VSV21 Version 2.0
User Guide

AZ-FV70B-TC

March 1986

This guide provides a functional description of the VSV21 hardware and software, and lists the peripheral devices and interfaces. The configuration and control of the device are also described. The methods of loading VT220 and VSV11 emulation are given.

This guide is part of the VSV21 Version 2.0 document set that supersedes the VSV21 Version 1.0 document set.

Operating Systems: RSX-11M-PLUS Version 3.0
Micro/RSX Version 3.0
MicroVMS Version 4.2

Software: RSX-11M-PLUS VSV21 Version 2.0
Micro/RSX VSV21 Version 2.0
MicroVMS VSV21 Version 2.0

digital equipment corporation • maynard, massachusetts

First Edition, July 1985
Second Edition, March 1986

Using Digital's networked computer systems, this book was produced electronically by the Media, Publishing and Design Services department in Reading, England.

The information in this document is subject to change without notice. Digital Equipment Company Ltd. assumes no responsibility for any errors herein.

Printed in U.K.

CONTENTS

CHAPTER 3 CONFIGURATION

CHAPTER 4 MODES OF OPERATION

INDEX

FIGURES

TABLES

PREFACE

MANUAL SCOPE

This manual provides the user of the VSV21 graphics option with all the information necessary to operate it successfully.

Chapter 1 contains a general description of the VSV21, together with a list of peripherals, cable connections, and interfaces. This chapter also includes a technical specification.

Chapter 2, Functional Description, contains information on the hardware and relevant software, and includes a detailed system block diagram of the VSV21.

Chapter 3, Configuration, includes information on address selection, display resolution selection, initialization, and self-test. The chapter also includes information on the console emulator mode and the VCP utility, complete with a list of commands.

Chapter 4, Modes of Operation, covers all the relevant operating information on VT220 emulation and VSV11 emulation.

CHAPTER 1
GENERAL DESCRIPTION

1.1 INTRODUCTION

The VSV21 provides powerful medium-resolution graphics capabilities on Q22-bus systems. It can be used in many applications, but primarily it is aimed at process monitoring, low-end scientific, engineering, and CAD/CAM applications.

The main features of the VSV21 are:

- Switch-selectable resolution
- Supported by a range of systems

  PDP-11/23+

  PDP-11/73

  MicroVAX II
- Provided with either Micro/RSX and RSX-11M-PLUS or MicroVMS support software
- VSV11 emulation
- VIVID command language interpreter. VIVID is an instruction set developed for the VSV21
- Limited VT220 emulation
- Supports a range of color monitors
- Connector port for pointing device (joystick, trackball, tablet, or mouse)
- Transparent serial port.

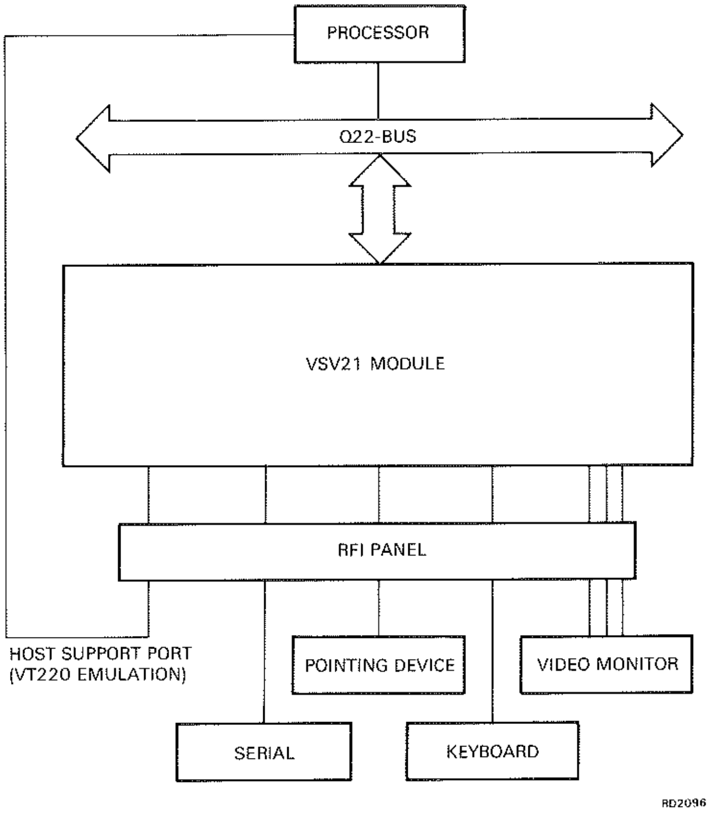


Figure 1-1 Simple Block Diagram of the VSV21

1.2 PHYSICAL DESCRIPTION

The VSV21 is based on a single quad-height module (VSV21-AA) which can be connected to a Q22-bus system interface or similar. Cables connect the module to a bulkhead panel which houses the four serial-port connectors and the color video output connectors. Two arrangements of bulkhead installation are used, depending upon the cabinet option. These are shown in Figure 1-2.

The VSV21 is supplied in three specific bulkhead kits to suit the following cabinets.

- BA23 box, I/O panel, and cables (VSV21-AB)
- BA123 box, I/O panel, and cables (VSV21-AC)
- BA11 box, I/O panel, and cables (VSV21-AD)

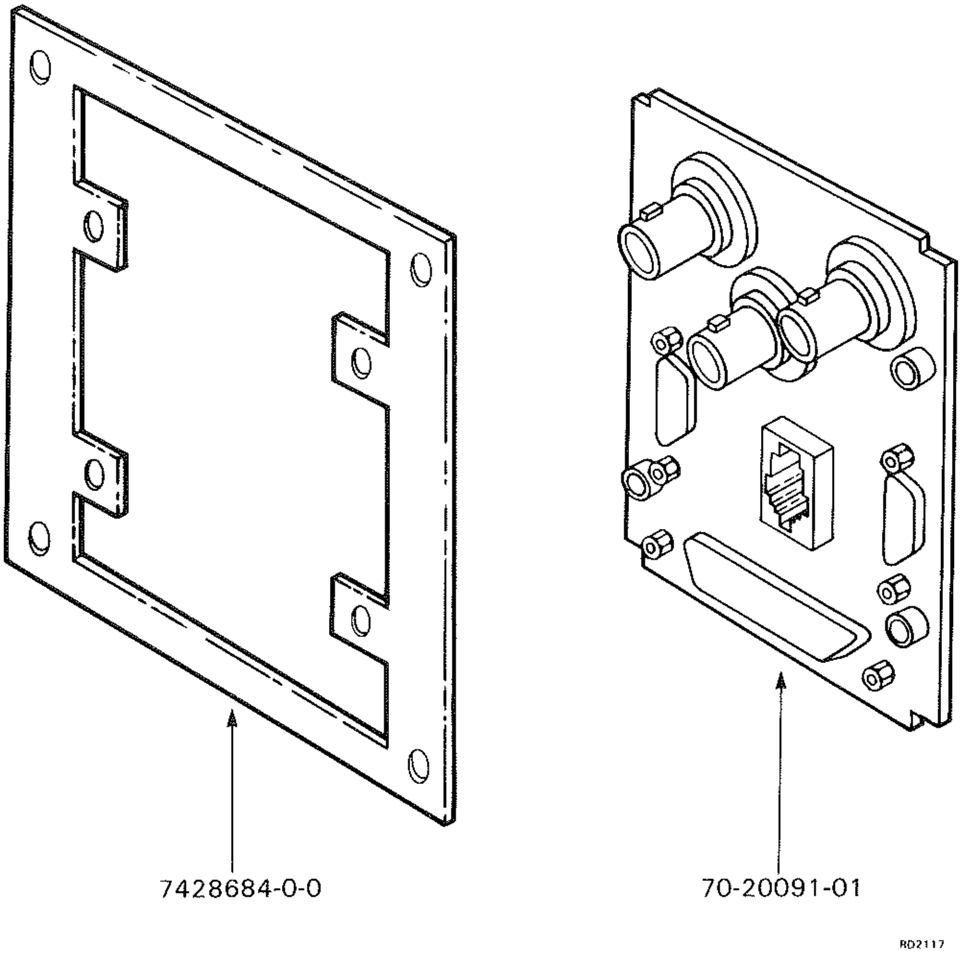


Figure 1-2 VSV21: I/O Panel Arrangement

1.3 INTERFACES

1.3.1 System Bus Interface

The VSV21-AA module connects directly to the Q22-bus backplane connector of the host processor via the A or B connectors. It will therefore connect via any slot.

1.3.2 Serial Interfaces

Four function-dedicated serial I/O ports are provided on the I/O panel for:

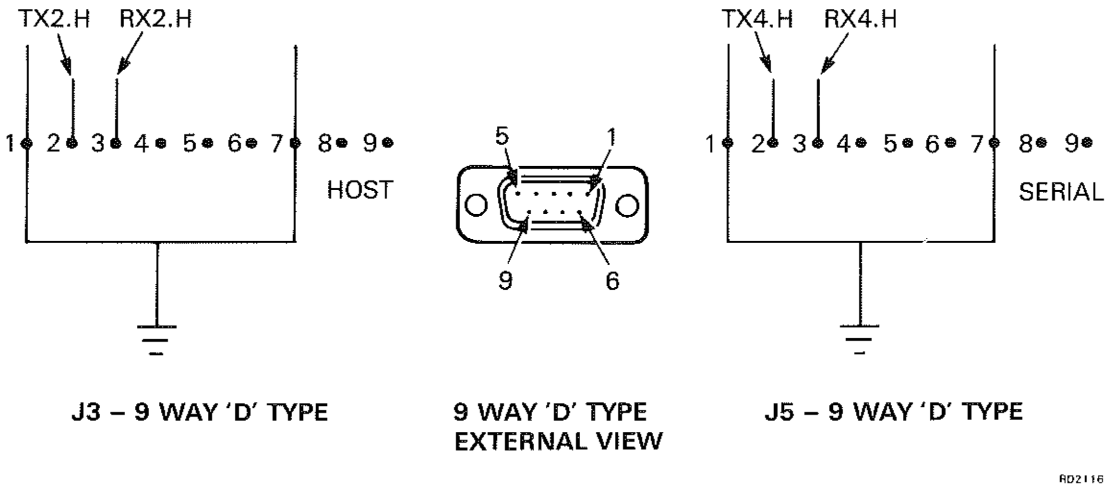
- Host processor support via the console port or a serial line (D9-type connector)

- Serial joystick, trackball, or mouse (D25-type connector)

- Serial transparent port (D9-type connector)

- LK201 keyboard (keyboard jack connector, LK201)

CAUTION: Do not try to connect the keyboard or a pointing device while the power is ON as this will blow the on-board fuses. ALWAYS switch the power OFF before attempting to connect these devices.



