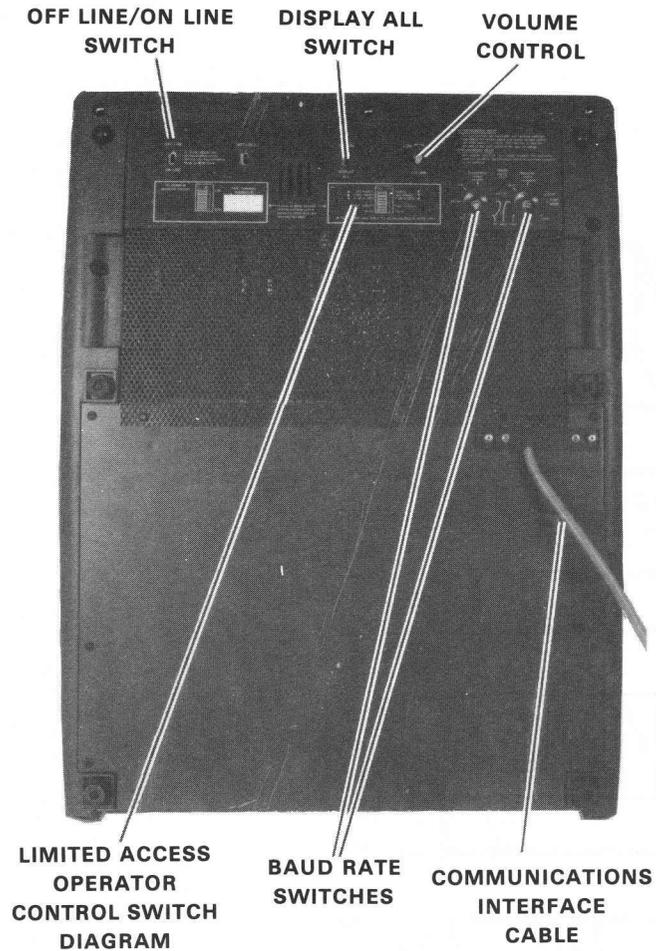
**Table 3-1 Operator Controls**

Control	Location	Function
ON/OFF	Rear	Applies power to unit
Baud Rate Switches	Front, bottom	Set the transmit and receive speeds of the terminal:  75 baud 110 baud 150 baud 300 baud 600 baud 1200 baud 1800 baud 2400 baud 4800 baud 9600 baud and select SYNC Mode.
ON-LINE/OFF-LINE	Front, bottom	Selects remote or local operation
DISPLAY ALL Switch	Front, bottom	Allows display of normally non-displayed characters
VOLUME	Front, bottom	Provides volume control of keyclick and bell alarm
INTENSITY	Rear	Sets video display intensity level



8595-5-A0005

Figure 3-1 Terminal, Rear View



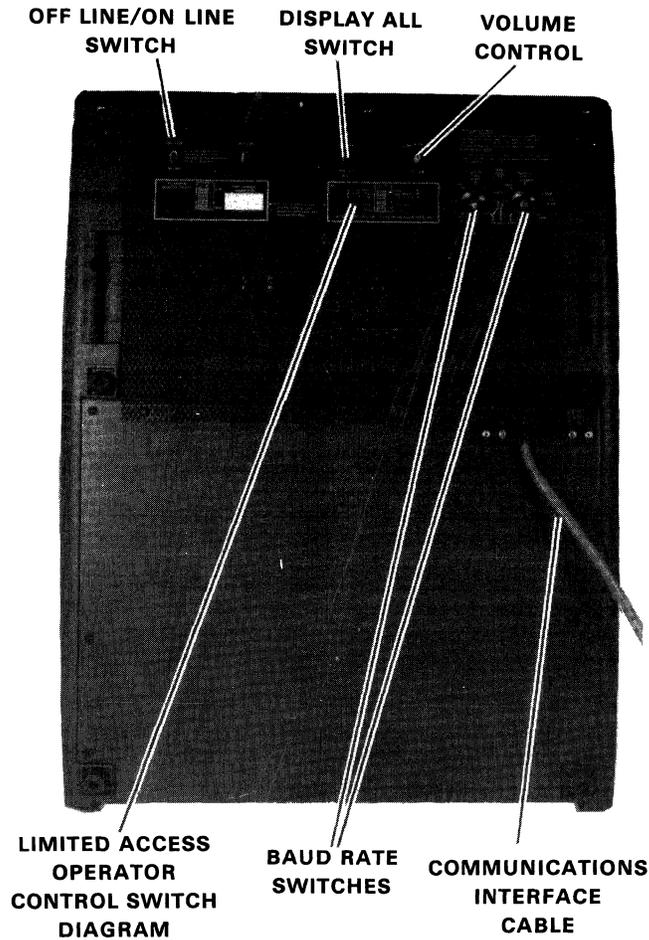
8595-9-A0003

Figure 3-2 Terminal, Bottom View

### 3.1.2 Keyboard

Input data to the host computer is entered by typing keys on the main keyboard or the auxiliary keypad. The keyboard transmits uppercase and lowercase alphabetic code, numeric code, firmware-defined code, and user-defined code. Figure 3-3 shows the main keyboard of the VT62.

Fifty-two of the keyboard keys are used to generate any of the ninety-five displayable ASCII characters (040 through 176). The ten remaining keyboard keys are used to implement keyboard operations which are useful in a form-filling application.



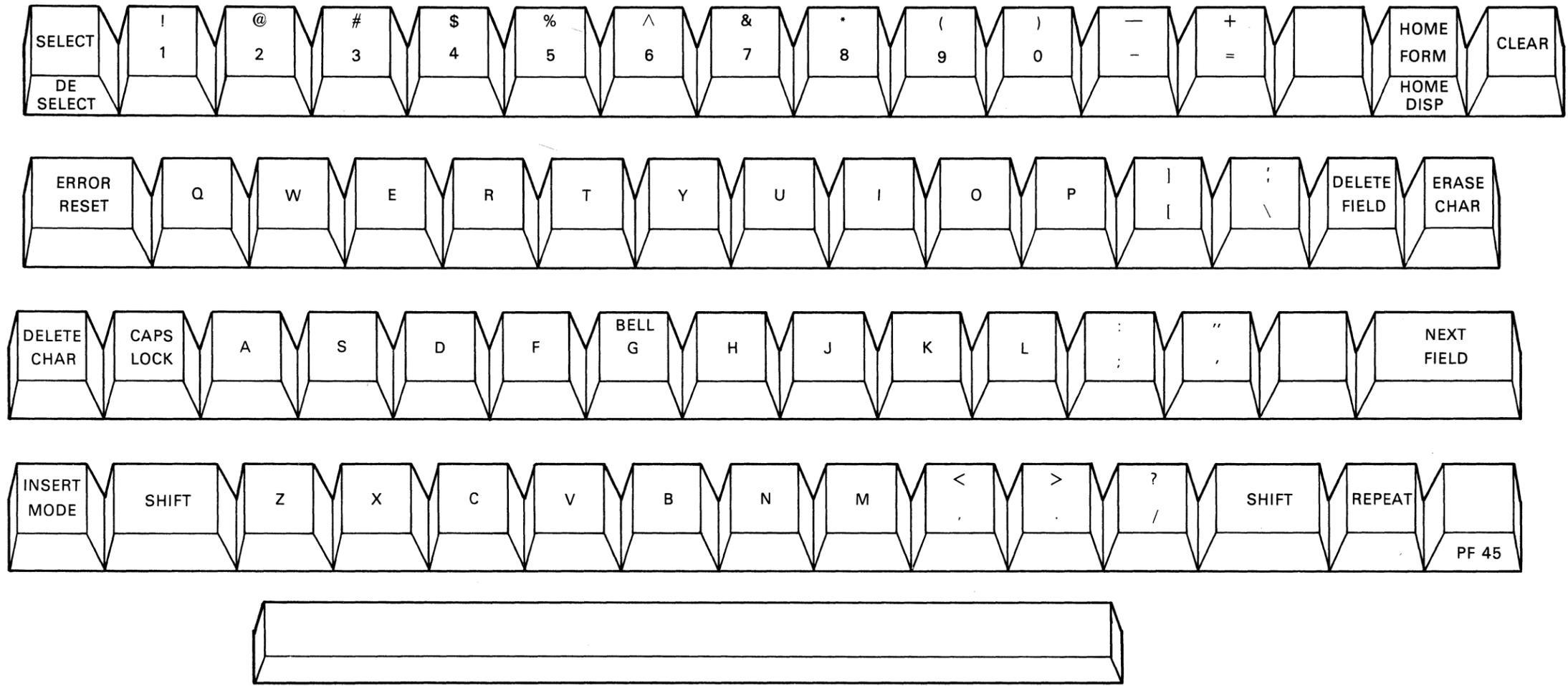
8595-9-A0003

Figure 3-2 Terminal, Bottom View

### 3.1.2 Keyboard

Input data to the host computer is entered by typing keys on the main keyboard or the auxiliary keypad. The keyboard transmits uppercase and lowercase alphabetic code, numeric code, firmware-defined code, and user-defined code. Figure 3-3 shows the main keyboard of the VT62.

Fifty-two of the keyboard keys are used to generate any of the ninety-five displayable ASCII characters (040 through 176). The ten remaining keyboard keys are used to implement keyboard operations which are useful in a form-filling application.



MA-0930

Figure 3-3 VT62 Main Keyboard

**3.1.2.1 Alphabetic Keys** – All alphabetic keys transmit in uppercase and lowercase code. Uppercase is transmitted when a key is typed while either or both SHIFT keys are down, or while the CAPS LOCK key is down. (The CAPS LOCK key does not affect codes transmitted by keys other than the alphabetic keys. The CAPS LOCK key does have a maintenance function which will be discussed later.) Uppercase and lowercase codes are listed in Table 3-2.

**3.1.2.2 Numeric/Symbol Keys** – Numeric and symbol key codes are listed in Table 3-3.

**Table 3-2 Alphabetic Key Codes\***

<b>Key</b>	<b>Uppercase ASCII Code</b>	<b>Lowercase ASCII Code</b>
A	101	141
B	102	142
C	103	143
D	104	144
E	105	145
F	106	146
G	107	147
H	110	150
I	111	151
J	112	152
K	113	153
L	114	154
M	115	155
N	116	156
O	117	157
P	120	160
Q	121	161
R	122	162
S	123	163
T	124	164
U	125	165
V	126	166
W	127	167
X	130	170
Y	131	171
Z	132	172

\*All codes are expressed in octal.

**Table 3-3 Numeric/Symbol Key Codes**

<b>Key</b>	<b>Neither SHIFT Key Down</b>	<b>Either or Both SHIFT Keys Down</b>
1	061	041 (!)
2	062	100 (@)
3	063	043 (#)
4	064	044 (\$)
5	065	045 (%)
6	066	136 (^)
7	067	046 (&)
8	070	052 (*)
9	071	050 (())
0	060	051 (())
-	055	137 (-)
=	075	053 (+)
[	133	135 ([])
;	073	072 (:)
'	047	042 (")
,	054	074 (<)
.	056	076 (>)
/	057	077 (?)
\	134	174 (:)
^	140	176 (=)
μ	173	175 (())
{		(~)
		()

**3.1.2.3 Function Keys** – The ten main keyboard function keys are used to implement keyboard operations in a form-filling application. These keys and their associated functions are listed as follows:

**HOME**

**FORM:**

If the form fields are enabled, the HOME FORM key moves the cursor to the first character of the first home field. This is the unshifted mode of the HOME key.

**DISPLAY:**

If the menu fields are enabled, the HOME DISPLAY key moves the cursor to the first character of the first home field. HOME DISPLAY is the shift mode of the HOME key.

**ERASE CHAR**

If the current field is left justified and the cursor is not at the beginning of the field, the ERASE CHAR key will cause the cursor to backspace and then erase the character under the cursor. If the cursor is at the beginning of the field, the ERASE CHAR key leaves the cursor there and erases the character under the cursor. If the field is left justified, the text collapses from right to left; if the field is right justified, the text collapses from left to right. The positions left vacant by the justify operation will be filled with the clear character specified in the Field Definition for that field.

### **NEXT FIELD**

If the current field is the last field, the NEXT FIELD key will clear to the end of the current field. If the current field is not the last field, the NEXT FIELD key will clear to the end of the current field and position the cursor to the first character of the next field. A field check will be done no matter what position the cursor is in when the key is pressed.

### **DELETE FIELD**

When selected, the DELETE FIELD key will set the current field to all clear characters and move the cursor to the beginning of the current field.

### **CLEAR**

#### **DISPLAY AREA:**

When the CLEAR key is selected, all menu fields are restored to the normal video state. They are deselected and the cursor is positioned to the beginning of the first menu field.

#### **FORM AREA**

When the CLEAR key is selected, all form fields are cleared to their clear character values and the cursor is positioned to the beginning of the first form field.

### **BLANK KEY (PF45)**

The BLANK key (PF45) is a user programmable function key. See Appendix A for more information.

### **DELETE CHAR**

When selected, the DELETE CHAR key will delete the character under the cursor. If the field is left justified, the text will collapse from right to left; if the field is right justified, the text will collapse from left to right.

### **SELECT/DESELECT**

This key is only enabled when the terminal is in menu mode and the cursor is in the Display area. Unshifted a menu item is selected. Shifting will deselect a menu item. Selected menu items are displayed in reverse video and deselected menu items are displayed in normal video.

### **INSERT MODE**

Unshifted enables INSERT mode; shifted disables INSERT mode.

In the INSERT mode, if a field is left justified, each character typed into the field is entered under the cursor and all characters at or to the right of the cursor are shifted one position to the right. If the field is right justified, each character typed is entered under the cursor and all characters at or to the left of the cursor are shifted one position to the left.

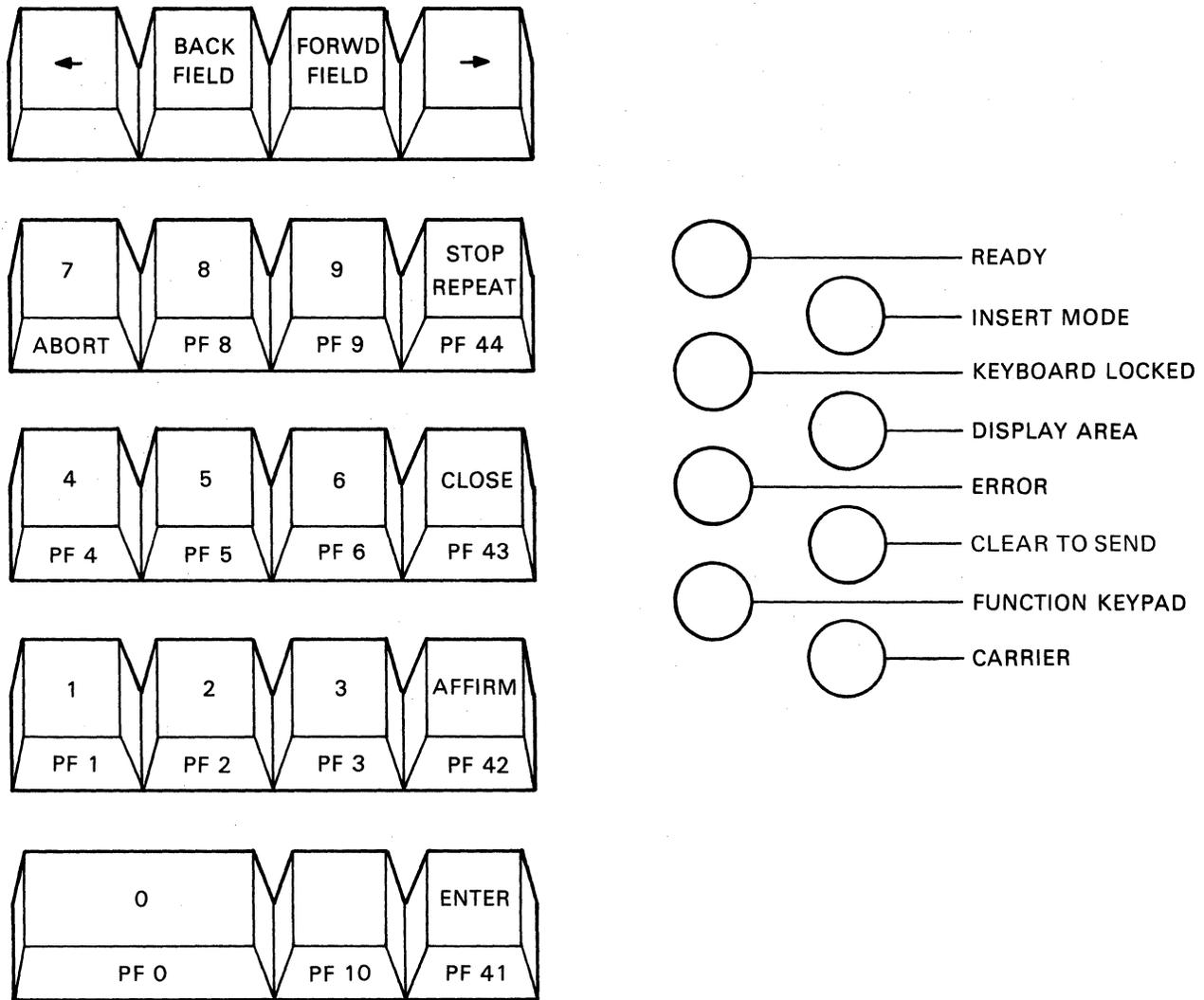
### **BOUNDARY CONDITION**

When a field is filled (no clear characters on the left for right justified field or on the right for left justified field), additional characters typed will not be entered into field and the bell will ring.