J3 – 9 WAY 'D' TYPE

9 WAY 'D' TYPE EXTERNAL VIEW

J5 – 9 WAY 'D' TYPE

Figure 1-3 9-Way Connector Pin Designations (J3, J5)

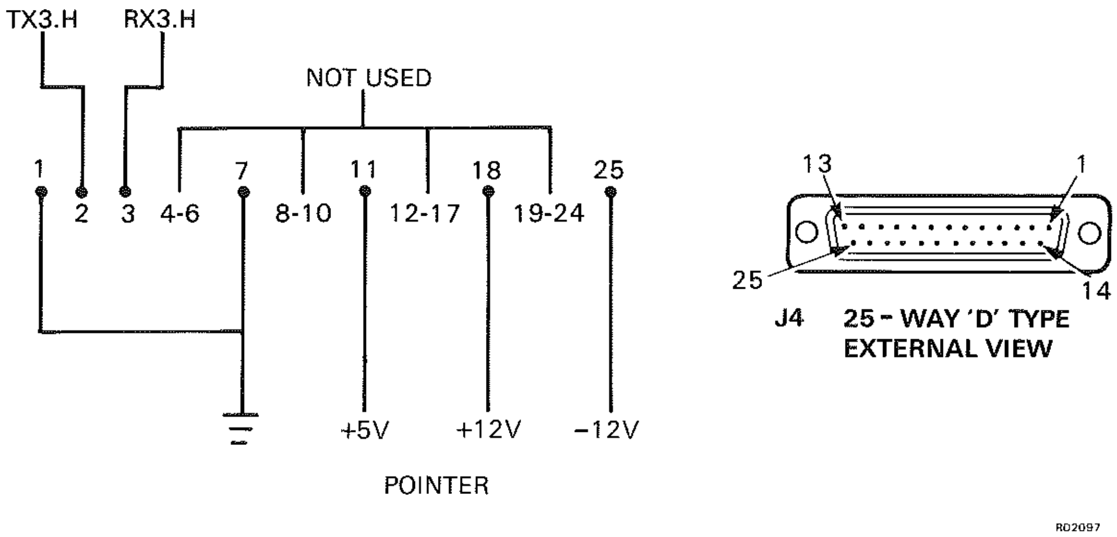


Figure 1-4 25-Way Connector Pin Designations (J4)

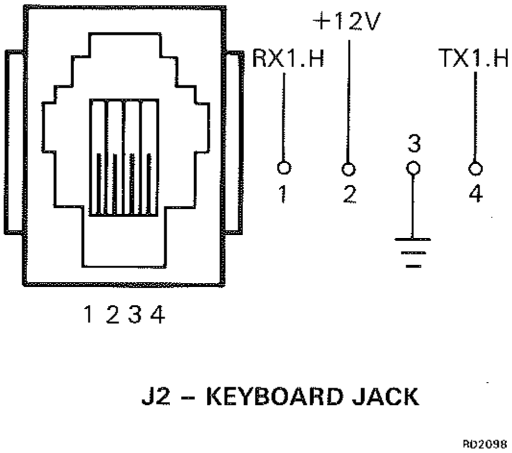


Figure 1-5 LK201 Keyboard Jack Connector

1.3.3 Video Interface

The video output is provided as red, green, blue (with sync on green) in RS-343 standard. It interfaces to recommended monitors. Currently these are:

- Barco 5337 14-inch NP
- Barco 5351 20-inch NP
- DIGITAL VR241

The Barco monitors must be used in high-resolution mode. The DIGITAL VR241 is intended for low-resolution modes only (low resolution is 640 × 240 and 512 × 256 pixels).

You need to set up the DIGITAL VR241 monitor at the installation stage to produce the correct aspect ratio. Refer to the *VSV21 Installation Manual (AV-FV71B-TC)* for the necessary information.

1.4 GENERAL SPECIFICATIONS

1.4.1 Environmental Conditions

● Storage temperature:	−40 degrees C to 66 degrees C (32 degrees F to 151 degrees F)
● Operating temperature:	5 degrees C to 60 degrees C (41 degrees F to 140 degrees F)
● Relative humidity:	10% to 95% non-condensing

1.4.2 Electrical Conditions

- +5 V dc ±5% at 5.5 A (typical) 7.6 A (maximum)
- +12 V dc ±3% at 0.15 A (typical) 0.2 A (maximum)

These figures do not include any power provided to a peripheral through the serial interface. If needed, this will be +12 V dc with fuse protection of 3 A, and +5 V dc with fuse protection of 3 A, and −12 V dc at 50 milliamps unprotected.

The LK201 keyboard current consumption is 0.4 A at +12 V dc.

The bus loading figures are:

- 1.9 Q-bus ac bus loads
- 0.5 Q-bus dc bus loads

1.4.3 Performance

1.4.3.1 Drawing Modes – The VSV21 can operate in one of two pixel memory updating modes. These are:

- Display priority
- Drawing priority

Display priority is the recommended mode, as it allows the user to access preconfigured graphic models. This allows convenient creation of finished graphic presentations. In display priority mode the pixel memory is only updated during line and frame fly-back times. This mode eliminates update flicker, at the cost of throughput (60% to 70% of the drawing priority output).

Drawing priority allows the user to draw completely new characters if the preconfigured graphic models do not suit the application. In drawing priority mode the pixel memory update can occur at any time, but the display must be blanked during the period of update.

The user is not provided with any method of selecting the pixel memory update mode, and there is no VCP command to perform such a selection. The mode can be changed, however, by means of the native software VIVID.

1.4.3.2 Graphics Performance – The following figures refer to display priority mode.

The line-drawing speed is better than 15 000 short vectors per second (640 × 480 pixels) (based on a short vector length of 16 pixels, and including processing time for the vector parameters).

The text drawing speed for characters of standard size is 1250 characters per second for 640 × 480 and for 512 × 512; it is 2750 characters per second for 64 × 240 and for 512 × 256. (Standard size text is defined as 8 × 20 pixels for high resolution, and as 8 × 10 pixels for low resolution.)

The area fill speeds are better than 200 000 pixels per second (640 × 240) or 400 000 pixels per second (640 × 480).

Pixel area copying speed is at least 200 000 pixels per second.

Any screen resolution can be cleared to a predetermined color within one frame scan period.

1.4.3.3 Pixel Memory – The pixel memory (for a resolution of 640 × 480 pixels) can be loaded from host memory within one second, or dumped to host memory within eight seconds.

1.4.3.4 Communications Links – The four serial ports are programmable, and support data rates of up to 9600 baud.

- The host console port is initialized at 9600 baud.
- The keyboard port is initialized at 4800 baud.
- The pointing device port is initialized at 9600 baud.
- The transparent serial port is initialized at 9600 baud.

1.4.3.5 Peripheral Devices –

- The local VSV21 response to a peripheral input signal occurs within 16 milliseconds.
- The maximum cable length between the host and peripheral equipment is 4.25 metres (14 feet) for the keyboard and 7.6 metres (25 feet) for other devices.

CHAPTER 2
FUNCTIONAL DESCRIPTION

2.1 VSV21 HARDWARE

The VSV21 is a graphics device giving four selectable screen resolutions, and using four color channels to provide 16 selected colors from a palette of 4096. The hardware is controlled by a microprocessor system which controls the other components on the board: graphics generation, serial communications, and the Q22-bus interface. Figure 2-1 is a block diagram representing the major hardware components. The function of each component is described in more detail in the next section.

2.1.1 The Hardware Module

Each paragraph in this section corresponds with one block of the diagram given in Figure 2-1.

Software running on the on-board **microprocessor** controls the whole operation of the VSV21 module. ROM memory contains the essential parts of the software needed to start up the VSV21 module. The main part of the software is downloaded from the host into RAM memory when the host operating system is up and running. Non-volatile RAM is used to store configuration and operating parameters so that they are not lost during power-down. The microprocessor communicates with the other components of the VSV21 module using an on-board dedicated bus.

The **graphics generation** component has three main sections. The graphics controller uses a purpose-designed integrated circuit, an Advanced CRT Controller (ACRTC), which converts drawing commands into data patterns in memory. These patterns are stored in the pixel memory in such a way as to give four color-channels for each pixel on the screen. The data is converted using the Color Look-Up Table (CLUT), to red, green, and blue outputs suitable for driving a color video monitor. The CLUT is able to provide one of sixteen colors, selected from a palette of 4096 colors, for each pixel on the screen.

The **serial communications** component handles the four serial I/O ports. There is one port dedicated to each of the following functions: host communications, a pointing device, and a keyboard. There is also a transparent programmable serial port. For more information on each I/O port see Section 2.1.3.