### **ERROR RESET**

When an error occurs, a message is printed on the bottom line (if error messages are present), the bell rings, the ERROR LED is lit, and the terminal enters ERROR MODE. Until the operator presses the ERROR RESET key, thereby, acknowledging the error, all keystrokes simply ring the bell. Errors occur if field validation fails, or if the host computer selects an error message to be printed.

**3.1.2.4 Auxiliary Keyboard** – The 19-key auxiliary keypad (Figure 3-4) is provided for user applications.



MA-0931

Figure 3-4 VT62 Auxiliary Keypad and LED Display

The numeric keypad is functionally self-contained. If all replies in the form are numeric, then the entire form can be entered and sent with one hand which need never move from the pad. Most keys on the numeric pad serve two functions. The first function is indicated by the label on the top of the key cap. The second function is indicated by a label on the front of the key cap. This second function is obtained by striking the desired key while the SHIFT key is pressed.

The terminal can be placed in a special mode, selectable by the host software, in which the PROGRAM FUNCTION keys operate in single keystrokes, without the use of the SHIFT key. In this case, the program foregoes the normal function of the numeric keypad.

These keys and their associated functions are as follows:

**Forward Field**

If the current field is the last field, move the physical cursor to the last position in field and the logical cursor past field so that no data entry is allowed if a key is pressed. Otherwise, move cursor to the first character position of next field without changing anything in the rest of the current field. Field check will be done no matter what position the cursor is in when the key is pressed.

**Backward Field**

If the current field is the first field, then move cursor to beginning of current field. If the cursor is at the beginning of the current field then move cursor to beginning of previous field. Otherwise, move cursor to beginning of current field.

**Cursor Left**

If the cursor is at the leftmost position of a field, then no-operation. Otherwise, move cursor left one position.

**Cursor Right**

If cursor is at the rightmost position of a field, then no-operation. Otherwise, move cursor right one position. If auto-tab is enabled and pressing this key causes the cursor to be moved to the next field, field checking will be done on the field exited.

**Enter**

User programmable\*

Affirm-PF42

User programmable\*

Stop Repeat-PF44

User programmable\*

Close-PF43

User programmable\*

**0 to 9 and .**

If the keypad is in “numeric” mode then pressing unshifted 0 to 9 or . results in 0 to 9 or .; pressing shifted 0 to 9 or . results in PF0 to PF6 and abort and PF8 to PF10. If the keypad is in “function” mode, then pressing 0 to 9 or . results PF0 to PF6 and abort and PF8 to PF10, independent of the shift key.

**3.1.2.5 The REPEAT Key** – The REPEAT key (on the main keyboard) is used in conjunction with other keys; it does not transmit a code. Any key that transmits a code to the host will transmit that code repeatedly if pressed while the REPEAT key is down. Keys that transmit more than one code will transmit their sequence repeatedly if pressed while the REPEAT key is down.

The following keys repeat if held down (without the use of the REPEAT key):

VT62 main keyboard – NEXT FIELD, ERASE CHAR, and DELETE CHAR.

VT62 auxiliary keyboard – Cursor left and cursor right. Backward Field and Forward Field

All keys for displayable characters can be forced to repeat by the use of the REPEAT key on the main keyboard.

---

\*See Appendix A for programming information

### 3.1.3 Indicators

The LED display to the right of the keypad is labeled as follows:

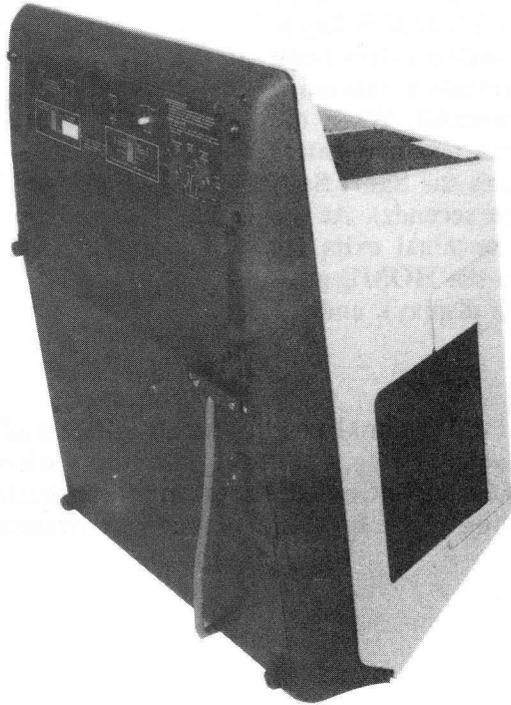
The LED labeled. . .	If on when. . . .
READY	The DDCMP startup sequence has been successfully received.
KEYBOARD LOCKED	The operator may not type in data or select menu items.
FUNCTION KEYPAD	The keypad is in function (as opposed to numeric) mode.
INSERT MODE	The terminal is in insert mode.
DISPLAY AREA	The cursor is in the display area.
CLEAR TO SEND	A communication line has been established between the terminal and the host computer and the terminal can now send data.
CARRIER	A communication line is established between the terminal and the host computer.
KEYING ERROR	The terminal is in error.

The layout of the LED display is shown in Figure 3-4.

## 3.2 OPERATING PROCEDURE

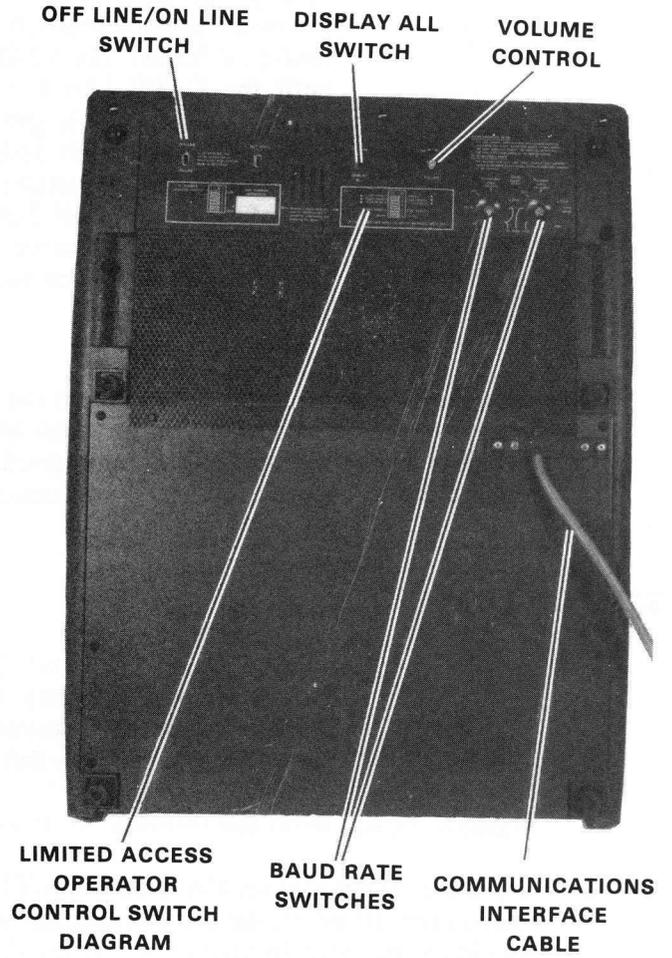
### 3.2.1 OFF-LINE

1. Make certain that the terminal line cord is not connected to an electrical outlet and that the power (ON/OFF) switch is in the OFF position.
2. After making sure that there is room on the platform on which the terminal is located, tip the unit back until it is stable. This must be done to gain access to the switches on the underside of the unit (Figure 3-5).
3. Set the ON-LINE/OFF-LINE switch to the OFF-LINE position (Figure 3-6).
4. Return the VT62 to its proper operating position.
5. Plug the three-prong line cord of the unit into an electrical outlet that meets the requirements of the terminal.



8595-10-A0028

Figure 3-5 Terminal, Operator Control Switch Access Position



8595-9-A0003

Figure 3-6 Terminal, Bottom View, Base Front Screen Installed

6. Apply power to the terminal by setting the ON/OFF switch to ON. This switch is located on the rear of the terminal.

#### **NOTE**

**The terminal enters a self-test mode immediately upon power-up. If the terminal CAPS LOCK key is not in the pressed position, the terminal enters Long Self-Test Mode. The VT62 will remain in this mode until the CAPS LOCK key is pressed. When the CAPS LOCK key is pressed, the terminal enters Short Self-Test Mode and executes the Short Self-Test one time (approximately ten seconds). At the end of the Short Self-Test, the terminal exits the self-test mode, the cursor goes to the HOME position (at the top left corner of the display), and the keyboard is enabled.**

7. Use the brightness control located on the rear of the terminal to adjust the brightness of the cursor to the desired level. For the most comfortable viewing, set the brightness to the lowest possible level while still maintaining good character definition. High screen intensity causes characters to become fuzzy and prolonged viewing of the bright display is very tiresome.

### **3.2.2 ON-LINE**

#### **CAUTION**

**Do not attempt to switch the unit ON-LINE now. Damage to the host operating system could result. Please comply with the following procedure.**

1. Remove power from the terminal by switching the power (ON/OFF) switch to OFF.
2. Tip the unit up and set the BAUD RATE switches to the desired frequency. Transmission speed must be set to the same frequency as the host's receiving speed; receiving speed must be set to the same frequency as the host's transmission speed.
3. Set the ON-LINE/OFF-LINE switch to the ON-LINE position.
4. Set the ON/OFF switch to ON to restore power to the terminal. When the Short Self-Test has terminated, the terminal can communicate with the host.

### **3.3 USER PROGRAMMING**

Appendix A explains the concepts necessary to understand the workings of the VT62, and describes in detail the effects of all the commands to which the terminal responds.

### **3.4 USER MAINTENANCE**

The keyboard circuits contain the only moving parts of the terminal and require no preventive maintenance by the owner. The terminal may be cleaned with soap and water or any mild detergent. Although the terminal's shell provides superior resistance to damage from solvents, cleaners with solvents should not be used.

The unit packaging is not meant to be weatherproof: there are several openings in the case through which liquids, coins, paper clips, and other objects can fall. Such objects will disturb the electronic operation of the terminal if they come in contact with the circuitry. For this reason, avoid putting drinks and metal objects on the top of the terminal or using excessive water to clean the terminal. The keyboard is an area where the electronics are particularly close to the exterior. Rubbing the keys with a dry or barely moist cloth should suffice to clean them. Do not remove the keycaps to clean them more thoroughly; damage may result in the switch contacts if they are replaced incorrectly.

#### **3.4.1 Irregularities in the Appearance of the Screen**

At some time in its life, the picture tube may appear to darken at each character position. This is a normal phenomenon; it does not affect the life expectancy of the tube and will not interfere with the display of characters when the unit is in operation. It is caused by the fact that the terminal lights up the screen only at the character positions, breaking in the phosphor at those positions on the screen.

Not all units contain the same type of picture tube. For this reason, the color of the screen may not be the same from unit to unit.

If, for these or other reasons, the appearance of the screen is not satisfactory, colored filters may be attached to the screen. A special nib was designed at the base of the screen in order to hold such filters in place.

### **3.5 DEC MAINTENANCE ARRANGEMENTS**

There are four ways that an owner can arrange for service on a VT62.

1. **Depot Maintenance** – Service will be done at the DEC depot. You buy a mailer, pack the VT62 in it, and return it to the depot. Upon completion of service, DIGITAL returns the unit.
2. **On-Site Contract** – This is the traditional priority service contract, available with various coverage hours.
3. **On-Site, Per-Call Service** – In this arrangement, DIGITAL charges for time and materials actually used.
4. **On-Site User Maintenance** – Module-swapping kits are available. Failed modules are returned to the DEC depot for repair in the mailers in which the swapping kits are mailed.

In some geographical areas, additional options are available; contact your local DIGITAL Field Service Office for information.



## **APPENDIX A**

### **VT62 FUNCTIONAL SPECIFICATION**

#### **A.1 GENERAL**

The VT62 is a video terminal with major characteristics as follows:

- Block Mode
- DDCMP Protocol
- Polled Slave Station
- Multi-drop
- Full or Half-duplex
- Switchable:
  - Synchronous up to 4800 baud
  - Asynchronous up to 9600 baud
- Dial-up or Leased
- Forms Filling Functionality

The VT62 is a terminal designed specifically to be a part of a form-filling or other transaction processing system. The host computer defines the areas on the screen which the operator is allowed to enter, to fill in, or to select. The terminal provides most of the real-time detailed checking of operator responses, but the ability to extend some of the more complicated checking, such as file searches, to the host computer is provided. Operator errors are signaled by an audible tone, an error light, and a short, but descriptive, message of what went wrong. Since the host computer loads the message array, it is not necessary to restrict the messages to the English language.

The visible cursor is associated only with keyboard activity. Text sent by the host must be preceded by either the screen position where it begins, or a code which indicates that it is to be sent directly to the printer. Host activity, printer activity, and keyboard activity proceed simultaneously.

The VT62 communicates in block mode as specified by the DDCMP protocol, either asynchronously or synchronously, in full or half-duplex, utilizing the common datasets over either dial-up or leased lines.

The VT62 will operate as a DDCMP slave station and will operate only in a polled environment, regardless of whether the connection is multi-point or point-to-point.

All exchanges of information between the host and the terminal are completely controlled by the host. The terminal cannot, under any circumstances, send information of any kind to the host that was not requested and allowed by the host.

## **A.2 SCREEN FORMAT**

The display screen consists of 24 lines, with 80 character positions on each line.

The display screen may logically be divided into three areas: the Display/Menu Area, the Form Area, and the Error Line.

Lines 1 to 24 of the screen may be defined to be all Display/Menu Area, or all Form Area, or part Display/Menu Area and part Form Area and part Error Line. If both are defined, the Display/Menu Area and the Form Area are adjacent and separated on a whole line boundary, and the Display/Menu Area must be above the Form Area. The screen format is defined by means of the RESET AND DEFINE SCREEN command (Section A.9.11.1).