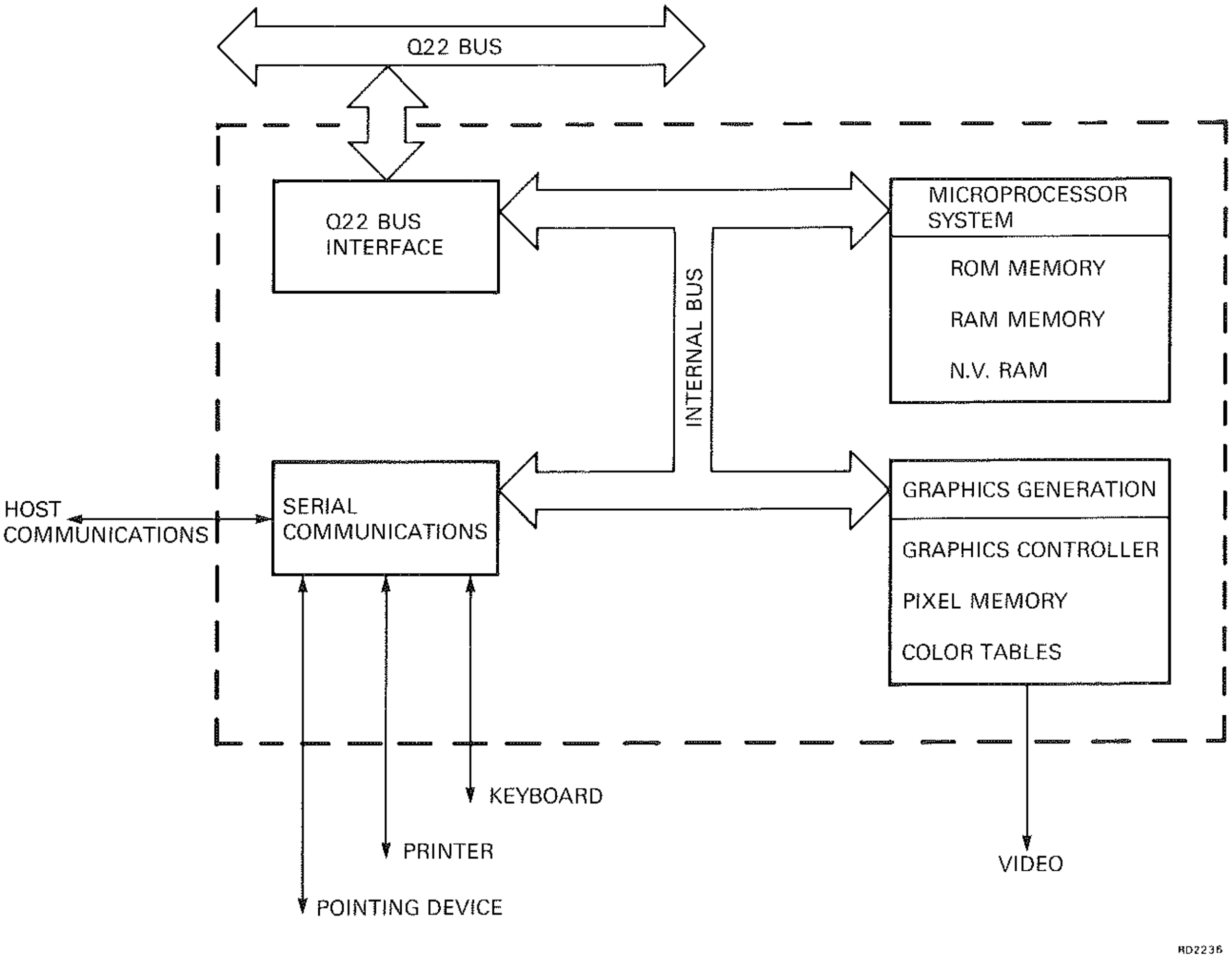


Figure 2-1 VSV21 Hardware Block Diagram

The **Q22-bus interface** handles all the communications with the host system through the Q22-bus. It includes a Direct Memory Access (DMA) controller which is used to load display files and downloadable software into the on-board RAM memory. There is no direct connection between the on-board microprocessor and the host system. All communication between the host and the VSV21 is done through this Q22-bus interface.

2.1.2 Graphics Functions

The VSV21 can operate in one of three modes.

- Full-screen VT220 emulation
- Split-screen: graphics with a text window
- Full-screen graphics

The VSV21 supports the following graphics functions.

NOTE: The functions available to host software (from those listed) depend on the on-board software. Different emulators will support different sets of functions. For more information on the on-board software see Section 2.2.

Drawing functions:

- Polyline
- Polymarker
- Fill area
- Polygon
- Text
- Circle
- Circular arc
- Ellipse
- Elliptical arc

Screen functions:

- Copy
- Clear screen
- Load color look-up table
- Cursor control
- Load text font
- User-definable text fonts
- Write text in eight directions
- Text inclination

- Text enlargement
- Text aspect ratio
- Save and restore pixel memory

Input functions:

- Read keyboard
- Read locator device
- Select/deselect input device
- Pick
- Sample, stroke, event

The VSV21 allows you to control the following graphic attributes:

- Line style, width, color
- Marker symbol
- Fill pattern, color
- Polygon edge style, edge color, interior fill style, color
- Circles, arcs, and ellipses similar to lines and polygons

The VSV21 allows you to control the following segment attributes:

- Visibility
- Detectability
- Highlighting

You define the mapping of an image to the viewing surface by specifying parameters for the following.

- Window
- Viewport

2.1.3 Serial I/O Ports

There are four serial I/O ports on the VSV21:

- Host communications port
- Keyboard port
- Transparent serial port
- Pointing device port

2.1.3.1 Host Communications Port – An I/O port is provided to connect the VSV21 to the host processor through a standard communications interface (such as a DZV11) as shown in Figure 2-2. This allows the VSV21 to emulate a VT220 terminal connected to the host in the normal way. All communication to the host relating to the VT220 emulation, except host console support, is done through this serial interface.

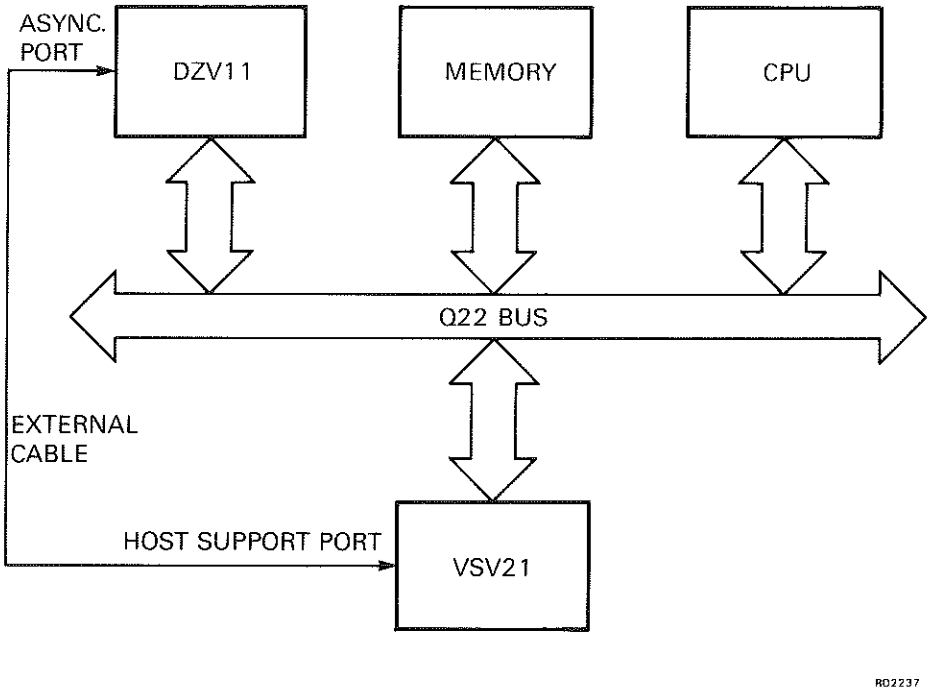


Figure 2-2 Host Communications Through a DZV11

If the same I/O port is connected to the host console port, as shown in Figure 2-3, you can use the VSV21 to boot the host system by using the basic terminal emulation built in to the VSV21. The VSV21 will then behave as a VT220 connected to the host console port.

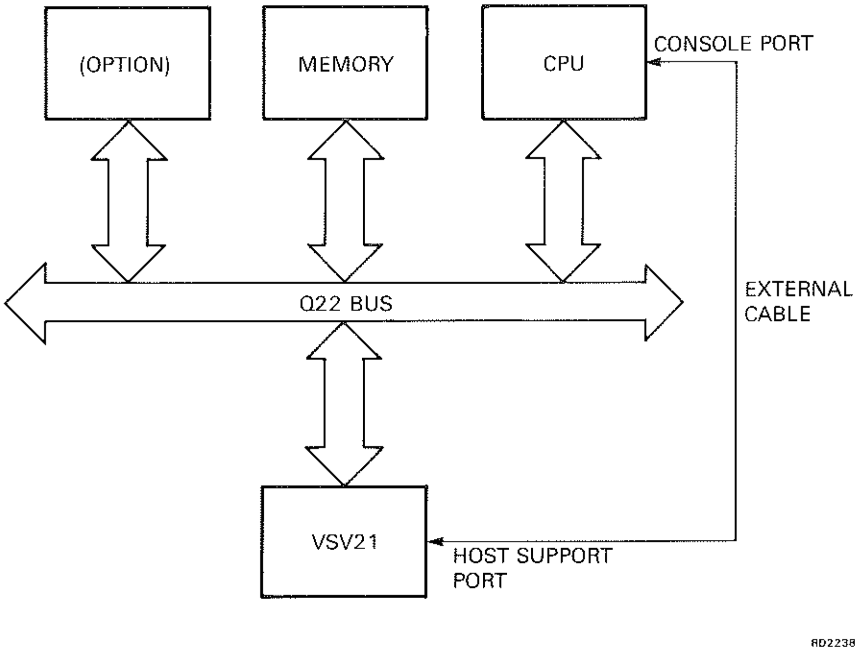


Figure 2-3 Host Communications Through the Host Console Port

2.1.3.2 Keyboard Port – The keyboard port is designed to connect to the standard DIGITAL LK201, versions AA to AH inclusive, AK to AN inclusive, and AP and AS. This is needed for the VT220 emulation. Communications from the keyboard to the host processor are through the external host communications link.

2.1.3.3 Transparent Serial Port – This port is a general-purpose input/output port. It is supported by a separate device driver which co-exists at the same level as the graphics interpreters. The device driver can thus be loaded and used without the interpreters, though this is not a practical nor useful configuration. See also section 3.2.5 which explains memory occupancy.

2.1.3.4 Pointing Device Port – The pointing device port can be connected to a joystick or trackball, and used to control the drawing on the screen under software control. See DIGITAL brochure YK-AF01A-53 for the list of recommended third party pointing devices.