If error messages are defined, then line 24 will be used as the Error Line. The terminal displays an error message by first clearing line 24, and then printing the appropriate message there. Thus, it is a logical error to attempt to use line 24 as both part of the Display/Menu Area or Form Area and as the Error Line. If no error messages have been defined since the terminal has been powered up or since a GENERAL RESET command has been executed (which clears all error messages), line 24 is not cleared when an error occurs and can be used as part of the Display/Menu Area or Form Area.

## **A.3 CURSOR**

There is one visible cursor. When the terminal is in menu mode the cursor may be positioned in any menu field in the Display/Menu Area. When the terminal is in form mode, the cursor may be positioned within any form field in the Form Area. (The terminal can be in menu mode and form mode simultaneously).

The cursor is only used by the terminal operator to select menu fields or to indicate where data is to be entered within a form field.

The cursor is not used by the host to write text. Each command which the host issues to the terminal contains sufficient information to direct the terminal microprocessor to the position of the data in the screen buffer where data is to be entered.

## **A.4 MENU DEFINITIONS**

A menu is a sequence of menu fields. A menu field is a defined sub-area of the Display Area. A menu field definition consists of a row, column, count, and video attributes specification. A menu field may begin at any character position in the Display Area and extends from left to right. If the count for a menu field does not expire at the end (column 80) of a line, the field continues at the beginning (column 1) of the next line.

The operator moves the cursor from one menu field to another by means of the Forward Field or Next Field or Backward Field keys. The cursor may move within menu field by means of the Cursor Right and Cursor Left keys.

The host issues a set of menu definitions to the terminal by means of the <CLEAR MENU DEFINITIONS AND DEFINE MENU FIELDS> command. The terminal stores menu definitions received from the host and uses these definitions to identify which field (if any) the terminal operator selects when the menu select key is typed.

A maximum of 128 total menu fields and form fields may be defined with no restriction as to the number of each so long as the total of 128 is not exceeded.

## A.5 FORM DEFINITIONS

A form field is a defined sub-area of the Form Area. A form field definition consists of a row, column, count, video attribute, and field attribute specification. A form field may begin at any position in the Form Area and extends from left to right. If the count for a form field does not expire at the end (column 80) of a line, the field continues at the beginning (column 1) of the next line.

The operator moves the cursor from one field to another by means of the Forward Field or Next Field or Backward Field keys. The cursor may move within a form field by means of the Cursor Right and Cursor Left keys.

The host issues a set of form definitions to the terminal by means of the <RESET FORM DEFINITIONS AND DEFINE FORM FIELDS> command. The terminal stores the form definitions and uses them to control cursor positioning in the Form Area and to check the data entered into each field against the field attributes (e.g., alphabetic, numeric) for that field.

A maximum of 128 total form and menu fields may be defined with no restriction on the number of each so long as the total of 128 is not exceeded.

### A.5.1 Form Field Attributes

The following attributes may be defined for each field:

- C1: Alphabetic (A-Z or a-z or Blank)
- C2: Alphanumeric (A-Z or a-z or Blank or 0-9)
- C3: Numeric (0-9)
- C4: Signed Numeric (0-9, or., or., or -, or +)
- C5: Any Displayable (ASCII 040-176)
- C6: Field must not be empty (i.e., not all clear characters)
- C7: Field must be full or empty
- C8: Auto-transmit
- C9: Auto-tab
- C10: Clear Character: The character to which a form field is initialized by a <CLEAR> Command; the character used to fill in places in a form field when some operator key entry is made which causes characters to be deleted in that form field
- C11: Right Justified
- C12: Protected Field: Field may not be modified from the keyboard.

C to C5 are mutually exclusive (only one may be set).

C1 to C5 is checked as each character is entered.

C6 to C9 are checked when field is exited in the forward direction.

C10 takes effect when field is cleared.

C11 right justified.

C12 is checked when the field is entered; if the field cannot be modified from the keyboard, then the cursor is moved on to the next field in the same direction (forward or backward).

If a data character is keyed into a field and is not an allowed character as defined by the Form Field Attribute, the terminal will always enter Error Mode (the bell will ring, the keying error light will be on, and the keyboard will be locked). If an error message has been defined by the <DEFINE ERROR MESSAGE> command, line 24 will be cleared and that message will be displayed. If an error message has not been defined, line 24 will not be cleared or altered.

## **A.6 DISPLAY AREA USAGE**

The Display Area, if defined, may be used in one of two modes: menu mode and non-menu mode.

### **A.6.1 Menu Mode**

Menu mode is enabled if: there are one or more menu fields defined; and there are one or more function keys which are defined by a <REQUEST INPUT> command to send menu fields.

The purpose of menu mode is to allow the terminal operator to select, in general, multiple items from a menu, and then to transmit these choices to the host by pressing a function key that has been appropriately defined.

The cursor (by means of the <Home Display> key) may be placed in the Display Area only when the terminal is in menu mode. Once in the Display Area, the cursor may be moved among the menu fields in the Display Area. The terminal operator, upon positioning the cursor over any character position in a menu field, may record the choice of that field by pressing the <Select> key. In order to let the terminal operator see what choice has been made, the menu field will be displayed in reverse video when selected. If the cursor is in a position which is defined to be part of more than one menu field, only one menu field will be selected and that menu field will be displayed in reverse video mode. The <Deselect> key deselects the current menu item and causes it to be displayed in normal video. <Deselect> is the shifted position of the <Select> key.

The terminal operator may make as many menu selections as allowed by the menu attributes (Section F.6.1.1). When an appropriately defined function key is typed, those menu field numbers (and possibly content) which are selected will be transmitted to the host, providing that an input request is pending.

A <REQUEST INPUT> command can specify that either menu field numbers only, or menu field numbers together with the content are to be transmitted to the host. For single menus, the "numbers only" transmission would be sufficient. However, if the host wishes to sequentially transmit identically formatted menus, the host can identify a particular operator choice by the menu content being transmitted along with the field number. The terminal operator would control the reception of successive menus by pressing an appropriately defined function key when a new menu is desired.

**A.6.1.1 Menu Attributes** – A menu may have one or both of the following two attributes:

1. no more than one menu item may be selected
2. no less than one menu item may be selected.

### **A.6.2 Non-Menu Mode**

If menu mode is not enabled the terminal defaults to non-menu mode. Non-menu mode is used to dynamically write text to the Display Area.

The host can clear, write text to, and scroll the Display/Menu area by means of the commands listed in Section A.9.8.2.1.

By using these commands and appropriately defined function key responses, an interaction can be set up with the operator such that the operator can control the flow of text to the display area and could, for example, call for text on a line by line basis by using one function key or perhaps call for the host to send a new set of text by using another function key.

## **A.7 FORM AREA USAGE**

### **A.7.1 Form Mode**

Form mode is enabled if: there are one or more form fields defined, and there are one or more function keys defined by a <REQUEST INPUT> command to send form fields.

The purpose of form mode is to allow the terminal operator to make keyboard data entries into the form fields, to have the terminal validate the data entered using the attributes for each field, and then to transmit the validated data to the host when the operator types an appropriately defined function key. When the terminal is in form mode, the cursor may be positioned within any form field. Inter field cursor movement is accomplished by using the <next field> <forward field>, and <backward field> keys. Intra field cursor positioning is accomplished by means of the <cursor left> and <cursor right> keys.

### **A.7.2 Packed and Unpacked Transmissions**

The terminal can transmit data in either packed or unpacked form. In packed form the terminal transmits only those fields which have modified state set and does not send trailing clear characters. In unpacked form, the terminal transmits every character in every field. The host can dynamically select packed or unpacked transmissions. Packed transmissions use the communications lines most efficiently, and unpacked transmissions use host CPU time most efficiently.

A field is set to a modified state by:

1. Simulating forms activity if so specified.
2. Typing a character (including a character which has been defined as the clear character) which is entered into the field.
3. Pressing the <delete character> key if the operation of the key can be accepted into the field and regardless of whether the content of the field was actually changed.
4. Pressing the <erase character> key if the operation of the key can be accepted into the field and regardless of whether the content of the field was actually changed.
5. Pressing the <delete field> key if the field did not already contain all clear characters when the key was pressed.
6. Pressing the <next field> key if the remainder of the field did not contain all clear characters.
7. Pressing the <clear> key if the field did not contain all clear characters at the time the key is pressed.

A field is cleared from a modified state by:

1. A <REQUEST INPUT> command if so specified in the command.
2. The following commands:  
**<RESET FORM DEFINITIONS AND DEFINE FORM FIELDS>**  
**<CLEAR FORM FIELDS>**  
**<CLEAR THEN WRITE FORM FIELDS>**  
**<WRITE ALL FORM FIELDS>**

The modified/unmodified state of a field is not altered by:

1. Exiting a field by pressing the <back field>, <forward field>, <home display>, <home menu>, <cursor left>, or <cursor right> keys.
2. Exiting a field by pressing the <next field> key if the remainder of the characters in the field are already clear characters.
3. Deleting a field by pressing the <delete field> key if the field is all clear characters when the key is pressed.

### **A.7.3 Auto-Transmit Mode**

A form field may be defined to be auto-transmit. An auto-transmit field is one which is transmitted to the host as soon as the terminal operator leaves the field in a forward direction. Prior to the auto-transmission, field validation is performed on the field according to any attributes specified for that field.

All form fields which have the auto-transmit attribute set will take effect as auto-transmit fields when the terminal is placed in auto-transmit mode. The terminal is placed in auto-transmit mode by executing a <REQUEST INPUT> command with <AUTO-TRANSMIT MODE> enabled. When auto-transmit mode is not enabled, auto-transmit attributes for individual fields are not in effect.

Auto-transmit is used primarily when the host wishes to extend the field validation accomplished by the terminal.

### **A.8 COMBINED DISPLAY/FORM AREA USAGE**

A <REQUEST INPUT> command can be issued with both the <MENU MODE> and the <FORM MODE> set. The terminal operator will then be able to make menu selections in the Display Area and enter data into the Form Area before typing an appropriately defined function key. The cursor can be freely moved back and forth between the Display Area and the Form Area by means of the <HOME FORM/HOME DISP> key. When an appropriately defined function key is finally typed, responses will be generated for the menu selections and/or the forms field data entered according to the definition for the function key typed.

### **A.9 HOST/TERMINAL/HOST MESSAGES**

All exchanges of information between the host and the terminal are completely controlled by the host. The terminal cannot, under any circumstances, send information of any kind to the host that was not requested and allowed by the host.

All information from the host to the terminal is in the form of Command Packets.

All information from the terminal to the host is in the form of Response Packets.

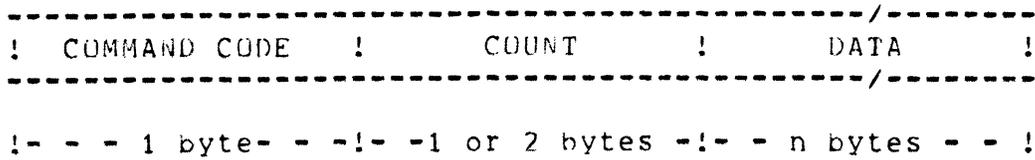
#### **A.9.1 Command**

A command consists of a command code, count and data sequence. The terminal upon receiving all the bytes in a command will validate the command and then execute the command.

A command is in no way related to a DDCMP block. It is possible that a single command would require several DDCMP blocks or that several commands could be transmitted in a single DDCMP block.

If a command requires more than one DDCMP block, there are no restrictions on the size of each block other than that they must be no greater than the maximum block size accepted by the VT62. (Refer to the definition of a Response Packet. The Response Packet Header contains information regarding the size of the terminal's command buffer.)

The format of a command is as follows:



**COMMAND CODE:** 8-bit binary encoded command identifier. Range: 1-255; is an illegal COMMAND CODE.

**COUNT:** A 7-bit or 14-bit binary encoded count of the length (in bytes) of the following data field. If the high-order bit of the first 8-bit byte of the count is set, then that first byte of the count is followed by a second 7-bit binary encoded character which is the most significant 7 bits of a 14-bit, binary encoded count.

**NOTE**

**A count byte is REQUIRED with ALL commands.  
A command byte=0 is legal for some commands. Refer to specific command formats.**

**REMARK**

**Counts which are in the range from 128 to 255 require a 14-bit representation.**

**DATA:** Any character in the range 0 to 255.

**A.9.2 Command Packet**

A Command Packet is a complete message which the host transmits to the terminal. A Command Packet consists of at least a Command Packet Header and may also have one or more commands. A complete packet may require more than one DDCMP block. There are no restrictions on how the Command Packet may be partitioned into DDCMP blocks other than that they must be no greater than the maximum block size accepted by the VT62. After the terminal has received a complete command packet, it validates and executes each command in the sequence before proceeding to the next command.

If an invalid command is encountered, the terminal aborts execution of the command packet and indicates an unsuccessful completion by setting the packet completion bit in the Response Packet Header. All of the commands in the command packet before the invalid command are executed correctly and all of the commands after the invalid command are ignored. If another Command Packet has been loaded after the Command Packet with an invalid command, the terminal will process it by validating and executing commands in a normal fashion. A response packet will be sent to the host even if not requested after an invalid command is encountered.

When all of the commands in a packet are received validly and executed, the Response Packet Header will have the Command Error (CE) bit set to 0 indicating successful packet completion. Note that with successful packet completion, the Response Packet will not be sent unless requested.

A Command Packet is limited in size to that specified for the VT62. If a Command Packet larger than this size is sent, the VT62 will then issue a DDCMP START to the host in response to any poll with the select bit set indicating that the terminal is in an undefined state and that the host must issue a DDCMP START for the terminal to become functional again.

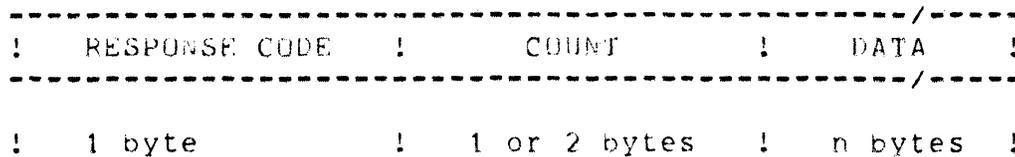
The terminal can have only one Command Packet in the process of being validated or executed at any point in time. Other Command Packets can be completely or partially loaded in the terminal if there is available Command Packet buffer space but will not be acted upon in any way until the current packet has been completely validated and all execution begun or been aborted by a command error. Refer to potential errors described under Input Commands. In particular, note that the commands which initiate input requests do not execute for the duration of the request; their execution is completed immediately and further commands (and Command Packets) may be processed while the operator is typing and before or while the data requested is being sent to the host. Also refer to the section on DDCMP protocol activity if DDCMP blocks cannot be received due to full Command Packet buffer.

### A.9.3 Response

A response characterizes the data (e.g., menu data, form data, function key data) which the terminal transmits to the host. A response consists of a response code, count and data sequence.

There is no limit to the size of a Response. The size of a Response does not affect the space available in the Command Packet buffer. (Internal remark – the terminal does not process commands while it is creating responses.)

The format of a response is as follows:



**RESPONSE CODE:** 8-bit binary encoded response identifier. Range: 1–255; 0 is an illegal Response Code.

**COUNT:** A 7-bit or 14-bit binary encoded count of the following data field. If the high-order bit of the first 8-bit byte of the count is set, then that first byte of the count is followed by a second 7-bit binary encoded character which is the most significant 7 bits of a 14-bit, binary encoded count.

#### REMARK

**Counts which are in the range from 128 to 255 require a 14-bit representation.**

**DATA:** Any character in the range 0 to 255.

#### A.9.4 Response Packet

A Response Packet is a complete message which the terminal transmits to the host only if the host has initiated activity from the terminal. A Response Packet consists of at least a Response Packet Header and may also contain one or more Responses. A Response Packet is used to group together, for transmission to the host, all responses which are generated by the terminal.

All requests for status (Send Status or Send Command Packet Buffer Size) and any requests for non-data responses (Send Terminal ID or Send Terminal Type or Send Self-Test Results) will be sent after the terminal finishes processing the Command Packet in which the requests occur.

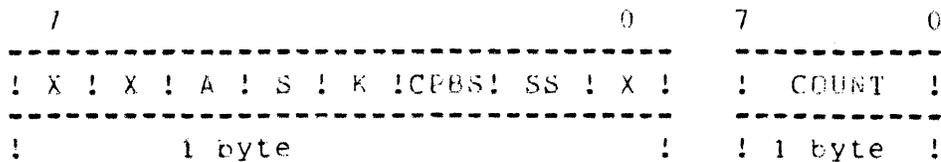
At the present time in the VT62, all requests for status and other non-data responses will be sent in a separate Response Packet from other input.

#### A.9.5 Command Packet

The format of a command packet is as follows:



The format of the command packet header is as follows:



- XX: Two unused bits; must be 0, and are ignored by the VT62.
- A: 0 – Ignore “S” bit; leave keyboard in present state. 1 – Set keyboard to state specified by “S” bit
- S: 0 – Unlock the keyboard (see “A” bit definition)  
1 – Lock the keyboard (see “A” bit definition)
- K: Kill input bit. When 1, the terminal will cancel a pending input request.
- CPBS: Command Packet Buffer Size. 1 indicates send Command Packet Buffer size.
- SS: Send Status. When 1, causes the terminal to send back a Response Packet (defined in Section 8.6), thus transmitting its current status to the host processor.
- X: One unused bit; must be 0 and is ignored by the VT62.
- COUNT: A 7- or 14-bit counter, formatted as described in F.9.1, which counts the number of bytes in the following sequence of commands.

### A.9.6 Response Packet

A response packet is a packet which flows from the terminal to the host.

The format of a response packet is as follows:

```
-----  
!   PACKET HEADER   !           SEQUENCE OF RESPONSES           !  
-----
```

The format of the response packet header is as follows:

```
          7                               0  
-----  
! X ! I ! L ! HC ! HCE ! CE ! CPBS ! X !  
-----  
!           1 BYTE           !  
  
          7       0 7       0  
-----  
!   CEBO   !  
-----  
!   2 BYTES !  
  
          7       0 7       0  
-----  
!   CPBSZ   !  
-----  
!   2 BYTES !  
  
          7       0 7       0  
-----  
!   COUNT   !  
-----  
!   2 BYTES !
```

- X:** An unused bit; the terminal sends a 0.
- I:** Input request bit. 1 indicates an input request is pending, and 0 indicates that no input request is pending.
- L:** Keyboard locked bit. 1 indicates that the keyboard is locked and 0 indicates that the keyboard is unlocked.
- HC:** Hard copy bit. 1 indicates that output to a printer is in progress and 0 indicates that output to a printer is not in progress.
- HCE:** Hard copy error bit. 1 indicates that a printer error occurred in the last hard copy operation.
- CE:** Command Error bit. 1 indicates that a command error occurred and that a Command Error Byte Offset follows; a 0 indicates that all commands processed since the last Response Packet was sent have been successfully completed.

CPBS:	1 indicates that the Command Packet buffer size follows.
X:	Unused bit; the terminal ends a 0.
CEBO:	A 16-bit binary number, least significant bits first, which is the number of bytes of offset from the beginning of the command packet where the command error occurred.
CPBSZ:	Size of Command Packet buffer: 16-bit binary number, least significant bits first, of size of terminal Command Packet buffer, in bytes.  The size of the VT62 Command Packet buffer is 2000 (decimal) bytes.
COUNT:	A 7- or 14-bit counter, formatted as described in A.9.1, which counts the number of bytes in the following sequence of responses.

## A.9.7 Summary of Commands

### A.9.7.1 Definitions

- D1. <RESET AND DEFINE SCREEN> <COUNT> <DISP END>
- D2. <RESET FORM DEFINITIONS AND DEFINE FORM FIELDS> <COUNT> <R1> <C1> <L1> <VA1> <FA1> . . . <Rn> <Cn> <Ln> <VAn> <FAn>
- D3. <RESET MENU DEFINITIONS AND DEFINE MENU FIELDS> <COUNT> <MA> <R1> <C1> <L1> <VA1> . . . <Rn> <Cn> <Ln> <VAn>
- D4. <DEFINE ERROR MESSAGE> <COUNT> <MSG #> <VA> <STRING>

### A.9.7.2 Area Manipulation

#### A.9.7.2.1 Display Area Manipulation

- A1. <ERASE DISPLAY AREA> <COUNT>
- A2. <WRITE DISPLAY AREA> <COUNT> <VA> <STRING>
- A3. <ERASE DISPLAY AREA THEN WRITE AREA> <COUNT> <VA> <STRING>
- A4. <WRITE DISPLAY SUB-AREAS> <COUNT> <R1> <C1> <L1> <VA1> <STRING>  
. . . <Rn> <Cn> <Ln> <VAn> <STRING>
- A5. <ERASE DISPLAY AREA THEN WRITE SUB-AREAS> <COUNT> <R1> <C1> <L1> <VA1> <STRING> . . . <Rn> <Cn> <Ln> <VAn> <STRING>
- A6. <SCROLL DISPLAY> <COUNT>

### A.9.7.2.2 Form Area Manipulation

- A7. <ERASE FORM AREA> <COUNT>
- A8. <WRITE FORM AREA> <COUNT> <VA> <STRING>
- A9. <ERASE FORM AREA THEN WRITE AREA> <COUNT> <VA> <STRING>
- A10. <WRITE FORM SUB-AREAS> <COUNT> <R1> <C1> <L1> <VA1>  
<STRING>  
... <Rn> <Cn> <Ln> <VAn> <STRING>
- A11. <ERASE FORM AREA THEN WRITE SUB-AREAS> <COUNT> <R1> <C1>  
<L1> <VA1>  
<STRING> ... <Rn> <Cn> <Ln> <VAn> <STRING>

### A.9.7.3 Error Line Manipulation

- L1. <ERASE ERROR LINE> <COUNT>
- L2. <WRITE ERROR LINE> <COUNT> <VA> <STRING>
- L3. <WRITE NUMBERED ERROR MESSAGE> <COUNT> <MSG #>

### A.9.7.4 Field Manipulation

#### A.9.7.4.1 Form Field Manipulation

- F1. <CLEAR FORM FIELDS> <COUNT> <FIELD #> ... <FIELD #>
- F2. <CLEAR THEN WRITE FORM FIELDS> <COUNT> <FIELD #> <L>
- F3. <WRITE ALL FORM FIELDS> <COUNT> <STRING>

#### A.9.7.4.2 Menu Field Manipulation

- F4. <SELECT MENU FIELDS> <COUNT FIELD #> ... <FIELD #>
- F5. <DESELECT MENU FIELDS> <COUNT FIELD #> ... <FIELD #>
- F6. <PROMPT FIELD> <COUNT> <AREA> <FIELD #>

### A.9.7.5 Input Requests

- I1. <REQUEST INPUT> <COUNT> <FLAGS> <FCN KEY #0> ... <FCN KEY #N>
- I2. <RESUME INPUT> <COUNT>
- I3. <RETRANSMIT> <COUNT>
- I4. <READ IDENTIFICATION> <COUNT>
- I5. <SIMULATE FORMS ACTIVITY> <COUNT> <FNC KEY #> <MOD FLD #>  
... <MOD FLD #>
- I6. <SIMULATE MENU ACTIVITY> <COUNT> <FCN KEY #> <FIELD #>  
... <FIELD #>

### A.9.7.6 Kills

There are no commands in this category. Refer to format of Command Packet Header.

### A.9.7.7 Hard Copy Control

- 1. <PRINT DISPLAY LINES> <COUNT> <L#> <L#> ... <L#>
- 2. <SEND TEXT DIRECTLY TO PRINTER> <COUNT> <TEXT>

### A.9.7.8 Diagnostics

1. <RUN SELF TEST> <COUNT>
2. <SEND RESULTS OF SELF TEST> <COUNT>

### A.9.7.9 Control

1. <RING THE BELL> <COUNT> <TIME>
2. <WRITE IDENTIFICATION> <COUNT> <STRING>
3. <SEND TERMINAL TYPE> <COUNT>
4. <GENERAL RESET> <COUNT>
5. <WAIT FOR TRANSMISSION TO HOST> <COUNT>
6. There are two command codes which must not be used. (Refer to Section A.9.10.9)
7. All Command Codes not otherwise defined are treated as No-Ops.

### A.9.8 Summary of Responses

#### Form Responses

1. <ALL FORM FIELDS> <COUNT> <FCN KEY #N> <ID> <STRING>
2. <MODIFIED FORM FIELDS> <COUNT> <FCN KEY #N> <ID> <FIELD #>  
< L> <STRING> . . . <FIELD #> <STRING>
3. <AUTO-TRANSMIT FORM FIELD> <COUNT> <FIELD #> <L> <STRING>

#### Menu Responses

1. <MENU FIELD NUMBERS> <COUNT> <FCN KEY #N> <ID> <FIELD #>  
. . . <FIELD #>
2. <MENU FIELD CONTENTS> <COUNT> <FCN KEY # N> <ID> <FIELD #>  
< L> <STRING> . . . <FIELD #> <L> <STRING>

#### Control Response

1. <IDENTIFICATION FIELD> <COUNT> <STRING>
2. <TERMINAL TYPE> <COUNT> <TERMINAL DEVICES>
3. <CURRENT STATUS>

#### Function Key Response

1. <FUNCTION KEY> <COUNT> <FCN KEY #> <AREA> <FIELD #>

#### Diagnostic Responses

1. <SELF TEST RESULTS> <COUNT> <DATA>

### A.9.9 Common Notations

The following notations are common to the following detailed command descriptions. In what follows, all counting starts at 1 and all numbers are decimal, unless explicitly stated to the contrary.

#### COMMAND CODE

A byte, binary encoded, in the range 1–255. It selects a particular command out of 255 possible.

#### <COUNT>

Counts the number of bytes in the following string. This is either a 7-bit or a 14-bit binary encoded counter, formatted exactly as specified in Section A.9.1.

<MA>

The Menu Attributes associated with the group of all menu field defined. This is a single byte, bit encoded as described below:

```

7           0
-----
!XXXXXX!ML!
-----

```

XXXXXX:

Six bits which must be zero and which are ignored by the VT62.

ML:

Two bits which are encoded as follows:

00 - No restriction on number of menu items selected

01 - At least one menu item must be selected

10 - No more than one menu item may be selected

11 - One and only one menu item must be selected

<R> (or <Ri>)

Denotes a row of either the display area or the form area. It is a one byte field, binary encoded, in the range 1 through 24. It selects a row relative to the start of an area; e.g., the 6th row of the form area.

<C> (or <Ci>)

Denotes a column of a row. It is a one byte field, binary encoded, in the range 1 through 80.

<L> (or <Li>)

Counts the number of characters in either a following string or the length of a field in a form definition. This is either a 7-bit or 14-bit binary encoded counter, formatted exactly as is a count field, discussed in Section A.9.1.

<VA> (or <VAi>)

The video attributes associated with either a following string or the particular form field, the definition of which it is a part. This is a single byte, bit encoded described as follows.

```

-----
!X!RV!XXXX!11!
-----

```

X:

Reserved for future expansion; must be sent as 0 and ignored by the terminal.

RV:

Reverse video. If RV=1, then the associated string or field definition will have the reversed video (black-on-white) attribute.

XXXX:

Reserved for future expansion; must be sent as 0 and ignored by the terminal.

II: Intensity field. Two of four possible intensities are defined. II=00 means normal intensity and II=01 means invisible. II=10 and II=11 are undefined and will result in normal intensity on the VT62.

**RESTRICTION**

**There can be only one invisible field and it is limited to 40 or less characters. If more than one invisible field is defined, the first field so defined will be invisible and the others will be visible. Truncation occurs if an attempt is made to store more than 40 characters.**

<FA> (or <FAi>) Describes the field attributes of the particular field, the definition of which this is a part. This is a 24-bit field, encoded as described below:

```

7          0          FIRST BYTE
-----
!X!CCCCCCC!
-----

```

```

7          0          SECOND BYTE
-----
!AI!AX!F!E!DDDD!
-----

```

```

7          0          THIRD BYTE
-----
!XX!M!RJ!XXXX!
-----

```

CCCCCCC: Clear Character – the 7-bit ASCII encoded character to which the field is initialized and cleared.

X: An unused bit; must be a 0 and ignored by the terminal.

DDDD: Data type descriptor (Section A.2.4)  
 0000: = Alphabetic  
 0001: = Alphanumeric  
 0010: = Numeric  
 0011: = Signed Numeric  
 0100: = Any displayable character  
 0101

. Reserved for future expansion

1111

Data type checking is only done on characters entered from the keyboard. Data sent from the host is never validity checked. Clear characters are never validity checked.

Empty field. 1 indicates that this field cannot be left empty.

Full or empty field. 1 indicates that this must be completely filled or be empty (contain all clear characters).

Note that Empty/Full checking is done on fields when:

1. they are exited in the forward direction;
2. a function key is pressed which has been defined to transmit forms.

The operation performed when the Backward Field key is pressed is performed without a Full/Empty error occurring.

**AX:** Auto-transmit. 1 indicates that this is an auto-transmit field.

**AT:** Auto-tab. 1 indicates that this is an auto-tab field.

**XXXX:** Four unused bits; must be zeros and ignored by the terminal.

**RJ:** Right justified. 1 indicates that this field is right-justified during entry. 0 indicates that this is a left-justified field.

**M:** Protected field. 1 indicates that this field cannot be modified from the keyboard. The operator cannot position the cursor into a protected field.

**XX:** Two unused bits; must be zeros and are ignored by the terminal.

## **A.9.10 Description of Commands**

### **A.9.10.1 Definitions**

These commands define the screen format, menu definitions, form definitions and error message definitions. These commands do not cause data to be written to the screen. Instead, the information in these commands is stored by the terminal and used as context for the execution of other commands and for operation of the keyboard.

**D1. <RESET AND DEFINE SCREEN> <COUNT> <DISP END>**

**COMMAND CODE:** 1

**COUNT:** Number of bytes of data in the command (=1)

**DISP END:** Line at which the Display Area ends. A byte, binary encoded, in the range 0 to 24. 0 means no Display Area.

**Description:** All previous menu definitions and form definitions are cleared. The screen buffer is cleared. The form area starts at the line after the display area ends. Error message definitions are not cleared. The keyboard is locked. Any pending input request is aborted.

D2. <RESET FORM DEFINITIONS AND DEFINE FORM FIELDS> <COUNT>  
<R1> <C1> <L1> <VA1> <FA1> . . . <Rn> <Cn> <Ln> <VAn> <FAn>

COMMAND CODE:

2

COUNT:

Number of bytes of data in the command, binary encoded. If count = 0 then form definitions are only cleared; no new form definitions are stored.

Ri:

Row within Form Area in which form field begins.

Ci:

Column at which form field begins.

Li:

Length of form field. Range: 1-1920.

VAi:

Video attributes to form field.

FAi:

Field attributes to form field.

Description:

All previous form field definitions are cleared. The form field definitions in this command are stored. The form field definitions define the areas into which the keyboard operator may position the cursor when an input request pertaining to the forms area is accepted; they define the video characteristics of data entered and the validation criteria to be applied to data typed into the fields.

Note:

Execution of this command does NOT clear defined form fields to the clear character but leaves in the fields whatever data was previously stored.

Errors:

A <RESET AND DEFINE SCREEN> command must have been accepted by the terminal sometime prior to issuance of this command, otherwise the command will be rejected.

An input request may not be pending when this command is received, otherwise this command will be rejected.