2.2 VSV21 SOFTWARE

2.2.1 On-Board Software

Figure 2-4 is a block diagram of the main components of the VSV21 on-board software. The software runs on the on-board microprocessor, and the central part is the **kernel and self-test** module. The self-test is used to check out the VSV21 hardware when the module is powered on. It is decribed in the *VSV21 Installation Manual (AZ-FV71B-TC).*

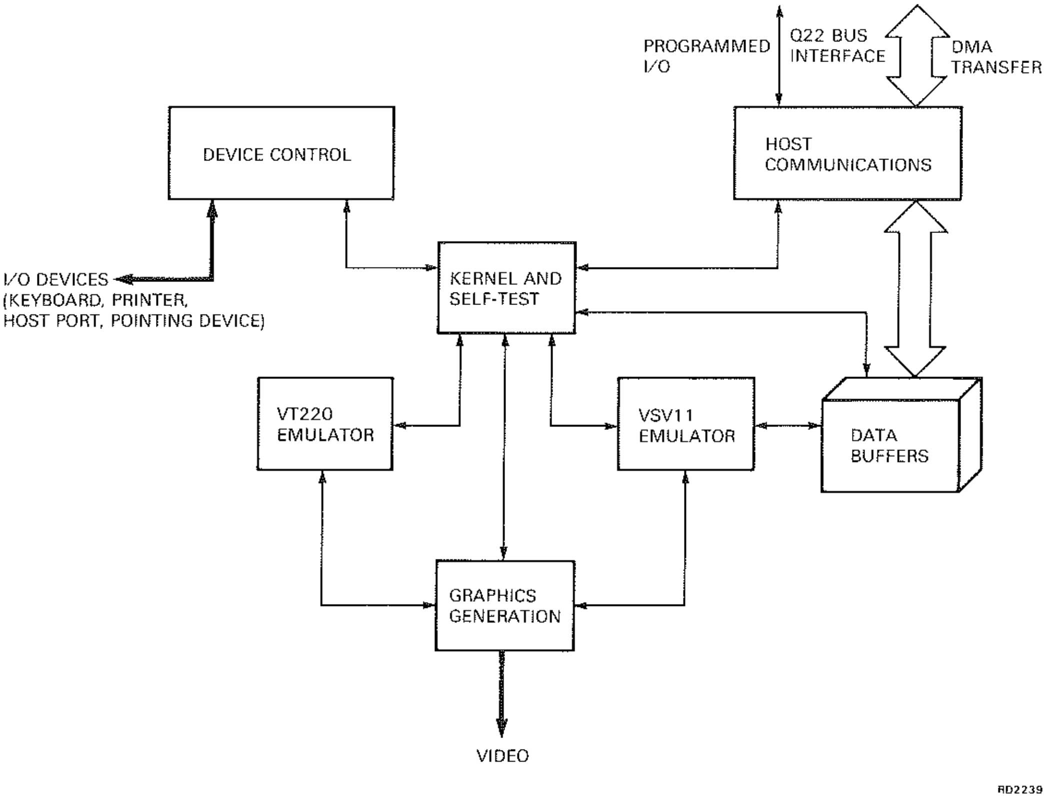


Figure 2-4 VSV21 On-Board Software Block Diagram

Commands to the VSV21 are received by the **host communications** module from the device driver running on the host system, using a programmed I/O mechanism. These commands are passed via the kernel module to the appropriate software component. Some commands are handled by the kernel itself; for example, a command to the VSV21 to download a display list. The display list would be loaded into the VSV21 data buffers (in RAM memory) by the hardware DMA controller.

The host communications module, the kernel and self-test module, and sufficient VT220 emulation and graphics generation to allow the VSV21 to boot the host system, are permanently resident on the VSV21 module (that is, they are stored in ROM memory). All the rest of the on-board software has to be downloaded into the VSV21 when the host operating system is running. This allows the user to select the appropriate software to perform the functions needed. A utility is provided to load the software into the VSV21 (see Chapter 3).

The following sections briefly describe the software modules that can be downloaded into the VSV21. Further details of the different modes of operation are given in Chapter 4.

2.2.1.1 VIVID Interpreter – The VIVID interpreter translates VIVID instructions.

2.2.1.2 VSV11 Emulator – When the VSV11 emulator software is loaded, the VSV21 performs exactly as a minimum-configuration single-channel VSV11 device.

2.2.1.3 VT220 Emulator – With the on-board VT220 emulator software, the VSV21 can perform as a limited VT220. The keyboard is configured as that of a VT220 in VT100 mode.

2.2.2 Host System Software

Figure 2-5 gives a block diagram of the VSV21 host system software. This diagram shows that the device driver communicates with the VSV21 through the programmed I/O mechanism, and that display files and the VSV21 on-board software are loaded into the VSV21 by DMA transfer.

2.2.2.1 VSV21 Support Services –

NOTE: A detailed description of the VSV21 Support Services is given in the VSV21 Programmer's Guide (AA-FV67B-TC).

The VSV21 Support Services provide the applications programmer with software routines to create, control, and modify display files containing graphics primitives and attributes that describe graphic objects, and also to configure and control the VSV21 device.

The programmer can manipulate the display files directly (for example, by writing MACRO-11 programs) and issue QIO system calls to the VSV21 device driver. The support services allow higher-level programming.

The support services and the QIO mechanism are fully described in the *VSV21 Programmers Guide.*

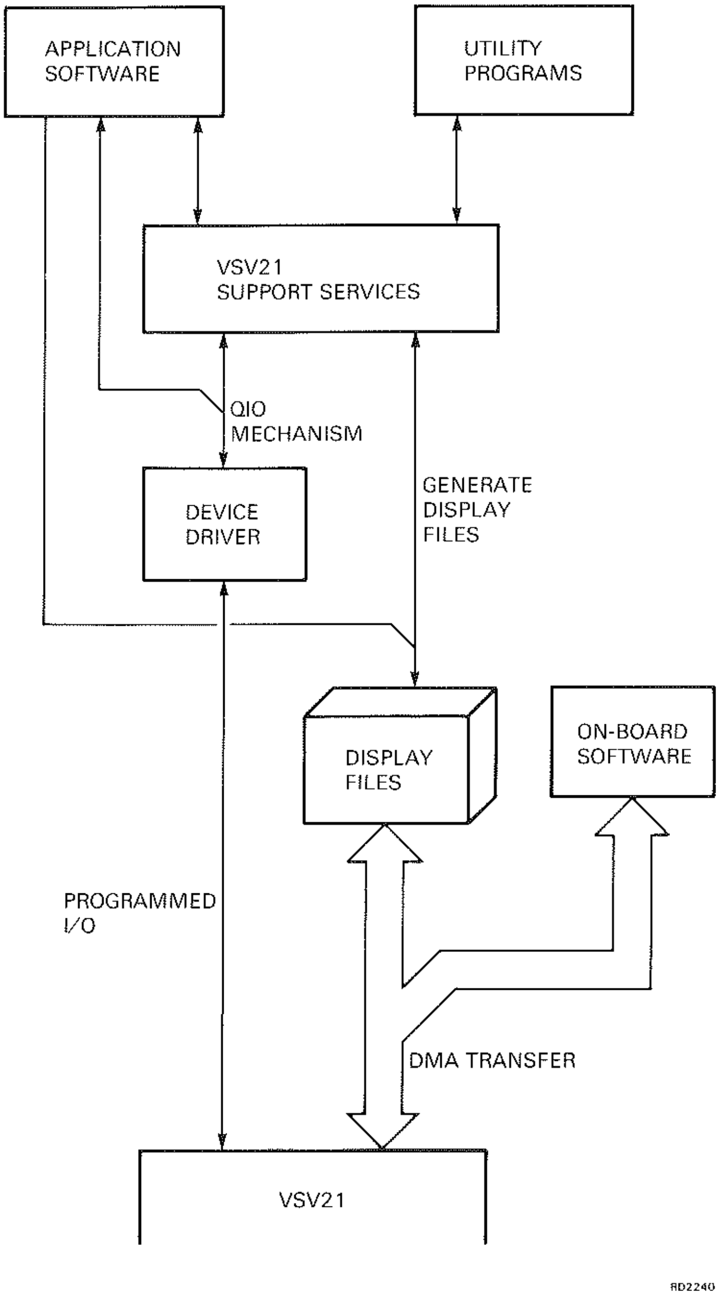


Figure 2-5 VSV21 Host System Software Block Diagram

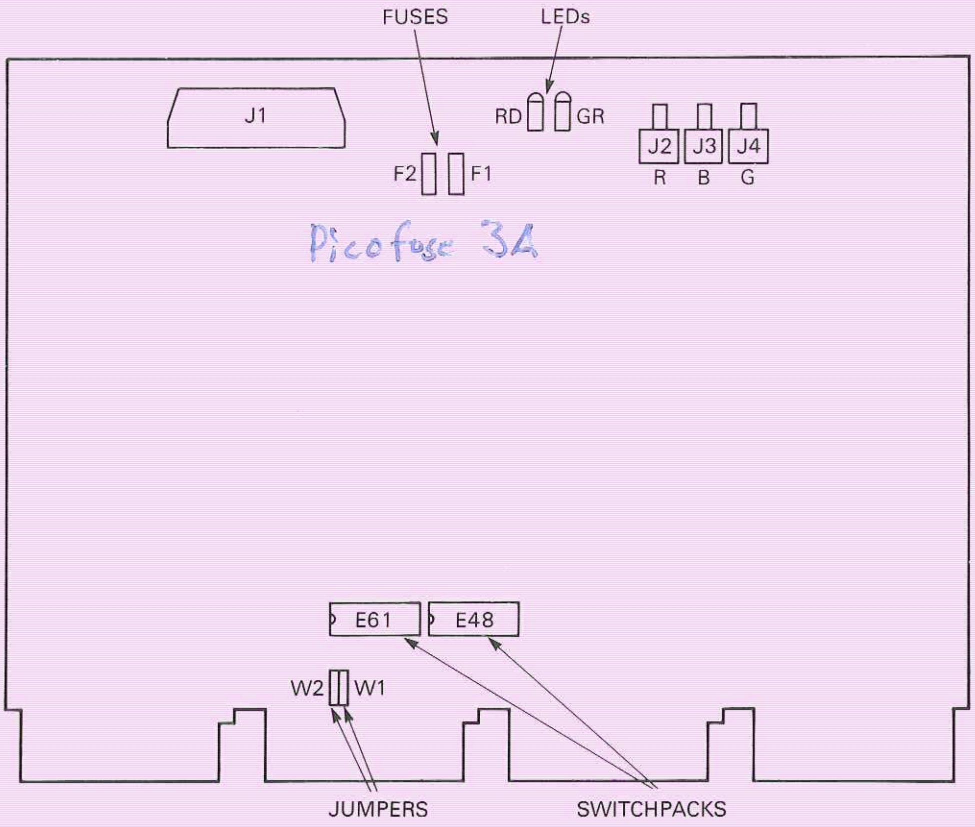
2.2.2.2 The VCP Utility – The VCP (VSV21 Control Program) utility is provided to allow the user to configure and control the VSV21 device by using simple commands. The VCP utility provides a DCL interface to the control routines in the VSV21 support services.

The utility is described in Chapter 3.

CHAPTER 3
CONFIGURATION

3.1 ADDRESS SELECTION

The VSV21 module (M7656) has switches which allow the device CSR and vector addresses to be selected. There are two 10-way DIL switchpacks situated on the logic module. Switchpack E61 is for device address selection and switchpack E48 is for selection of interrupt vectors and device configuration (display resolution selection). A full description of these switches is given in the *VSV21 Installation Manual (AV-FV71B-TC).* Figure 3-1 shows their layout and orientation, and Figures 3-2 and 3-3 give details of the VSV21 addresses and vectors.



RD2102

Figure 3-1 Layout and Orientation of the Module Switchpacks

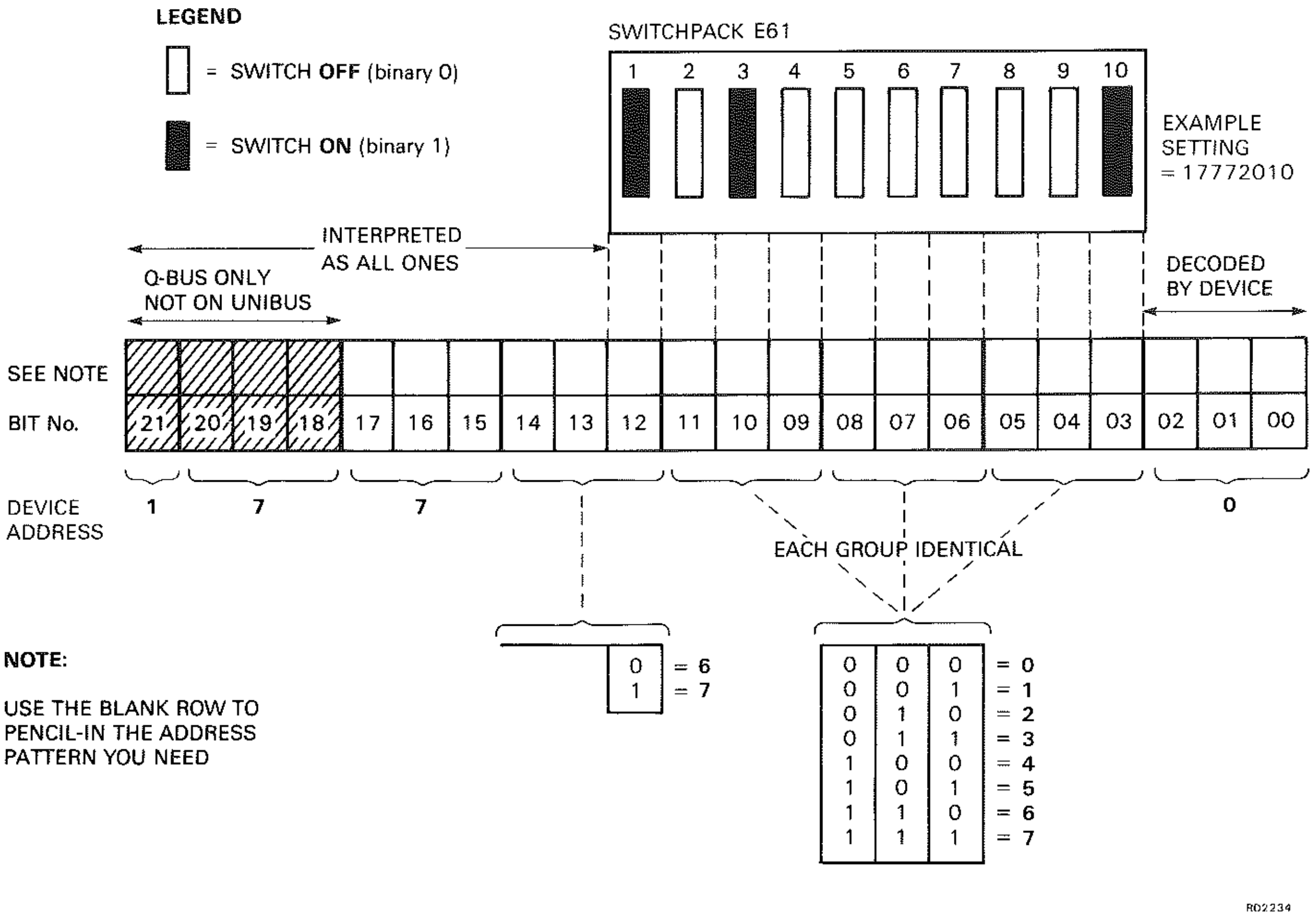


Figure 3-2 Device Address Selection Guide

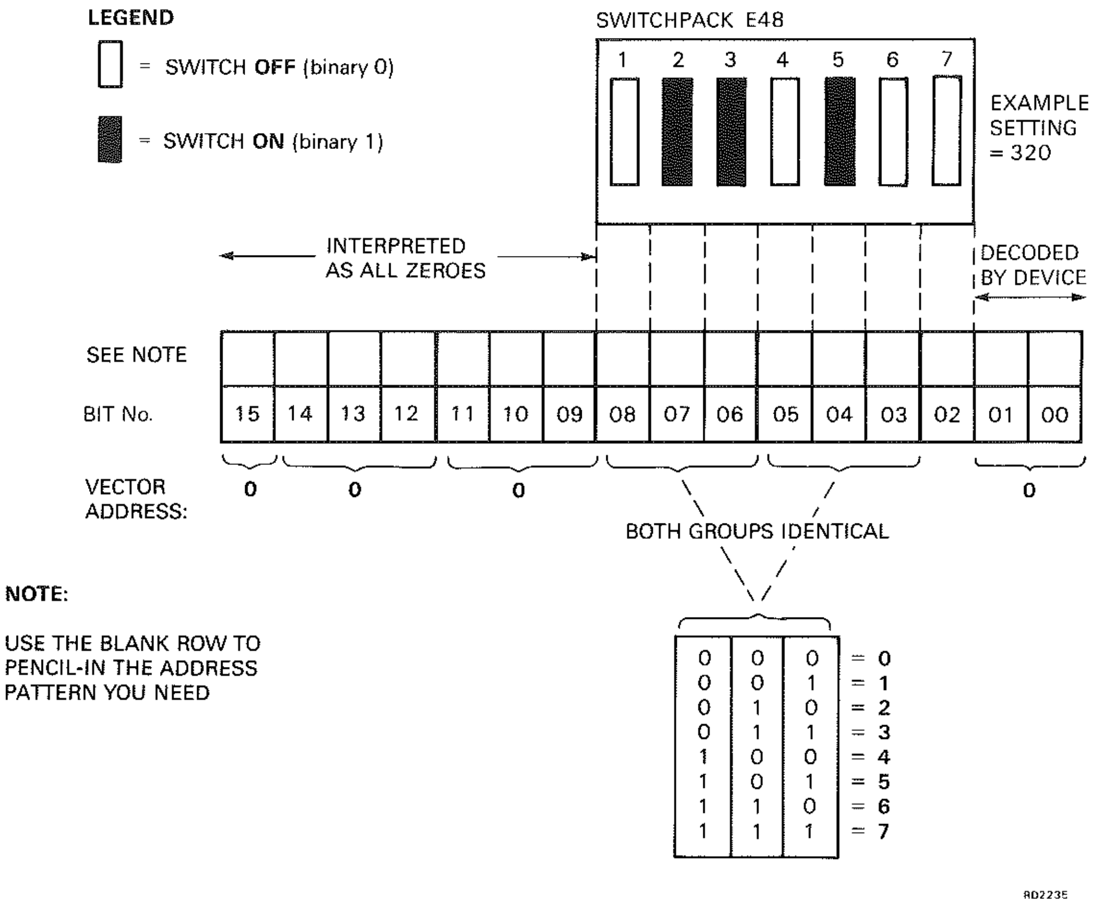
3.2 INITIALIZATION

3.2.1 Display Resolution Selection

Two switches (9 and 10 at E48) are provided on the module to allow the selection of one of the four display resolutions. These are:

- 640 × 480 pixels
- 640 × 240 pixels
- 512 × 512 pixels
- 512 × 256 pixels

A full description of these switches is given in the *VSV21 Installation Manual.* Make sure that switch 8 is ON for normal operation (see the installation manual for the use of switch 8 when running diagnostics).



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

Figure 3-3 Vector Address Selection Guide

3.2.2 Initialization and Self-Test

A self-test is performed at power-on. If the self-test fails, initialization stops. If the VSV21 fails the self-test routine, the board must be changed by a field service engineer. There is no user-level maintenance for the VSV21.

When the self-test is passed, the VSV21 initializes to a state determined by the contents of the non-volatile RAM. The default state is the console emulation mode (see Section 3.2.4). The user can alter the contents of the non-volatile RAM, and therefore the initialization state, by using the VCP utility (see Section 3.3).

NOTE: The NVRAM has a limited life, so it must not be used for unnecessary operations, such as backing up. Temporary changes are not dealt with by the NVRAM, but any use of the IPERM switch will cause a permanent rewrite to the NVRAM. The estimated life of the NVRAM is considered to be a minimum of 1000 operations.

3.2.3 Error Logging

Error logging occurs during self-test and normal operation of the VSV21. Non-catastrophic hardware faults can occur which do not affect the function of the device. They are logged at system level. Error messages are described in the *VSV21 Installation Manual.*

A full description of the self-test is given in the *VSV21 Installation Manual.*

3.2.4 Console Emulator

In this mode the VSV21 operates as a limited VT220 terminal. If the VSV21 is connected to the host through the host console port, the system can be booted from the VSV21. If the host system is one of those listed (see Chapter 1) the VSV21 board provides sufficient VT220 features (such as clear screen, screen addressing, and keypad operation) to allow the system to be booted. Once this has occurred the on-board software can be loaded into the VSV21.

If an attempt is made at typing a non-formatted file on the VSV21 in host console mode, the VSV21 can hang up. The only exit from such a state is by resetting the system.

3.2.5 Start-up Procedure

The sequence given below should be followed to allow you to get started.

1. RUN VCP or
   RUN SYS$SYSTEM:VCP – this allows you to issue various commands as detailed in Section 3.3

2. LOAD KERNEL – this loads part of the on-board control software, and should be loaded first

3. LOAD JSTICK or
   LOAD PGSTICK or
   LOAD DECTAB – loads pointing device driver if required

4. LOAD TRANSP – loads the transparent port driver if required

5. LOAD VIVID or
   LOAD VS11EM – the interpreter which translates graphics instructions and data into pictures on the screen

6. ^Z – return to the Operating System

The VSV21 will now be interpreting VIVID or VS11.