D3. <RESET MENU DEFINITIONS AND DEFINE MENU FIELDS> <COUNT> <MA>  
<R1> <C1> <L1> <VA1> . . . <Rn> <Cn> <Ln> <VAn>

COMMAND CODE:

3

COUNT:

Number of bytes of data in the command, binary encoded. If count = 0 then menu definitions are cleared; no new menu definitions are stored.

MA:

Menu attributes.

Ri:

Row within Menu Area in which menu field begins.

**Ci:** Column at which menu field begins.

**Li:** Length of menu field. Range: 1-1920.

**Description:** All previous menu field definitions are cleared. The menu field definitions in this command are stored. The menu field definitions define the areas such that if the cursor is positioned over a menu field and the <Select> key is typed, a menu selection is recorded and all characters in the menu are forced into reverse video. If the cursor is positioned over a menu field and the <Deselect> key is typed, the menu selection is deselected and all characters in the menu are forced into normal video. If a function key which has been appropriately defined is pressed, all selected fields are transmitted to the host.

**Errors:** A <RESET AND DEFINE SCREEN> command must have been accepted by the terminal sometime prior to the issuance of this command, otherwise the command will be rejected.

An input request may not be pending when this command is received, otherwise this command will be rejected.

An attempt to define a field that extends beyond the Display Area, or that overlaps another field definition already accepted will cause the command to be rejected.

**D4. <DEFINE ERROR MESSAGE> <COUNT> <MSG #> <VA> <STRING>**

**COMMAND CODE:** 4

**COUNT:** Number of bytes of data in the command. If count = 0, then the terminal assumes no textual error indications are present. In this case, line 24 can be used for other purposes.

**MSG #:** Number by which this message is to be referred. A byte, binary encoded, in the range 1 to 32.

**VA:** Video attributes of the following string.

**STRING:** Text to be written on the Error Line when the host or the terminal requests a message be written using the MSG #.

**Description:** This definition supersedes any previous definition of the message denoted by MSG #. This definition remains in effect until explicitly redefined by another <DEFINE ERROR MESSAGE> command or a <GENERAL RESET> command is executed. A <RESET AND DEFINE SCREEN> command does not clear this message definition.

**Errors:** Text must be 40 characters or less or truncation will occur.

Example:

```
<DEFINE ERROR MESSAGE> <14> <2> <0> <NUMBERS ONLY>
```

This causes the string "NUMBERS ONLY" to be written on the Error Line when either the host or the terminal requests MSG #2 be written.

Message numbers 1 through 16, although defined and usable by the host computer, are used by the terminal as indicated below:

Message #	Purpose
1	Illegal entry in alphabetic field
2	Illegal entry in alphanumeric field
3	Illegal entry in numeric field
4	Illegal entry in signed numeric field
5	Field cannot be left empty
6	Field must be full or empty
7	Not enough menu selections
8	Too many menu selections
9-16	Reserved for future expansion

Note that these messages are only present if loaded by a <DEFINE ERROR MESSAGE> command. These messages are cleared to all nulls by a <GENERAL RESET> command or by a power up sequence or by Self Test. A DDCMP START does NOT clear the error messages.

The text of Message Numbers 1 through 8 that are loaded by the host do not have to be the same as the messages given in the table above but do have to follow the restriction of being 40 characters or less in length.

Message numbers 17 through 32 are defined by and reserved for the exclusive use of the host computer.

The following error messages will be posted if the operator attempts to exit a field with a violation of Form Field Attributes C6 and/or C7:

C6, C7	Field Is:	Error Message #
0, 1 partly filled		6
1, 0 empty		5
1, 1 empty		5
1, 1 partly filled		6

#### A.9.10.2 Area Manipulation

These commands cause either the Display Area or the Form Area to be manipulated. There exist pure clear commands (those which do not write following the clear), pure write commands (those which do not clear prior to the write), and clear/write commands, which clear the area and then write text.

Two methods of writing are available. One method is an area write, which causes a single contiguous string of characters to be written, beginning at the home (1, 1) position of the area. The other method allows a sequence of discontinuous strings to be written, each string beginning at a directly addressed (row, column) position in the area.

When text is written, the cursor position is not affected.

All of these commands will be rejected if the specified area has not been previously defined by a <RESET AND DEFINE SCREEN> command.

Note that writing into line 24 with commands for writing Display/Menu Area or Forms Area does not cause the terminal to enter error mode.

#### **A.9.10.2.1 Display Area Manipulation**

##### **A1. <ERASE DISPLAY AREA> <COUNT>**

**COMMAND CODE:** 5

**COUNT:** Number of bytes of data in the command (=0).

**Description:** The Display Area is cleared to NULs.

**Errors:** None

##### **A2. <WRITE DISPLAY AREA> <COUNT> <VA> <STRING>**

**COMMAND CODE:** 6

**COUNT:** Number of bytes of data in the command.

**VA:** Video attributes of following <STRING>.

**STRING:** Text which is written to the Display Area.

**Description:** The text is written to the Display Area, beginning at (row, column) position (1, 1). If the amount of text exceeds the size of the Display Area, the excess text will be discarded without error notification.

**Errors:** None

##### **A3. <ERASE DISPLAY AREA THEN WRITE AREA> <COUNT> <VA> <STRING>**

**COMMAND CODE:** 7

**COUNT:** Number of bytes of data in the command.

**VA:** Video attributes of following <STRING>.

**STRING:** Text which is written on the Display Area.

**Description:** Same as <WRITE DISPLAY AREA> command, except that the Display Area is cleared to NULs before the write operation.

**Errors:** None

A4. <WRITE DISPLAY SUB-AREAS> <COUNT> <R1> <C1> <L1> <VA1> <STRING>  
... <Rn> <Cn> <Ln> <VAn> <STRING>

COMMAND CODE: 8

COUNT: Number of bytes of data in the command.

Ri: Row within Display Area on which <STRING> begins.

Ci: Column at which <STRING> begins.

Li: Length of <STRING>. Range: 1-1920.

VAi: Video attributes of following <STRING>.

STRING: Text which is written on Display Area.

Description: A sub-area is defined by an <R> <C> <L> sequence. Each <STRING> is written according to the <R> <C> <L> sub-area specification which precedes the <STRING>.

If a sub-area extends past the end of the Display Area, the excess text will be discarded.

Note: A sub-area has no "line boundaries." If the <LENGTH> for a sub-area does not expire at column 80 of a line, the sub-area continues at column 1 of the next line.

A5. <ERASE DISPLAY AREA THEN WRITE SUB-AREAS> <COUNT> <R1> <C1> <L1>  
<VA1> <STRING> ... <Rn> <Cn> <Ln> <VAn> <STRING>

COMMAND CODE: 9

COUNT: Number of bytes of data in the command.

Ri: Row within Display Area on which <STRING> begins.

Ci: Column at which <STRING> begins.

Li: Length of <STRING>. Range: 1-1920.

VAi: Video attributes of following <STRING>.

STRING: Text which is written on Display Area.

Description: Same as <WRITE DISPLAY SUB-AREAS> command, except that the Display Area is cleared to NULs before the write operation.

A6. <SCROLL DISPLAY> <COUNT>

COMMAND CODE: 10

COUNT: Number of bytes of data in the command (=0).

Description: The display area is scrolled up one line. The new bottom line of the display is filled with NULs. The old top line of the display is lost.

**A.9.10.2.2 Form Area Manipulation**

Commands A7 to A11 operate on the Form Area. Except for the fact that they operate on the Form Area, these commands are identical in operation to their Display Area Counterparts.

A7. <ERASE FORM AREA> <COUNT>

COMMAND CODE: 11

A8. <WRITE FORM AREA> <COUNT> <VA> <STRING>

COMMAND CODE: 12

A9. <ERASE FORM AREA THEN WRITE AREA> <COUNT> <VA> <STRING>

COMMAND CODE: 13

A10. <WRITE FORM SUB-AREAS> <COUNT> <R1> <C1> <L1> <VA1> <STRING> ...  
<Rn> <Cn> <Ln> <VAn> <STRING>

COMMAND CODE: 14

A11. <ERASE FORM AREA THEN WRITE SUB-AREAS> <COUNT> <R1> <C1> <L1>  
<VA1> <STRING> ... <Rn> <Cn> <Ln> <VAn> <STRING>

COMMAND CODE: 15

**A.9.10.3 Error Line Manipulation**

These commands cause a line of text to be written to the Error Line, thus putting the terminal in error mode, or clear the Error Line, thus removing the terminal from error mode. (Error mode is discussed in Section A.10.2 – refer to the description of the error reset key.)

The cursor position is unaffected by these commands.

L1. <ERASE ERROR LINE> <COUNT>

COMMAND CODE: 16

COUNT: Number of bytes of data in the command (=0).

Description: The Error Line is cleared to all nulls, and the terminal is removed from error mode. The operator may also clear the error line and remove the terminal from error mode by typing the error reset key.

Errors: None.

L2. <WRITE ERROR LINE> <COUNT> <VA> <STRING>

COMMAND CODE:	17
COUNT:	Number of bytes of data in the command.
VA:	Video attributes of following <STRING>.
STRING:	Text which is written on the Error Line.
Description:	The Error Line is cleared to all nulls and then the text is written on that line. Text in excess of 80 characters will be discarded without error notification. The terminal enters error mode.
Errors:	None

L3. <WRITE NUMBERED ERROR MESSAGE> <COUNT> <MSG #>

COMMAND CODE:	18
COUNT:	Number of bytes of data in the command. If count = 0, then the terminal enters error mode without disturbing line 24.
MSG #:	A byte, binary encoded, in the range 1 to 32, which selects one of the previously defined error messages.
Description:	If count=1, the Error Line is cleared to all nulls and then the error message with number "MSG #" is written on the Error Line. The terminal enters error mode.
Errors:	If the MSG # is over 32 or if it does not correspond to a message number which has been previously defined, no message will be printed but the terminal will enter Error Mode.

#### A.9.10.4 Field Manipulation

These commands cause either menu fields or form fields to be manipulated by reference to field numbers. Fields are numbered beginning with 1, as they were sequentially defined by the <CLEAR FORM DEFINITIONS AND DEFINE FORM FIELDS> command or by the <CLEAR MENU DEFINITIONS AND DEFINE MENU FIELDS> command.

##### A.9.10.4.1 Form Field Manipulation

F1. < CLEAR FORM FIELDS> <COUNT> <FIELD #> . . . <FIELD #>

COMMAND CODE:	19
COUNT:	Number of bytes of data in the command.
FIELD #:	A byte, binary encoded in the range 0 to 126, which selects fields by number. 0 indicates all fields.
Description:	All indicated fields are cleared to their clear characters as specified in the field definitions. All modified field flags are reset to the unmodified state.

**Errors:** Both a <RESET AND DEFINE SCREEN> and a <CLEAR FORM DEFINITIONS AND DEFINE FORM FIELDS> command must have been received by the terminal before this command may be executed. All fields referenced by this command must have been previously defined, otherwise an error will result, in which case the terminal will ignore this command.

**F2. <CLEAR THEN WRITE FORM FIELDS> <COUNT> <FIELD> <L> <STRING> ... <FIELD #> <L> <STRING>**

**COMMAND CODE:** 20

**COUNT:** Number of bytes of data in the command.

**FIELD #:** A byte, binary encoded, in the range 1 to 126.

**L:** Length of STRING.

**STRING:** Text which is written in the field.

**Description:** All indicated fields are cleared to their clear characters as specified in the field definitions. All modified field flags are reset to the unmodified state.

Each <STRING> is written into the field designated by <FIELD #>, using the associated field video attribute, and justified according to the field attributes. <L> characters are written, unless <L> exceeds the size of the field, in which case all excess characters will be discarded without error notification.

**Errors:** Both a <RESET AND DEFINE SCREEN> and a <CLEAR FORM DEFINITIONS AND DEFINE FORM FIELDS> command must have been received by the terminal sometime prior to the time that this command is executed. All fields referenced by this command must have been previously defined, otherwise an error will result, in which case the terminal will ignore (the erroneous part of) this command.

**F3. <WRITE ALL FORM FIELDS> <COUNT> <STRING>**

**COMMAND CODE:** 21

**COUNT:** Number of bytes of data in the command, binary encoded.

**STRING:** Text which is written within the form fields.

**Description:** The <STRING> is written to the form fields, beginning with the first defined field and ending with the last defined field. As each field is filled, the next character of <STRING> is written to the first character position of the next defined field until <STRING> has expired. <STRING> appears in the video attribute associated with the various field definitions. If <STRING> exceeds the total number of character positions in all fields, the excess text will be discarded.

Errors: Both a <RESET AND DEFINE SCREEN> and a <CLEAR FORM DEFINITIONS AND DEFINE FORM FIELDS> command must have been received by the terminal sometime prior to the time that this command is executed or an error will result, in which case the terminal will ignore this command.

#### A.9.10.4.2 Menu Field Manipulations

F4. <SELECT MENU FIELDS> <COUNT> <FIELD #> . . . <FIELD #>

COMMAND CODE: 22

COUNT: Number of bytes of data in the command.

FIELD #: A byte, binary encoded in the range 0 to 126, which selects specific menu fields by number. 0 indicates all fields.

Description: All indicated fields are set to their selected state.

Errors: Both a <RESET AND DEFINE SCREEN> and a <CLEAR MENU DEFINITIONS AND DEFINE MENU FIELDS> command must have been received by the terminal sometime prior to the time that this command is executed. All fields referenced by this command must have been previously defined, otherwise an error will result, in which case the terminal will ignore (the erroneous part of) this command.

F5. <DESELECT MENU FIELDS> <COUNT> <FIELD #> . . . <FIELD #>

COMMAND CODE: 23

COUNT: Number of bytes of data in the command.

FIELD: A byte, binary encoded in the range 0 to 126, which selects specific menu fields by number. 0 indicates all fields.

Description: All indicated fields are set to their deselected state.

Errors: Both a <RESET AND DEFINE SCREEN> and a <CLEAR MENU DEFINITIONS AND DEFINE MENU FIELDS> command must have been received by the terminal sometime prior to the time that this command is executed. All fields referenced by this command must have been previously defined, otherwise an error will result, in which case the terminal will ignore (the erroneous part of) this command.

F6. <PROMPT FIELD> <COUNT> <AREA> <FIELD #>

COMMAND CODE: 24

COUNT: Number of bytes of data in the command (=2).

AREA:	A byte indicating the area to be prompted, binary encoded as: 0 = Display Area 1 = Form Area
FIELD #:	A byte, binary encoded, in the range 1 to 126.
Description:	<p>A &lt;PROMPT FIELD&gt; command will be treated as a &lt;NO-OP&gt; unless the terminal is in a state to accept a &lt;REQUEST INPUT&gt; or a &lt;RESUME INPUT&gt; command.</p> <p>The terminal stores the prompt information of the last &lt;PROMPT FIELD&gt; command issued.</p> <p>Both the &lt;REQUEST INPUT&gt; and &lt;RESUME INPUT&gt; commands, when executed, will initiate a prompt based on prompt information which has been stored from a &lt;PROMPT FIELD&gt; command from a previous Command Packet or from a preceding command in the current Command Packet so long as the prompt information has not been cleared. If no prompt information exists, the terminal will prompt at the first form field if form fields are enabled. If form fields are not enabled, the terminal will prompt at the first menu field. If neither form fields or menu fields are enabled, the terminal will prompt in the lower right hand corner of the screen and no data input or cursor movement function will be allowed.</p> <p>When the keyboard is unlocked by an &lt;INPUT REQUEST&gt; or a &lt;RESUME INPUT&gt; command, the cursor will be positioned to the first character position of the &lt;FIELD&gt; located in the &lt;AREA&gt; specified in the &lt;PROMPT FIELD&gt; command.</p> <p>When the terminal does the prompt, the prompt information is cleared and a new &lt;PROMPT FIELD&gt; command must be issued in order for another prompt to be initiated.</p> <p>The prompt will also be cleared by a &lt;RESET AND DEFINE SCREEN&gt; command or by a &lt;GENERAL RESET&gt; or by Self-Test.</p>

#### **A.9.10.5 Input Requests**

These commands enable keyboard operations to be performed at the terminal. The default keyboard state is locked. The keyboard is unlocked when the terminal executes a <REQUEST INPUT> or <RESUME INPUT> command. The keyboard is locked if the terminal operator hits a function key which has been defined. When the keyboard is locked, the corresponding LED is lit, and the flashing cursor disappears.