Remember that device drivers are loaded into upper memory locations and the interpreters are loaded into lower memory locations. If two conflicting device drivers are loaded in sequence (for example, PGSTICK and JSTICK), then the last loaded driver will over-write the previously loaded one. If another item – such as the attributes – is saved or loaded between the two device drivers, then the previous device driver is logically deleted, but physically continues to occupy memory. This will reduce the amount of memory available for segments.

NOTE: If you load VIVID or VS11M before loading any pointing device driver(s), it will not be possible to subsequently display or modify the pointing table via VCP, and other problems may arise.

3.3 THE VCP UTILITY

The VCP utility provides the user with control of the VSV21 graphic devices within the system, from either command level or command files (such as system startup files). The VSV21 can be configured independently from routines in System Support or from the VCP utility (VCP and System Support offer the same functions).

The VCP utility accepts commands directly from the command terminal or from the command file. Error messages returned by the VCP utility are output to the invoking terminal device on RSX systems. On VMS systems, error messages are output to the logical SYS$OUTPUT and are compatible with RMS. RSX and VMS systems share the same command format for VCP. This allows easy migration of user control packages between systems.

The VCP utility runs as a user process under the operating system.

3.4 THE VCP COMMANDS

This section describes the commands which you can issue to VCP, and the basic format of the command line. For a description of the individual commands, refer to Sections 3.4.1 to 3.4.6.

On-line HELP is available. To see the format of any command, type HELP and the command. The Help facility prompts for parameter-names and supplies the range of values for each parameter.

3.4.1 Types of VCP Commands

You can command the VCP to perform the following operations:

- Download code from the host

  LOAD [dev] filnam

- Download segments from the host
  VIVID__LOAD__SEG filnam

- Set a VSV21 parameter
  SET__PARAMETER [dev] ctab[/perm] dsp gval[/perm]

- Display configuration and status
  TABLE [dev] ctab
  STATUS dev

- Dump VSV21 memory to a host file
  DUMP filnam

Where:

filnam	=	filename
dev	=	device specification
ctab	=	the table of options to be used
dsp	=	a display parameter
gval	=	a value associated with dsp
perm	=	a qualifier which denotes whether the parameter is to be set permanently, or only for this session

Parameters in a command line must be separated by one or more spaces, commas, or tab characters. The format of these parameters is described in Sections 3.5 and 3.6.

3.4.2 Command Syntax

VCP commands are of the form:

```
command__name [dev] parameters
```

Where:

command__name is one of those listed in Section 3.4.1. You can shorten a command or parameter to the smallest number of characters necessary to identify it uniquely. For example, SET__PARAMETER can be shortened to SET.

dev is the device specification, which can be one of the following on RSX-11M-PLUS and Micro/RSX systems:

VS0:
VS1:
VS2:
VS3:

If the system has just been loaded, VS0: is the default specification. Otherwise, the last device specification entered is the default.

On MicroVMS systems, dev is one of the following:

VSA0:
VSA1:
VSA2:
VSA3:

You must enter additional parameters for most commands. There are three categories of parameter as follows.

ctab = a string of up to 16 characters

dsp = a string of up to 16 characters

gval = a parameter containing data of the following types:

- Character string
- Decimal integer
- Hexadecimal string

The format of these parameters is described in Section 3.6.

3.5 DOWNLOADING CODE OR DATA FROM A HOST FILE

3.5.1 Downloading Software

The kernel is downloaded to the VSV21 at the start of the session. The LOAD command is used to load this routine and other on-board software, including the VIVID interpreter, emulators and device drivers.

The general format of the LOAD command is:

```
LOAD dev filnam
```

Where:

dev = device specification. This can be one of the following:

VS0:
VS1:
VS2:
VS3:

filnam = the name, without extension and UIC, of the file to be loaded. It can be one of the following:

KERNEL	Downloaded at the start of the session
VIVID	VIVID interpreter
VS11EM	VSV11 emulation software
VT220EM	VT220 emulation software
JSTICK	MSI device driver
PGSTICK	Penny and Giles device driver
DECTAB	Digitizing tablet driver
TRANSP	Transparent device driver

The files are located on the host system as follows:

- RSX-11M-PLUS Directory [3,54]
- Micro/RSX Directory [3,54]
- MicroVMS SYS$LIBRARY

The host system treats these filenames as unprivileged. Future versions of the system may include additional unprivileged filenames, as well as a list of privileged filenames.

3.5.2 Downloading Segments

VIVID segments are downloaded to the VSV21 with the VIVID_LOAD_SEG command. The general form of the command is:

```
VIVID_LOAD_SEG filnam
```

where filnam is the name of a host file containing the segments. If no file extension is given, the default .VIV is assumed.

If the VIVID default font on the VSV21 module has been deleted or overwritten, it can be reloaded from the host. The parameter filnam is as follows:

- RSX-11M-PLUS and Micro/RSX – [3,54]DFONT
- MicroVMS – SYS$LIBRARY:DFONT

3.6 SETTING VSV21 DISPLAY PARAMETERS

The main function of the VSV21 commands is to allow the user to set VSV21 display parameters, such as the color definitions and the peripheral device characteristics.

The general format of the command for setting VSV21 parameters is:

```
SET_PARAMETER [dev] ctab[/perm] dsp gval[/perm]
```

Where:

dev = device specification, as defined in Section 3.4.2

ctab = a parameter chosen from the following list:

GRAPHICS_DISPLAY
VT220_DISPLAY
SERIAL_KEYBOARD
SERIAL_HOST
SERIAL_POINTING
SERIAL_TRANSP
KEYBOARD
HOST
POINTING

dsp = a display parameter, for example:

```
TESTCHART
SPLIT__SCREEN
LINES__SPLIT
A__COLORS
B__COLORS
C__COLORS
D__COLORS

COUNT__BLINK
INDEX__BLINK
A__BLINK__COLORS
B__BLINK__COLORS
C__BLINK__COLORS
D__BLINK__COLORS
512__TIMINGS
640__TIMINGS
```

gval = a numeric or logical value associated with dsp

perm = a switch which decides if the parameter is to remain stored in non-volatile memory when the VSV21 is detached from the host. You can set it to one of the two following values.

PERMANENT	If the parameter is stored
NOPERMANENT	If the parameter is not stored (this is the default)

You can enter up to 511 characters values using one SET__PARAMETER command, as follows.

```
SET__PARAMETER [dev] ctab[/perm]   dsp gval[/perm]-
                                   dsp gval[/perm]-
                                   dsp gval[/perm]-
                                   dsp gval[/perm]
```

If the value of perm which qualifies ctab is set to PERMANENT, all values of dsp and gval in the command line which are not qualified by /NOPERMANENT are put in permanent storage. For example, suppose a SET__PARAMETER command has been included in the program as follows:

```
SET GRAPHICS/PERM VT220__COLORS 5 4 2 3 1
                  SPLIT__SCREEN YES/NOPERM
                  LINES__SPLIT 6
```

The VT220 parameters and the COUNT value are put in permanent store, while the SPLIT__SCREEN parameter is not.

The range of available SET_PARAMETER commands is summarized in Table 3-1. The parameters are described in the sections whose numbers are given in brackets.

Table 3-1 Parameter Values for SET_PARAMETER Commands

ctab	dsp	gval
GRAPHICS_DISPLAY	TESTCHART	DISPLAY/NODISPLAY
(3.7)	A_COLORS	24 hex chars
	B_COLORS	24 hex chars
	C_COLORS	24 hex chars
	D_COLORS	24 hex chars
	COUNT_BLINK	integer, range 0 to 16
	INDEX_BLINK	1 to 16 integers
	A_BLINK_COLORS	1 to 24 hex chars
	B_BLINK_COLORS	1 to 24 hex chars
	C_BLINK_COLORS	1 to 24 hex chars
	D_BLINK_COLORS	1 to 24 hex chars
	512_TIMINGS	1 to 20 hex chars
	640_TIMINGS	1 to 20 hex chars
VT220_DISPLAY	SPLIT_SCREEN	YES/NO
(3.7)	LINES_SPLIT	2 to 12
	VT220_COLORS	5 integers, range 0 to 15
SERIAL_KEYBOARD	LINE_STATUS	ENABLE/DISABLE
(3.9)		
SERIAL_HOST and	LINE_STATUS	ENABLE/DISABLE
SERIAL_POINTING	MODE_PARITY	WITH/FORCE/NONE
(3.9)	TYPE_PARITY	EVEN/ODD
	BITS_CHAR	7 or 8 for SERIAL_HOST; 5 to 8 for SERIAL_POINTING
	TX_SPEED	75 to 9600
	RX_SPEED	75 to 9600
SERIAL_TRANSP	LINE_STATUS	ENABLE/DISABLE
(3.9)	MODE_PARITY	WITH/FORCE/NONE
	TYPE_PARITY	EVEN/ODD
	BITS_CHAR	integer, range 5 to 8
	TX_SPEED	75 to 9600
	RX_SPEED	75 to 9600
	XON_CODE	2 hex chars
	XOFF_CODE	2 hex chars
HOST	MAX_CHAR	integer, range 1 to 80
(3.9)	UK_KEYBOARD	HASH/POUND
	SCROLL	SMOOTH/JUMP
	WRAP	YES/NO
	TAB_SETTINGS	1 to 20 hex chars
KEYBOARD	KEY_CLICK	ON/OFF
(3.10)	BELL	ON/OFF
	AUTOREPEAT	ON/OFF
	MODE	DP/TYPEWRITER
	BITS	7 or 8
	NATIONALITY	language name

Table 3-1 Parameter Values for SET__PARAMETER Commands

ctab	dsp	gval
POINTING (3.11)	X__SQUARE	YES/NO
	X__FACTOR	integer, range 0 to 32767
	X__SHIFT__COUNT	integer, range 0 to 15
	Y__SQUARE	YES/NO
	Y__FACTOR	integer, range 0 to 32767
	Y__SHIFT__COUNT	integer, range 0 to 15

3.7 SETTING UP THE GRAPHICS DISPLAY

You can set the following display characteristics:

- Whole screen or split screen display
- The range of colors in the display
- The range of colors to be blinked
- The number of colors to be blinked
- The monitor timings

To set these you use SET__PARAMETER GRAPHICS__DISPLAY, which is one of the commands for setting VSV21 parameters (Section 3.4.1).

The general format of the GRAPHICS__DISPLAY command is as follows:

```
SET__PARAMETER [dev] GRAPHICS__DISPLAY[/perm] dsp gval[/perm]
```

Where:

dev = device specification, as defined in Section 3.4.2

dsp = a display parameter

gval = a value associated with dsp

perm = a switch as described in Section 3.6

SET__PARAMETER can be conveniently abbreviated to SET.

If the graphics software is downloaded, you can choose whether or not to display a test chart by using the TESTCHART command as follows:

```
SET GRAPHICS__DISPLAY TESTCHART DISPLAY
```

or

```
SET GRAPHICS__ DISPLAY TESTCHART NODISPLAY
```

3.7.1 The Color Look-Up Table (CLUT)

The VSV21 holds the default VSV11 CLUT in PROM. You can use VCP to put this CLUT in NVRAM, and to change the values.

3.7.1.1 Default Colors – The default contents of the CLUT are the same as the default VSV11 colors. The default colors are as follows:

Position	Color	Position	Color
0	Black	8	Mid green
1	Blue	9	Pale blue
2	Red	10	Light orange
3	Violet	11	Pink
4	Dark green	12	Green
5	Mid blue	13	Magenta
6	Orange	14	Yellow
7	Pale violet	15	White

3.7.1.2 Setting Up the CLUT – If you want to use another set of colors in the VSV21 display, you can choose sixteen from a palette of 4096 colors. The colors parameters are used to define these colors. The colors you specify will be inserted in the color look-up table on the VSV21 module.

Six hexadecimal characters are used to define each of the 16 colors, so for each display a total of 96 characters define the colors. To allow the 96 color and intensity characters to be entered as a number of shorter, more convenient strings, the following four colors parameters are used.

A__COLORS gval	Defining colors 0 to 3
B__COLORS gval	Defining colors 4 to 7
C__COLORS gval	Defining colors 8 to 11
D__COLORS gval	Defining colors 12 to 15

Where gval is a 24-character hexadecimal string which defines four colors.

The characters are arranged as follows. Each of the sixteen available colors requires six hexadecimal characters, arranged in three pairs, to define it. The three pairs define the intensity of red, green, and blue respectively, and the system combines the data to generate a composite color.

The primary colors are coded as follows.

1	Red
2	Green
4	Blue

You can allocate an intensity value to any of these primary color codes. This value ranges from 0 (not included) to 15 (maximum intensity), that is, from 0 to F in hexadecimal notation. Black is defined by allocating zero intensity to each of the three colors, and white by allocating maximum intensity to each.

An error message is returned if the number of color characters entered is not a multiple of 18. No error message is returned if the color-intensity pair does not begin with 1, 2, or 4, but incorrect output will result.

Similarly, errors will result if VCP commands in your program attempt to extract data from unfilled positions in the CLUT. For example, if colors 0 to 7 are defined by A__COLORS and B__COLORS parameters, while colors 8 to F are left undefined, attempts to read colors 8 to F will produce unpredictable results.

3.7.1.3 CLUT Examples – To generate red, the three pairs are 1F, 20, 40. They are entered in a continuous string: 1F2040. In this case, green and blue have zero intensity, so only red is produced.

Yellow could be generated by equal intensities of green and blue, omitting red: 102F4F. A less intense yellow would be 102525, and turquoise could be produced by combining a little green with more blue: 10264B.

Black is 102040. White is a combination of the primary colors in maximum intensities: 1F2F4F. Shades of gray through to white are generated by intermediate intensity values which are the same for each primary color.

Suppose you want to put the following nine colors in the CLUT:

Position	Color	Position	Color
0	Green	5	Black
1	Cyan	6	Magenta
2	Turquoise	7	White
3	Red	8	Blue
4	Cyan		

You can do this by using A__COLORS, B__COLORS, and C__COLORS:

```
SET GRAPHICS__DISPLAY   A__COLORS   102F4010284810264C1F2040

SET GRAPHICS__DISPLAY   B__COLORS   1028481020401A204A1F2F4F

SET GRAPHICS__DISPLAY   C__COLORS   10204F
```

3.7.2 Defining Blink Colors

The four BLINK__COLORS parameters allow the user to define the colors to be shown blinking on the display, and store them in a blink colors look-up table (BCLUT). You can enter sixteen such colors by using the codes for primary colors and their relative intensities in the way described for the colors parameters in Section 3.7.1.2.

A__BLINK__COLORS	gval	Defining colors 0 to 3
B__BLINK__COLORS	gval	Defining colors 4 to 7
C__BLINK__COLORS	gval	Defining colors 8 to 11
D__BLINK__COLORS	gval	Defining colors 12 to 15

Where gval is a 24-character hexadecimal string which defines four colors.

3.7.3 Forming Pairs of Blink Colors

3.7.3.1 Defining the Range of Blink Colors – You can enter your choice of alternating colors in the form of a string of integers by using INDEX__BLINK. The value of each integer refers to the position of a color you have already defined in the color look-up table. The position of each integer in the string refers to the position of a color already defined in the blink color look-up table.

Example:

Suppose that the colors and blink colors tables have been set up as follows:

Position	Color	Blink Color
0	Green	Orange
1	Yellow	Indigo
2	Turquoise	Turquoise
3	Red	Black
4	Yellow	Red
5	Black	White
6	Mauve	Yellow
7	White	Blue

Then the command line:

```
SET GRAPHICS__DISPLAY INDEX__BLINK 6 5 7 4
```

Means that:

Blink color 0 alternates with color 6 – orange/mauve

Blink color 1 alternates with color 5 – indigo/black

Blink color 2 alternates with color 7 – turquoise/white

Blink color 3 alternates with color 4 – black/yellow

3.7.3.2 Selecting Blink Colors from the Range – The number of colors for which blinking is enabled is entered using COUNT__BLINK. COUNT__BLINK allows the user to enter the number of colors to be blinked on the display. Its associated gval is a hexadecimal integer between 0 (no colors blinked) and F (all colors blinked).

Example:

```
SET GRAPHICS__DISPLAY COUNT__BLINK 3
```

This allocates the first three color pairs defined by INDEX__BLINK for use in the VSV21 display. The remaining colors in the blink colors table are ignored.

3.8 SELECTING VT220 EMULATION

The VSV21 can emulate a VT220 to enable you to use it as a terminal on the host system. There are two kinds of VT220 emulation.

- Full-screen text

- Text window (the area above the window shows VSV21 graphics)

You use the SET VT220__ DISPLAY command to:

- Switch between full-screen text and a text window

- Select the colors to use in VT220 emulation.

The general format of the SET VT220__DISPLAY command is:

```
SET__PARAMETER [dev] VT220__DISPLAY[/perm] dsp gval[/perm]
```

Where:

dev = device specification, as defined in Section 3.4.2

dsp = a display parameter

gval = a value associated with dsp

perm = a switch as described in Section 3.6

For convenience you can abbreviate SET__PARAMETER to SET.

3.8.1 Splitting the Screen

You can divide the VSV21 screen display into two sections.

1. An upper section consisting of graphics display
2. A lower section consisting of host text output

SPLIT__SCREEN sets the screen mode to NORMAL or SPLIT. Its associated parameter gval is one of the following.

YES Splits screen (graphics and text display)

NO Does not split screen (graphics display only)

Example:

```
SET__PARAMETER VT220__DISPLAY SPLIT__SCREEN YES
```

When a new SPLIT__SCREEN value is entered, the terminal VSV11 emulation screen output is automatically reset; that is, the text area is cleared.

You can set the number of text lines shown under the graphics display at between two and twelve lines by using LINES__SPLIT. The default is four lines.

Example:

```
SET__PARAMETER VT220__DISPLAY LINES__SPLIT 2
```

produces a two-line text output under the graphics display.

3.8.2 Choosing Colors for VT220 Emulation

VT220__COLORS allows the user to decide which of the colors defined by the colors parameters are to be used in the VT220 emulation, and how these colors are to be used. Five colors must be specified, using the positions of those colors in the CLUT, separated by commas, spaces, or tabs as input. The function of each color in the display is defined by its position in the VT220__COLORS string as follows:

Position	Function
1	Normal
2	Background
3	Blink
4	Highlight
5	Blink and highlight

Example:

```
SET VT220__DISPLAY VT220__COLORS 5 4 6 0 7
```

This uses the color allocations described in Section 3.7.3 to define black text on a yellow background. The color of the text blinks to turquoise. The text is highlighted in yellow. Blinking lines are highlighted in pale red.

3.8.3 Setting Video Signal Timings

If you are using the VSV21 with a monitor not recommended by DIGITAL, you may need to set the monitor display characteristics during hardware installation.

The 512__TIMINGS and 640__TIMINGS commands set the VSV21 characteristics to match monitors with either 512-pixel or 640-pixel wide displays, used respectively for VSV11 emulation and VT220 emulation.

Each command defines the following:

- Scan time
- Sync width
- Display start position
- Display width

both vertically and horizontally.

Each command has eight parameters, grouped in five 4-character hex strings. The parameter groups are as follows:

Group 1:

- Horizontal Scan. Horizontal scan time, in units of memory cycles. Range 01 — FF.

- Horizontal Sync Width. Horizontal sync active low time, in units of memory cycles. Range 01 — 1F.

Group 2:

- Horizontal Display Start. Horizontal display start interval, in units of memory cycles. Range 00 — FF.

- Horizontal Display Width. Horizontal display period for one raster. Range 00 — FF.

Group 3:

- Vertical Scan. Vertical scan cycle period in units of rasters. Range 0000 — 0FFF.

Group 4:

- Vertical Display Start. Vertical display start interval in units of memory cycles. Range 00 — FF.

- Vertical Sync Width. Vertical low pulse width in units of rasters. Range 00 — 1F.

Group 5:

- Vertical Window Width. Vertical display period of window screen in units of rasters. Range 0000 — 0FFF.

3.9 CONFIGURING THE HOST SERIAL CONNECTION

3.9.1 Setting Up the Host Line Characteristics

You set up the communication characteristics by using the following commands.

```
SERIAL_KEYBOARD
SERIAL_HOST
SERIAL_POINTING
SERIAL_TRANSP
```

The SERIAL_KEYBOARD command enables or disables the line to the host. You set it either to ENABLE or to DISABLE.

The SERIAL_HOST command sets up the line characteristics of the link to the host. The link resembles the link to a VT220, and the SERIAL_HOST command enables you to perform the same functions as those defined by the VT220 SET-UP key. For further explanation of these functions, see the *VT220 User's Guide (EK-VT220-UG-002)*.