11. <REQUEST INPUT> <COUNT> <FLAGS> <FCN KEY DEF> ... <FCN KEY DEF>

COMMAND CODE: 25

COUNT: Number of bytes of data in the command.

FLAGS: One byte, bit encoded as follows:

bit 0:

0 - If forms enabled, transmit forms packed (modified form fields only)

1 - If forms enabled, transmit forms unpacked (all form fields)

bit 1:

0 - If menu mode enabled, transmit only menu field numbers

1 - If menu mode enabled, transmit menu field numbers and contents

bit 2:

0 - Auto-Transmit Mode disabled

1 - Auto-Transmit Mode enabled

bit 3:

0 - Keypad enabled in numeric mode - all numeric keys and <.> will attempt to enter decimal numbers. Other keys are unaffected by this bit.

1 - Keypad enabled in function mode - numeric keys and <.> will generate <PF0 to PF6>, <ABORT>, and <PF8> to <PF10> responses if enabled. Other keys are unaffected by this bit.

bit 4:

0 - Do not send 4 character ID field

1 - Send 4 character ID field.

bit 5:

0 - Don't clear modified bit flags

1 - Clear modified bit flags

bit 6:

Unassigned. Must be set to 0 for the VT62 terminal.

bit 7:

0 - No additional flag byte follows.

1 - Additional flag byte follows (only one additional allowed). The VT62 ignores the second flag byte.

**FUNCTION KEY DEF:**

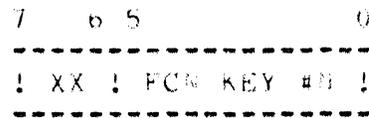
These are a sequence of bytes which indicate if function keys are enabled and in what mode.

If a function key is not defined by one of these bytes, nothing will happen if the key is pressed by the operator except a key click.

Menu fields cannot be enabled even if defined unless at least one function key is enabled to send menu fields. Form fields cannot be enabled even if defined unless at least one function key is enabled to send form fields.

If at least one function key is not enabled to send form fields, then form fields cannot be enabled even if defined.

The format of a function key byte is as follows:



**XX:**

- 00 – Function key response
- 01 – Send form
- 10 – Send menu
- 11 – Send form and menu

**FCN KEY #N:**

This is a six bit octal value. Keys which may be defined are:

- 0 to 6 – PF0 to PF6
- 7 – abort (PF7)
- 8 to 10 – PF8 to PF10
- 41 – enter/PF41
- 42 – affirm/PF42
- 43 – close/PF43
- 44 – stop repeat/PF44
- 45 – (blank key)/PF45

**Description:**

This command does six basic operations:

1. sets input modes
2. sets keypad mode
3. enables function keys
4. prompts a field
5. resets all modified field flags to modified
6. unlocks the keyboard

Description: (Cont)

Once the keyboard is unlocked, an input request becomes pending and remains pending until suspended or terminated by an appropriately defined function key being typed, or until cancelled by the host. If the function key that is typed has been defined to send a function key response, the keyboard is locked and the input request is suspended; the input request may be resumed by a <RESUME INPUT> command. If the function key that is typed has been defined to send form and/or menu fields, the keyboard is locked and the input request is terminated.

The execution of the <REQUEST INPUT> command completes when the last operation above is completed (i.e., the keyboard is locked). At this point the packet may be acknowledged by the terminal returning a packet completion code. The input request condition remains pending until either the <next> or the <enter> key is typed, or until cancelled by the host.

The input modes are <FORM MODE>, <MENU MODE> and <AUTO-TRANSMIT MODE>. <FORM MODE> and <MENU MODE> may simultaneously be enabled. <AUTO-TRANSMIT MODE> will be active only if <FORM MODE> is enabled.

If both <FORM MODE> and <MENU MODE> are enabled the cursor may be freely moved between the Display Area and the Form Area by means of the <HOME FORM/HOME DISP> key. When an appropriately defined function key is finally typed, a form response and/or a menu response will be generated according to the definition for the particular function key.

If <AUTO-TRANSMIT MODE> is enabled, an auto-transmit response is generated whenever the terminal operator tabs out of an auto-transmit field in a forward direction; menu selections are not sent at this time and only the auto-transmit form field is sent. At this point the keyboard is locked and the input request is suspended. Keyboard operations may be resumed by means of the <RESUME INPUT> command.

Upon commencing execution of the <REQUEST INPUT> command, the terminal clears previous function key definitions and then selectively enables the function keys defined by the current command. If form fields (or menu fields) are not defined and a function key is typed which has been defined to send form fields (or menu fields), an appropriate response will be generated with a count=0.

Errors: Only a single <REQUEST INPUT> or <RESUME INPUT> command may occur in a packet sent to the terminal.

If a <REQUEST INPUT> command is received while an input request is pending, that <REQUEST INPUT> command will be ignored.

## 12. <RESUME INPUT> <COUNT>

COMMAND CODE: 26

COUNT: Number of bytes of data in the command (=0).

Description: This command is used by the host when a function key which has been defined to send a function key response is typed or when an auto-transmit field causes a response to be returned to the host and the keyboard is therefore locked with the input request suspended. The host may "resume" the input operation at the terminal by simply unlocking the keyboard with the <RESUME INPUT> command for further operations.

The set of function keys enabled by the <REQUEST INPUT> command is not altered in any way by the <RESUME INPUT> command.

If prompt information is stored when this command is executed, the cursor will be positioned to the first character position of the field in the area specified by the prompt information. (Refer to Section A.9.10.4 command F6).

Errors: If the keyboard is already unlocked, then this command is ignored.

## 13. <RETRANSMIT> <COUNT>

COMMAND CODE: 27

COUNT: Number of bytes of data in the command (=0).

Description: This command is used in those instances where the host "loses" a transmission for some reason external to DDCMP. The terminal will retransmit the entire message just sent; the keyboard will remain locked.

Errors: If there have been no transmissions since the last <REQUEST INPUT> or if the keyboard is unlocked, then the terminal ignores this command.

## 14. <READ IDENTIFICATION> <COUNT>

COMMAND CODE: 28

COUNT: Number of bytes of data in the command (=0).

Description: This command causes the terminal to send exactly four bytes of identification to the host. This is the four byte field which the host may write using the <WRITE IDENTIFICATION> command.

Errors: If the <WRITE IDENTIFICATION> command did not send a string of at least 4 bytes, the 4 bytes sent by this command are undefined.

15. <SIMULATE FORMS ACTIVITY> <COUNT> <FCN KEY #> <MOD FIELD #> . . . <MOD FIELD #>

COMMAND CODE: 29

COUNT: Number of bytes of data in the command.

KEY #: Number of keys to be simulated, octal encoded as follows:

- 0 to 6 - PF0 to PF6
- 7 - abort/PF7
- 8 to 10 - PF8 to PF10
- 41 - enter key/PF41
- 42 - affirm key/PF42
- 43 - close key/PF43
- 44 - stop repeat key/PF44
- 45 - (blank key)/PF45

MOD FIELD #: Number of a form field, binary encoded in one byte per field identified.

Description: This is intended as a tool to aid in debugging system software, by simulating operator responses as if the operator had actually typed the key and modified the fields.

16. <SIMULATE MENU ACTIVITY> <COUNT> <FCN KEY #> <FIELD #> . . . <FIELD #>

COMMAND CODE: 30

COUNT: Number of bytes of data in the command.

KEY #: Same as above, refer to Section A.9.10.5, 15.

FIELD #: Number of a menu field, binary encoded in one byte per field identified.

Description: This is intended as a tool to aid in the debugging of system software, by simulating operator responses as if the operator had actually typed the key and selected the menu fields.

#### A.9.10.6 Kills

There are no commands in this category.

### A.9.10.7 Hard Copy Control

The printer is buffered, so that other areas are not tied up during printing operations. The size of the printer buffer is 512 characters.

If a command requiring printing is the next command to be executed and if there are print requests still in process such that there is not enough printer buffer space available without overflowing, the VT62 will not process that command or any other commands until such time as there is sufficient printer buffer space available.

#### NOTE

**It is possible to lock up the terminal if the printer is not available or becomes not available during the execution of printer requests. This would be dangerous in a system environment where it is desirable to never get into a situation where the terminal can be locked up without intervention or query being possible by the host.**

#### 1. <PRINT DISPLAY LINES> <COUNT> <L#> . . . <L#>

COMMAND CODE: 32

COUNT: Number of bytes of data in the command.

L#: A binary encoded byte which specifies a line in the display area. 0 means the entire display in increasing line order.

Description: The specified line is printed on the printer, formatted exactly as it appears on the screen (except that one cannot tell the video attributes by looking at the hard copy image) and terminated with a carriage return, line feed. Upon encountering a L# of 0, the terminal acts as if it has just encountered the string 1 2 . . . n , where n is the last line in the display area.

Errors: If there is no display area, then the terminal ignores this command. If the printer fails (or is off-line, etc.), the terminal will abnormally terminate the output and so inform the host computer at the next status report.

#### 2. <SEND TEXT DIRECTLY TO PRINTER> <COUNT> <TEXT>

COMMAND CODE: 33

COUNT: Number of bytes of text which follow.

TEXT: String of 8-bit bytes.

Description: The text is printed in the order it is received and exactly as it is received, e.g., carriage return line feed pairs must be in the text in the appropriate places. This printing will not interfere with keyboard activity in any way.

Errors: If the printer fails, the terminal will abnormally terminate the output and so inform the host computer at the next status report.

### A.9.10.8 Diagnostics

Loop-back is provided as specified in the current DDCMP and MOP specifications. Briefly:

1. All messages are preceded by a sync sequence. If asynchronous, then send 1 sync character; if synchronous, send 8 sync characters.
2. The DDCMP header is:  
DLE,COUNT, S =1, Q =1 R =0, MSG# =0, ADDR , CRC
3. The data portion is:  
TYPE . . . DATA . . . CRC  
TYPE = loop DATA = any sequence of 8-bit bytes
4. The terminal addressed by <ADDR> will loop back this message exactly as sent (received). The host must ignore the time delay between transmission and reception.
5. To exit loop back testing, the host will initiate the DDCMP start sequence.

#### 1. <RUN SELF TEST> <COUNT>

COMMAND CODE: 40  
COUNT: Number of bytes of text in the command (=0).  
Description: Causes the terminal to run its internal diagnostic.  
Errors: None

#### 2. <SEND RESULTS OF SELF TEST> <COUNT>

COMMAND CODE: 41  
COUNT: Number of bytes of text in the command (=0).  
Description: Invokes a transmission of the results of the most recent self test. Refer to Section A.9.11.5.1.  
Errors: None

### A.9.10.9 Control

#### 1. <RING THE BELL> <COUNT> <TIME>

COMMAND CODE: 50  
COUNT: Number of bytes of data in the command  
Count=0 - Normal short beep is produced:  
Count=1 - The beep is continuous for 7 (<TIME> +1)/60 seconds.  
TIME: An 8-bit binary byte in the range 0 to 255.  
Description: The terminal rings the bell.  
Errors: None

2. <WRITE IDENTIFICATION> <COUNT> <STRING>

COMMAND CODE: 51

COUNT: Number of bytes data in <STRING>

STRING: Any string of four 8-bit bytes.

Description: <STRING> is stored internally in the terminal. Optionally, the terminal will transmit this <STRING> with a response or by itself.

Errors: If string is longer than 4 bytes, all extra bytes will be ignored; no error indication will be made. If the string is shorter than 4 bytes, the response that is returned to the host in an <ID> field of a response is UNDEFINED.

3. <SEND TERMINAL TYPE> <COUNT>

COMMAND CODE: 52

COUNT: Number of bytes of data in the command (=0).

Description: Causes the terminal to transmit its type and local peripheral devices which it can access.

Errors: None

4. <GENERAL RESET> <COUNT>

COMMAND CODE: 53

COUNT: Number of bytes of data in the command (=0).

Description: Causes the terminal to execute its power up logic, after which the DDCMP start up sequence must be initiated: all form fields, menu fields, and error messages must be redefined.

Errors: None

5. <WAIT FOR TRANSMISSION TO HOST> <COUNT>

COMMAND CODE: 35

Description: If input from keyboard is pending, validation and execution of further commands in the present packet or a succeeding packet will stop after execution of this command until the input is completed (or temporarily suspended with keyboard locked) by a transmission to the host.

Description: (Cont)

The use of this command will allow the host to have multiple command packets which request input stacked up to the terminal pending execution without causing the inputs to be overridden.

This is a dangerous command to use in a system environment where it is desirable never to get into a situation where the terminal can be locked up without intervention or query being possible by the host.

Errors:

If keyboard input is not pending, this command is treated as a no-op.

6. The following op codes must not be used. They will cause undefined actions by the terminal.

COMMAND CODE: 31

COMMAND CODE: 34

7. ALL COMMAND CODES NOT OTHERWISE DEFINED ARE TREATED AS NO-OPs.

<NO-OP> <COUNT> <TEXT>

ALL command codes MUST have a COUNT BYTE!

**A.9.11 Description of Responses**

A response is in no way related to a DDCMP block. It is possible that a single response would require several DDCMP blocks or that many responses could be transmitted in a single DDCMP block.

No responses are ever generated or transmitted to the host on an unsolicited basis.

**A.9.11.1 Form Responses**

These responses are generated as a result of the terminal operator entering data into the Form Area and doing either of the following.

- 1. Typing an appropriately defined function key;
- 2. Tabbing out of an auto-transmit field while the terminal is in auto-transmit mode. (The tabbing may occur manually, by means of the forward field key or next field key, or automatically, if the field has the auto-tab attribute set.

1. <ALL FORM FIELDS> <COUNT> <FCN KEY #N> <ID> <STRING>

RESPONSE CODE: 80

COUNT: Number of bytes of data in the response.

FCN KEY #N: A byte in which the upper two bits are zero and the low order six bits represent two octal digits which are the number of the function key that was typed.

ID:	If the <REQUEST INPUT> requested that the ID field be sent, then exactly 4 bytes of identification are sent. This is the four byte field, which the host may write using the <WRITE IDENTIFICATION> command.
STRING:	The complete contents of each form field (including clear characters), sequentially arranged in the same order in which the form fields are defined. This is called unpacked transmission.
Description:	This response is generated by the terminal when an input request is pending and an appropriately defined function key is typed. At this point the keyboard is locked and the input request is terminated.
Errors:	None
2.	<MODIFIED FORM FIELDS> <COUNT> <FCN KEY #N> <ID> <FIELD #> <L> <STRING> . . . <FIELD #> <L> <STRING>
RESPONSE CODE:	80
COUNT:	Number of bytes of data in the response.
FCN KEY #N:	A byte in which the upper two bits are zero and the low order six bits represent two octal digits which are the number of the function key that was typed.
ID:	If the <REQUEST INPUT> requested that the ID field be sent, then exactly 4 bytes of identification are sent. This is the 4-byte field, which the host may write using the <WRITE IDENTIFICATION> command.
FIELD #:	Number of a form field (each field is numbered as it is defined in the form field definitions), binary encoded in one byte.
L:	Length of <STRING>.
STRING:	The contents of the modified form field identified by <FIELD #>. All trailing clear characters are discarded.
Description:	This response is generated by the terminal when an input request is pending and an appropriately defined function key is typed. At this point the keyboard is locked and the input request is terminated.
	The <FIELD #> <L> <STRING> sequences are arranged in the responses in increasing numerical order by <FIELD #>.
	When the terminal operator hits an appropriately defined function key, the contents (with clear characters suppressed) of all form fields whose modified field flags are set are arranged into a response to await terminal selection. This is called packed transmission.

3. <AUTO-TRANSMIT FORM FIELD> <COUNT> <FIELD #> <L> <STRING>

RESPONSE CODE: 82

COUNT: Number of bytes of data in the response.

FIELD #: Number of an auto-transmit field (each field is numbered as it is defined in the form field definitions), binary encoded in one byte.

L: Length of <STRING>.

STRING: The contents of the auto-transmit field identified by <FIELD #>. The transmission is packed or unpacked as specified by the <REQUEST INPUT> command.

Description: This response is generated by the terminal when an input request is pending, the terminal is in auto-transmit mode, and the terminal operator leaves an auto-transmit field (which has passed all of the field validation performed by the terminal) in a forward direction. The tabbing operation is performed by the terminal at this time. The keyboard is locked and the input request is suspended.

The host will then validate the auto-transmitted field according to attributes which generally are an extension of those attributes definable at the terminal.

If the host validation is successful, then the host may simply issue a <RESUME INPUT> command to continue. If the validation fails, the host will generally issue some form of error indication and a new <PROMPT FIELD> command to reposition the cursor back to the start of the auto-transmit field which was in error.

The fact that a field has been auto-transmitted has no effect on the "modified" flag. (Think of the processor as an extension of the terminal with respect to field validation.)

### A.9.11.2 Menu Responses

These responses are generated as a result of the terminal operator making menu selections in the Display Area and then hitting an appropriately defined function key.

1. <MENU FIELD NUMBERS> <COUNT> <FCN KEY #N> <ID> <FIELD #> . . .  
<FIELD #>

RESPONSE CODE: 83

COUNT: Number of bytes of data in the response.

FCN KEY #N: A byte in which the upper two bits are zero and the low order six bits represent two octal digits which are the number of the function key that was typed.

**ID:** If the <REQUEST INPUT> requested that the ID field be sent, then exactly 4 bytes of identification are sent. This is the 4-byte field which the host may write using the <WRITE IDENTIFICATION> command.

**FIELD #:** Number of a menu field (menu fields are numbered as they are defined in the menu field descriptions), binary encoded, one byte per field.

**Description:** This response is generated if an input request for menu input is pending, the terminal operator makes menu selections, and then types an appropriately defined function key.

Each menu selection is made by the terminal operator positioning the cursor over any character position of a menu field and hitting the select key. The menu field will then be displayed in reverse video and tagged as a selected field.

If the cursor is in a position which is defined to be part of more than one menu field, only one menu field will be selected and that menu field will be displayed in reverse video. The <Deselect> key deselects the current menu item and causes it to be displayed in normal video. <Deselect> is the shifted position of the <Select> key.

When an appropriately defined function key is typed, a response containing the numbers of all selected fields (arranged in the same order that they were defined in the menu definitions) will be generated. At this time the keyboard is locked and the input request is terminated.

2. <MENU FIELD CONTENTS> <COUNT> <FCN KEY #N> <ID> <FIELD #> <L> <STRING> . . . <FIELD #> <L> <STRING>

**RESPONSE CODE:** 84

**COUNT:** Number of bytes of data in the response.

**FCN KEY #N:** A byte in which the upper two bits are zero and the low order six bits represent two octal digits which are the number of the function key that was typed.

**ID:** If the <REQUEST INPUT> requested that the ID field be sent, then exactly 4 bytes of identification are sent. This is the 4-byte field which the host may write using the <WRITE IDENTIFICATION> command.

**FIELD #:** Number of a menu field (menu fields are numbered as they are defined in the menu field definitions), binary encoded, one byte per field.

**L:** Length of <STRING>.

**STRING:** The contents of the menu field identified by <+FIELD #>. The host will have previously written any contents by means of a command which writes data to the Display Area.

**Description:** This response is generated by the terminal and transmitted to the host in a manner identical to that of the <MENU FIELD NUMBERS> response, except for one difference. The <MENU FIELD CONTENTS> response contains a <L> <STRING> sequence which specifies the field contents, for each menu field selected.

This response format allows catalogue search operations to be performed, in which case successive pages of the catalogue are written to the Display Area while an input request is pending. The menu field definitions apply to each page written to the Display Area, effectively partitioning each page written according to the boundaries defined for each menu field.

#### NOTE

**Each <REQUEST INPUT> command specifying the menu input option must select either "numbers" or "numbers with contents" which will respectively generate a <MENU FIELD NUMBER> or a <MENU FIELD CONTENTS> response.**

#### A.9.11.3 Control Responses

1. <IDENTIFICATION FIELD> <COUNT> <STRING>

**RESPONSE CODE:** 85

**COUNT:** Number of bytes in <STRING>, (=4).

**STRING:** Exactly four bytes which the host wrote using the <WRITE IDENTIFICATION> command.

**Description:** The 4-byte identification area is sent to the host. The host must have sent a string of at least 4 bytes or the 4 bytes returned by this command are undefined. If the host did not write this area since the last power up, 4 nulls are transmitted.

2. <TERMINAL TYPE> <COUNT> <TERMINAL DEVICES>

**RESPONSE CODE:** 86

**COUNT:** Number of bytes of data in the response, (=2).

**TERMINAL:** A single byte, binary encoded, which identifies the terminal.

1 - VT62.

2 - 255 are reserved for future expansion.

**DEVICES:** A single byte, bit encoded, which lists the local peripheral devices to which the terminal has access.

**XXXXXXXX P**

**XXXXXXXX:** Seven unused bits. The VT62 will transmit zeros; these bits are reserved for future expansion and should be ignored by the host.

**P:** A bit which indicates whether or not the terminal has access to an (LA180-like) impact printer. The presence of such a printer is indicated by 1.

### 3. CURRENT STATUS

This response consists only of a packet header, thus transmitting only the current status of the terminal to the host processor. The packet header is defined in Section A.9.6. The count field in the header is 0, indicating that no responses are included in this transmission. The terminal sends current status whenever SS=1 is in a received command packet header, refer to Section A.9.5.

#### A.9.11.4 Function Key Response

This response is generated as a result of the terminal operator typing a function key which has been defined by a <REQUEST INPUT> command to send a function key response. The function keys which may be defined by a <REQUEST INPUT> are the following:

0 to 6 – PF0 to PF6  
 7 – abort PF7  
 8 to 10 – PF8 to PF10  
 41 – enter/PF41  
 42 – affirm/PF42  
 43 – close/PF43  
 44 – stop repeat/PF44  
 45 – blank key/PF45

1. <FUNCTION KEY> <COUNT> <FUNCTION KEY #> <AREA FIELD #>

**RESPONSE CODE:** 87

**COUNT:** Number of bytes of data in the response, (=3).

**FCN KEY #N:** A byte in which the upper two bits are zero and the low order six bits represent two octal digits which are the number of the function key that was typed.

**AREA:** The area in which the cursor was positioned when the function key was typed, binary encoded in one byte as follows:

0000 0000 – Display Area  
 0000 0001 – Form Area

**FIELD #:** The field in which the cursor was positioned when the function key was typed, binary encoded in one byte.

**Description:** This response is generated when an input request is pending which has defined one or more function keys to give a function key response, and one of those keys is typed. At this point the keyboard is locked and the input request is suspended.

The host must then respond with a <RESUME INPUT> command to unlock the keyboard and allow the input operations to be continued. The input request will be terminated only when a function key which has been defined to send form field and/or menu field responses is typed.

#### **A.9.11.5 Diagnostic Response**

##### **1. <SELF TEST RESULTS> <COUNT>**

**RESPONSE CODE:** 88

**COUNT:** Number of bytes of data in the self test results (=80).

#### **A.10 KEYBOARD**

Definable Function Keys – There are 15 definable function keys.

**Main Keyboard:** (blank)/PF45

**Keypad:** PF0 through PF6  
abort=PF7  
PF8 through PF10  
enter/PF41  
affirm/PF42  
close/PF43  
stop repeat/PF44

Any of these keys can be defined by a <REQUEST INPUT> command to do one of the following:

- 1 – lock keyboard and send menu fields to the host
- 2 – lock keyboard and send form fields to the host
- 3 – lock keyboard and send menu and form fields to the host
- 4 – lock keyboard and send only a function key response to the host.

If a function key has not been defined by a <REQUEST INPUT> command, when the key is pressed, there will be a key click but the keyboard will not be locked and the terminal will do nothing.

##### **A.10.1 Main Keyboard**

The main keyboard has 62 keys and is mechanically laid out as a DEC standard keyboard.

Fifty-two of the keys are used to generate any of the 95 displayable ASCII characters (040 to 176).

The 10 remaining keys are used to implement keyboard operations which are useful in a form-filling application. These keys, and their associated functions are:

<b>Key</b>	<b>Function</b>
<b>Next Field</b>	<p>If current field is last field, clear to end of current field:</p> <p>Otherwise clear to end of current field and move cursor to first character position of next field.</p> <p>A field check will be done no matter what position the cursor is in when the key is pressed.</p>
<b>Home Form/Home Disp</b>	<p><b>Home form:</b> If form fields are enabled, move cursor to first character of first form field.</p> <p><b>Home Disp:</b> If menu fields are enabled, move cursor to first character of first menu field. Home Display is shift/Home Form.</p>
<b>Clear</b>	<p><b>Display Area:</b> All menu fields are restored to the normal video state, they are deselected, and the cursor is positioned to the beginning of the first menu field.</p> <p><b>Form Area:</b> All form fields are cleared to their clear character value and the cursor is positioned to the beginning of the first form field.</p>
<b>Select/Deselect</b>	<p>This key is only enabled when the terminal is in menu mode and the cursor is in the Display Area. Shifted deselects a menu item and unshifted selects a menu item. Selected menu items are displayed in reverse video and deselected menu items are displayed in normal video.</p>
<b>Erase Char</b>	<p>If current field is left justified and cursor is not at beginning of field, backspace cursor and then erase character under cursor; if cursor is at beginning of field, leave cursor there and erase character under cursor. If left justified field, collapse text from right to left; if right justified field, collapse text from left to right. The positions left vacant by the justify operation will be filled with the clear character specified in the Field Definition for that field.</p>
<b>Delete Field</b>	<p>Set current field to all clear characters and move cursor to beginning of current field.</p>
<b>Delete Char</b>	<p>Delete character under cursor.</p> <p>If left justified field, collapse text from right to left; if right justified field, collapse text from left to right.</p>

<b>Key</b>	<b>Function</b>
Insert Mode	<p>Unshifted enables insert mode; shifted disables insert mode.</p> <p>In insert mode, if a field is left justified, each character typed into the field is entered under cursor and all characters at or to the right of the cursor are shifted one position to the right. If field is right justified, each character typed is entered under cursor and all characters at or to the left of the cursor are shifted one position to the left.</p> <p>Boundary condition: when a field is filled (no clear characters on left for right justified field or on right for left justified field), additional characters typed will not be entered into field and the bell will ring.</p>
(blank key)/PF45	See information on function keys.
Error Reset	<p>When an error occurs, a message is printed on the bottom line (if error messages are present), the bell rings, the ERROR LED is lit, and the terminal enters "error mode." Until the operator presses the error reset key, thus acknowledging the error, all keystrokes simply ring the bell. Errors occur if field validation, as discussed previously, fails or if the host computer selects an error message to be printed.</p>

### **A.10.2 Keypad**

The keypad has 19 keys. They are used as follows:

<b>Key</b>	<b>Function</b>
Forward Field	<p>If current field is last field, move physical cursor to last position in field and logical cursor past field so that no data entry is allowed if a key is pressed.</p> <p>Otherwise move cursor to first character position of next field without changing anything in the rest of the current field.</p> <p>Field check will be done no matter what position the cursor is in when the key is pressed.</p>
Backward Field	<p>If current field is first field, then move cursor to beginning of current field.</p> <p>Otherwise if cursor at beginning of current field then move cursor to beginning of previous field.</p> <p>Otherwise move cursor to beginning of current field.</p>
Cursor Left	<p>If cursor is at the leftmost position of a field, then no-operation. Otherwise, move cursor left one position.</p>

<b>Key</b>	<b>Function</b>
Cursor Right	<p>If cursor is at the rightmost position of a field, then no-operation. Otherwise, move cursor right one position.</p> <p>If auto-tab is enabled and pressing this key causes the cursor to be moved to the next field, field checking will be done on the field exited.</p> <p>This key is only enabled when the terminal is in menu mode and the cursor is in the Display Area. Shifted deselects a menu item and unshifted selects a menu item. Selected menu items are displayed in reverse video and deselected menu items are displayed in normal video.</p>
Enter	See information on function keys.
Affirm-PF42	See information on function keys.
Close-PF43	See information on function keys.
Stop Repeat-PF44	See information on function keys.
0 to 9 and .-	<p>If the keypad is in "numeric" mode then pressing unshifted 0 to 9 or . results in 0 to 9 or .; pressing shifted 0 to 9 or . results in PF0 to PF6 and abort and PF8 to PF10. If the keypad is in "function" mode, then pressing 0 to 9 or . results in PF0 TO PF6 and abort and PF8 TO PF10, independent of the shift key.</p>

### A.10.3 Repeating Keys

The following keys automatically repeat at 30 times/second after having been pressed for 1/2 second:

Cursor Left  
 Cursor Right  
 Forward Field  
 Backward Field  
 Erase Char  
 Delete Char

### A.11 LEDs

The LED display to the right of the keypad is labeled as follows:

<b>The LED labeled. . .</b>	<b>Is on when. . . .</b>
READY	The DDCMP startup sequence has been successfully received.
KEYBOARD LOCKED	The operator may not type in data or select menu items.
FUNCTION KEYPAD	The keypad is in function (as opposed to numeric) mode.
INSERT MODE	The terminal is in insert mode.

<b>The LED labeled...</b>	<b>Is on when...</b>
DISPLAY AREA	The cursor is in the display area.
CLEAR TO SEND	A communication link is established with the host computer and the terminal can send data.
CARRIER	An active communication link has been established with the host computer.
KEYING ERROR	The terminal is in error mode.

The layout of the LED display is:

```

0-----READY
 0-----INSERT MODE
0-----KEYBOARD LOCKED
 0-----DISPLAY AREA
0-----KEYING ERROR
 0-----CLEAR TO SEND
0-----FUNCTION KEYPAD
 0-----CARRIER

```

## **A.12 COMMUNICATIONS PROTOCOLS**

### **A.12.1 Logical Communications Protocol**

The VT62 will be a tributary DDCMP station and operate only in a polled environment, regardless of whether the connection is multipoint or point-to-point.

It is the host computer's responsibility to know the "address" of the VT62. The VT62 will only accept DDCMP blocks of up to 2048 characters in length. Blocks greater than 2048 characters will cause the terminal to respond with a <NAK-Message Too Long>. The terminal will send DDCMP blocks of no greater than 256 bytes of data (maximum DDCMP block would be 8 bytes of header, 256 bytes of data, and 2 CRC characters).

After the terminal is powered up or after it has run self test or if an overlength command packet has been received or if a <RESET> is issued, the terminal will respond to all polls that are not a START with a START to the host. The host must issue a DDCMP START for the terminal to respond to Command Packets.

A DDCMP START issued to the VT62 will cause a complete reset of the VT62 with the exception of the stored SELF TEST RESULTS.

The terminal will always respond with a STACK message to a START message from the host. The terminal always verifies the CRC before acting on data it has received.

If the VT62 command buffer is temporarily too full to accept the full DDCMP block, the terminal will respond with a <NAK-Buffer Temporarily Unavailable> until space is available in the buffer for that block.

## **A.12.2 Physical Communications Protocols**

### **A.12.2.1 Asynchronous**

The VT62 operates up to 9600 baud asynchronous. Support is provided for 103 data sets (or equivalent) on dial-up or leased lines, full duplex. Support is also provided for 202 datasets (or equivalent) on dial-up half duplex or leased full duplex or half duplex lines.

### **A.12.2.2 Synchronous**

The VT62 operates up to 4800 baud synchronous. Support is provided for 201 or 208 datasets (or equivalent) on dial-up half duplex lines or leased full duplex or half duplex lines.

## APPENDIX B

# INSTALLATION PROCEDURES FOR COMMUNICATIONS OPTIONS

### B.1 REMOVING THE COMMUNICATIONS OPTION

To remove the communications option, refer to Figures B-1 through B-4, unit assembly drawing 0-UA-VT62-0-0, and perform the following steps.

1. Place the unit upside down on a bench or table (Figure B-1).
2. Remove the cable clamp securing the option cable to the base assembly (two Phillips head screws, 8-32 × 0.56) using a Phillips screwdriver.
3. Remove the base front screen (three hex-head screws, 8-32 × 0.38) using a 6.35 mm (1/4 inch) nut driver.
4. Remove the switch panel board connector P1 from J4 on the I/O board (Figure B-2).

#### CAUTION

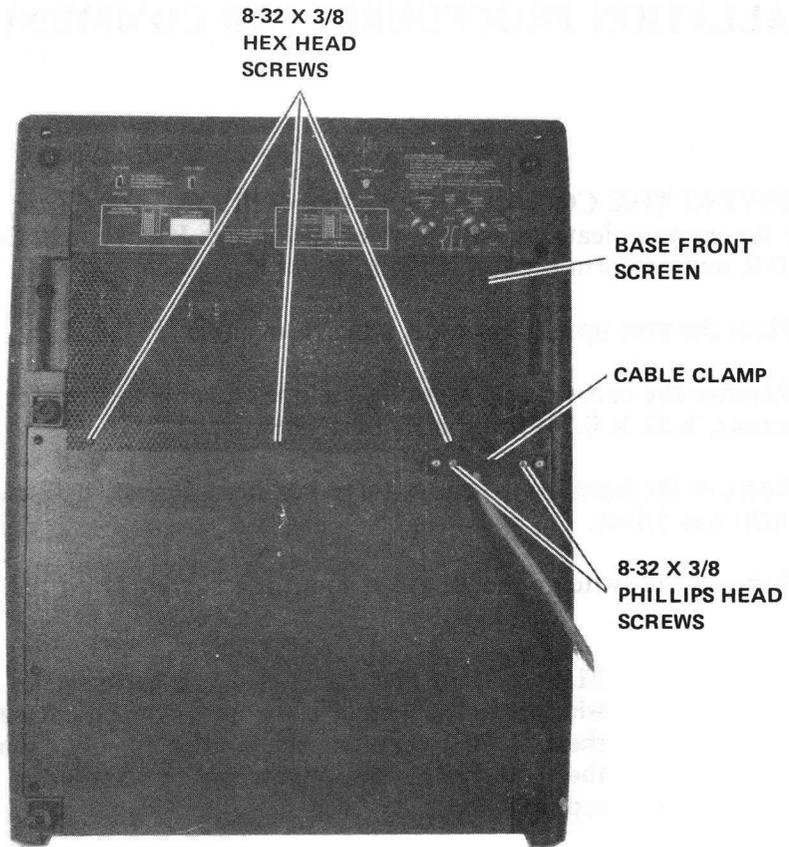
**Make certain that P1 is replaced correctly on J4, with the black wire of the cable toward the front of the unit. P1 incorrectly replaced on J4, will damage the I/O board upon power-up, necessitating its replacement.**

5. Remove the communications option card from its socket on the I/O board. The option card plugs into one (Figure B-3) or both (Figure B-4) of the two 11-pin sockets located near the center of the I/O board. The card is secured by arrowhead snap-on fasteners. Squeeze the fastener and lift the card from its socket.

### B.2 INSTALLING THE COMMUNICATIONS OPTION

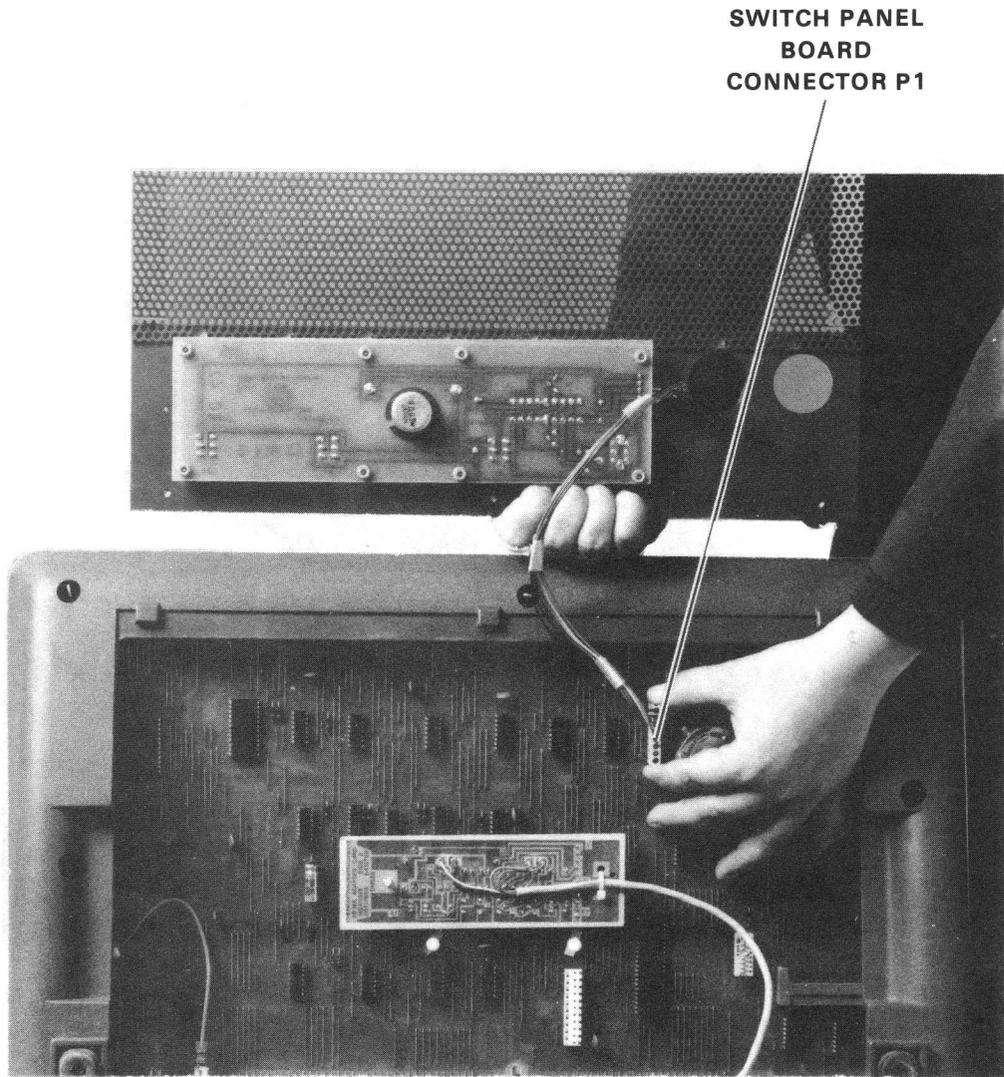
To install the communications option, refer to unit assembly drawing 0-UA-VT62-0-0 and perform the following steps.

1. Remove the communications option card presently installed (Paragraph B.1).
2. Install the new card by plugging the row of straight pins on the card into the proper socket on the I/O board as shown in the applicable figure for the communications option (which follows) and secure the card. Figure B-3 shows the installation of the 20 mA communications option card. Figure B-4 shows the installation of the EIA communications option card (with control), BN61B.
3. Route the card cable approximately as shown in the applicable figure for each type of card so that the cable will exit the unit through the notch in the cable clamp provided for it.
4. Reassemble the unit by reversing steps 1 through 5 of Paragraph B.1.



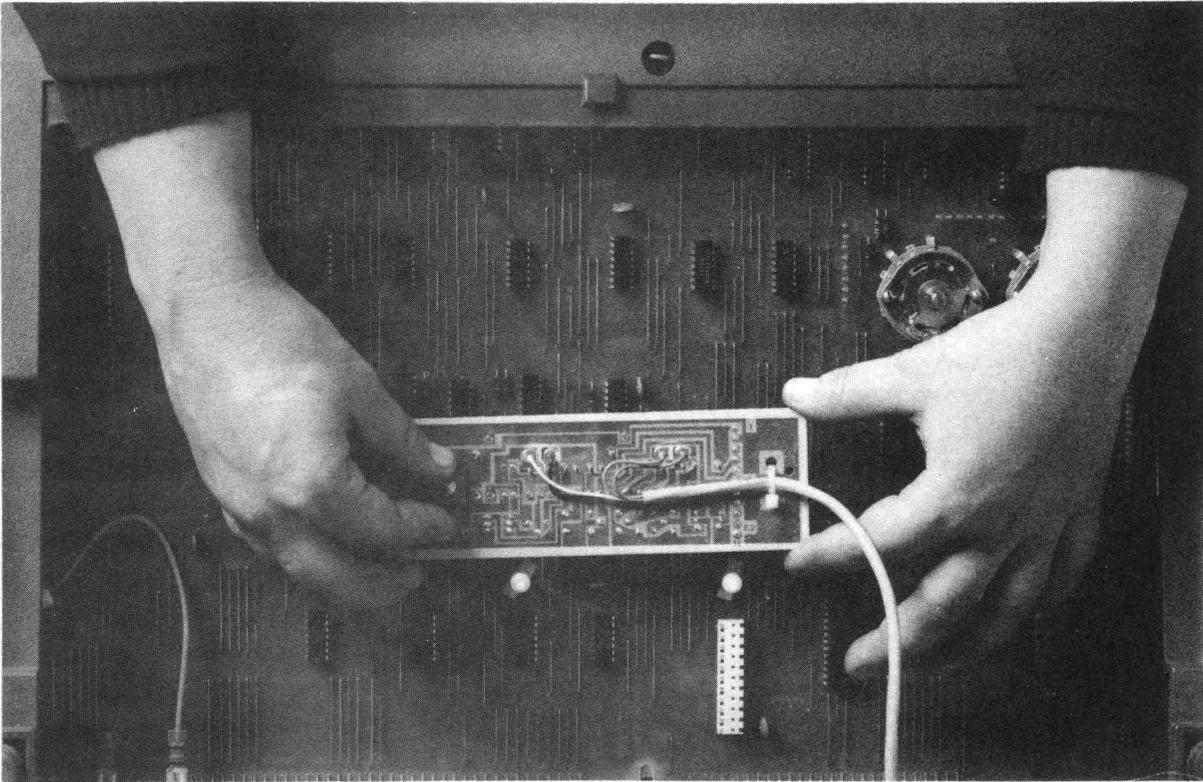
8595-9-A0001

Figure B-1 Terminal Servicing Position



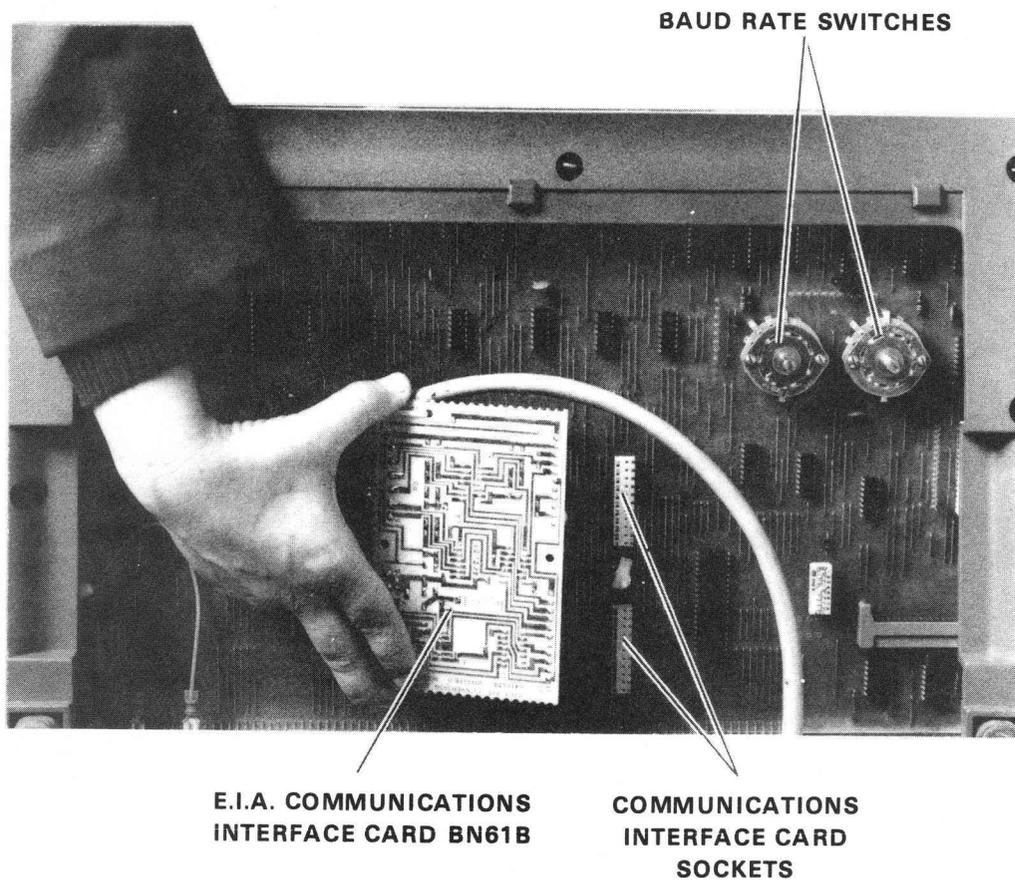
8284-14-A0008

Figure B-2 Switch Panel Board Connector Removal



8284-20-A0025

Figure B-3 20 mA Communications Option Card Removal



8284-12-A0026

Figure B-4 EIA Communications Option Card (BN61B) Removal

VT62 DECSCOPE USER'S GUIDE  
EK-VT62-UG-001

Your comments and suggestions will help us in our continuous effort to improve the quality and usefulness of our publications.

What is your general reaction to this manual? In your judgment is it complete, accurate, well organized, well written, etc.? Is it easy to use? \_\_\_\_\_

\_\_\_\_\_  
\_\_\_\_\_

What features are most useful? \_\_\_\_\_

\_\_\_\_\_  
\_\_\_\_\_

What faults or errors have you found in the manual? \_\_\_\_\_

\_\_\_\_\_  
\_\_\_\_\_

Does this manual satisfy the need you think it was intended to satisfy? \_\_\_\_\_

Does it satisfy *your* needs? \_\_\_\_\_ Why? \_\_\_\_\_

\_\_\_\_\_  
\_\_\_\_\_

Please send me the current copy of the *Technical Documentation Catalog*, which contains information on the remainder of DIGITAL's technical documentation.

Name \_\_\_\_\_ Street \_\_\_\_\_

Title \_\_\_\_\_ City \_\_\_\_\_

Company \_\_\_\_\_ State/Country \_\_\_\_\_

Department \_\_\_\_\_ Zip \_\_\_\_\_

Additional copies of this document are available from:

Digital Equipment Corporation  
444 Whitney Street  
Northboro, Ma 01532  
Attention: Communications Services (NR2/M15)  
Customer Services Section

Order No.           EK-VT62-UG-001

-----  
**Fold Here**  
-----

-----  
**Do Not Tear - Fold Here and Staple**  
-----

**FIRST CLASS  
PERMIT NO. 33  
MAYNARD, MASS.**

**BUSINESS REPLY MAIL  
NO POSTAGE STAMP NECESSARY IF MAILED IN THE UNITED STATES**

**Postage will be paid by:**

**Digital Equipment Corporation  
Technical Documentation Department  
Maynard, Massachusetts 01754**