The SERIAL_HOST command defines:

- Line status
- Length of input and output buffers
- Mode and type of parity
- Number of bits per character
- Baud rates of transmit and receive.

The format of the SERIAL_HOST table command is:

```
SET_PARAMETER dev SERIAL_HOST[/perm] dsp gval[/perm]
```

Where:

dev = device specification as defined in Section 3.4.2

dsp = a display parameter. This can be one of the following.

```
LINE_STATUS
MODE_PARITY
TYPE_PARITY
BITS_CHAR
TX_SPEED
RX_SPEED
```

gval = a value associated with dsp

perm = a switch as described in Section 3.6

A description of the dsp parameters follows.

- LINE_STATUS defines the status of the line. Its associated gval can be either ENABLE or DISABLE.

- MODE_PARITY defines the parity mode. The parameter gval can be either WITH, FORCE, or NONE.

- TYPE_PARITY defines the parity type. The parameter gval can be either EVEN or ODD.

- BITS_CHAR defines the number of bits in each character. The parameter gval can be either 5, 6, 7, or 8.

- TX_SPEED and RX_SPEED define the transmit and receive baud rates respectively. The parameter gval can be one of the following:

  75
  110
  134
  150
  300
  600
  1200
  1800
  2000
  2400
  4800
  9600

The dsp parameters defined here are described in more detail in the *VT220 User's Guide (EK-VT220-UG-002).*

You use the SERIAL_POINTING command to set the line characteristics of the serial port to which the peripheral device is connected.

The format of the SERIAL_POINTING command is as follows:

```
SET_PARAMETER dev SERIAL_POINTING[/perm] dsp gval[/perm]
```

Where:

dev = device specification, as described in Section 3.4.2

dsp = a display parameter. This can be one of the following.

BITS_CHAR
LINE_STATUS
MODE_PARITY
TYPE_PARITY
TX_SPEED
RX_SPEED

gval = a value associated with dsp

perm = a switch as described in Section 3.6

Example:

To define the number of bits in a character, use BITS__CHAR. You can give BITS__CHAR a value of 5, 6, 7, or 8.

```
SET SERIAL__POINTING BITS__CHAR 8
```

This defines an 8-bit character.

You use the SERIAL__TRANSP command to set the characteristics of peripheral device. This can be any serial device, such as a keyboard or printer, for which there is an installed driver.

The parameters are as for SERIAL__POINTING, with the following additions:

dsp	gval
INPUT__BUFFER__LEN	0 to 32767
OUTPUT__BUFFER__LEN	0 to 32767
XON__CODE	two hex chars
XOFF__CODE	two hex chars

You can set the lengths of the input and output buffers according to the nature of the device. XON__CODE and XOFF__CODE are communications protocol characters for your own device.

3.9.2 Setting Up the Host Device

You enter the characteristics of the host device by using the HOST command. The command defines:

- Keyboard type
- Screen display type
- Scrolling mechanism
- Existence of wraparound
- Tab settings

The format of the command is:

```
SET__PARAMETER [dev] HOST[/perm] dsp gval[/perm]
```

Where:

dev = device specification, as defined in Section 3.4.2

dsp = a display parameter. This can be one of the following.

MAX_CHAR
UK_KEYBOARD
SCROLL
WRAP
TAB_SETTINGS

gval = a value associated with dsp

perm = a switch as described in Section 3.6

MAX_CHAR defines the maximum length of the line. The line can be up to 132 characters in length.

The UK keyboard type is defined by UK_KEYBOARD. It is one of the following.

HASH	enables the hash character
POUND	enables the pound sign

SCROLL defines the scrolling mechanism. It is one of the following.

SMOOTH	Continuous scroll
JUMP	Discrete scroll

WRAP specifies whether or not wordwrap is enabled. It is one of the following.

YES	Set wordwrap
NO	Cancel wordwrap

TAB_SETTINGS specifies the character position of the TAB setting.

It is entered as a number of hexadecimal characters. The position of the tabs is derived from the position of the '1' characters in the equivalent binary string.

3.10 SETTING UP KEYBOARD CHARACTERISTICS

You set up the keyboard by using the KEYBOARD command. The KEYBOARD command defines:

- Type of interface to the driver
- Key click

- Bell
- Autorepeat on or off.

The format of the KEYBOARD command is as follows:

```
SET_PARAMETER dev KEYBOARD[/perm] dsp gval[/perm]
```

Where:

dev = device specification as defined in Section 3.4.2

dsp = a display parameter. This can be one of the following:

```
KEY_CLICK
BELL
AUTOREPEAT
MODE
BITS_CHAR
NATIONALITY
```

gval = a value associated with dsp

perm = a switch as described in Section 3.6

3.10.1 Key Sound

KEY_CLICK decides whether or not the keys produce a click. You can set it either to ON or to OFF.

3.10.2 Terminal Bell

BELL is a switch controlling the terminal bell. You can set it either to ON or to OFF.

3.10.3 Autorepeat

AUTOREPEAT sets the autorepeat facility to ON or OFF.

3.10.4 Multinational Mode

MODE sets the mode to data processing (DP) or typewriter.

3.10.5 Number of Bits

BITS__CHAR sets the keyboard to 7-bit or 8-bit, depending on the operating system type and version.

3.10.6 Character Sets

NATIONALITY sets the VSV21 to receive the character set of the keyboard being used. The VSV21 can handle the following character sets:

US
UK
SWEDISH
DUTCH
FLEMISH
CANADIAN__FRENCH
DANISH
FINNISH
GERMAN
ITALIAN
SWISS__FRENCH
SWISS__GERMAN
NORWEGIAN
FRENCH
SPANISH

3.11 SETTING UP POINTING DEVICE CHARACTERISTICS

You set up the pointing device characteristics by using the POINTING command. The POINTING command defines the sensitivity of the pointing device. Note that a pointing device driver must be loaded (Section 3.5.1) before you can set up the characteristics.

The format of the command is as follows:

```
SET__PARAMETER dev POINTING[/[perm] dsp gval[/perm]
```

Where:

dev = device specification as defined in Section 3.4.2

dsp = a display parameter. This can be one of the following:

X_SQUARE
X_FACTOR
X_SHIFT_COUNT
Y_SQUARE
Y_FACTOR
Y_SHIFT_COUNT

gval = a value associated with dsp. It can be set to YES or NO.

perm = a switch as defined in Section 3.6

3.11.1 Speed Sensitivity

X_SQUARE and Y_SQUARE set the sensitivity of the cursor to the speed at which the user moves the pointing device in the X and Y planes respectively. The sensitivity in the X or Y plane is proportional to the speed squared, if the parameter is set to YES; that is, the distance the cursor moves across the screen is greater per unit movement of the pointing device at higher user speeds. Cursor movement is in linear proportion to pointer speed if it is set to NO.

3.11.2 Sensitivity Factors

X_FACTOR and Y_FACTOR are factors by which sensitivity can be increased. They can be set to any integer value in the range 1 to 32767.

3.11.3 Shift Counts

The shift count is used to generate non-integer sensitivity factors (Section 3.11.2). A shift count value is combined with the X_FACTOR and Y_FACTOR parameters in the following calculation:

$$\text{FACTOR} = \text{FACTOR}/2^{\text{SHIFT}}$$

Where:

FACTOR = value of X_FACTOR or Y_FACTOR

SHIFT value of SHIFT_COUNT parameter (range 0 to 15)

This allows you to control sensitivity precisely.

3.12 DISPLAYING CONFIGURATION AND STATUS

To display the stored parameter values, use the TABLE command:

```
TABLE [dev] ctab
```

Where:

dev = device specification as defined in Section 3.4.2

ctab = one of the strings listed in Section 3.6

The system responds by displaying the parameter values.

To see the status of any VSV21 device, enter:

```
STATUS dev
```

Where:

dev = device specification as defined in Section 3.4.2

CHAPTER 4
MODES OF OPERATION

4.1 SCREEN MODES

The VSV21 can be used in full-screen VT220 mode, full-screen graphics mode, or split-screen graphics mode (where only the bottom four lines of the screen are used to display host serial output).

When the VSV21 is in full-screen graphics mode and some serial output is received from the host, it switches automatically to split-screen graphics mode. Split-screen mode can also be entered by a software VCP command. When the VSV21 is in split-screen mode, it can not be switched back to full-graphics mode by software. It is necessary to again down-load the emulator in order to return to full-graphics screen. The user, however, can switch modes by hardware using the F4 key.

4.1.1 VT220 Emulation

To load the VT220 mode emulator, use the VCP command:

```
LOAD device__spec VT220
```

4.1.2 Supported VT220 Functions

The VSV21 VT220-emulator offers only limited VT100 functionality with Advanced Video Option (AVO), in 80-column ANSI mode. It provides the following functions.

- Full cursor positioning
- Scrolling regions
- Selective erase
- Alternate keypad mode
- Character attributes
- Selectable character set

- Special characters and line-drawing characters
- XON/XOFF host synchronization
- 8-bit multi-national character set

4.1.3 Unsupported VT220 Functions

The following functions are NOT supported.

- The SET-UP key, and features controlled by it
- VT52 emulation
- 132 columns
- Alternate keypad mode
- XON/OFF host synchronization
- Double-height double width lines
- Support for a down-line loadable character set
- Printer facility (similar to VT102)

If an attempt is made at typing a non-formatted file on the VSV21 in host console mode, the VSV21 can hang up. The only exit from such a state is by resetting the system.

4.1.4 The Keyboard

An LK201 keyboard is needed for VT220 emulation. Refer to Section 2.1.3.2 for keyboard versions supported by the VSV21. The keyboard pattern is the same as that used by the VT220 in VT100 mode. That is, the top row of keys is defined as follows.

F1	Hold Screen (No-scroll)	F11	Escape
F2	Not used	F12	Backspace
F3	Not used	F13	Linefeed
F4	Swap Display Mode	F14	Not used
F5	Break Swap Display Mode	Help	Not used
F6	Not used	Do	Not used
F7	Not used	F17	Not used
F8	Not used	F18	Not used
F9	Not used	F19	Not used
F10	Not used	F20	Not used

4.2 VSV11 EMULATION

In this mode, the VSV21 emulates a minimum configuration VSV11 option consisting of:

- One M7064 processor board
- One or two M7062 memory boards
- One M7061 sync generator/cursor control board

The VSV21-based VSV11 emulator is compatible with the standard VS11 device driver. RSX-11M support is not provided.

4.2.1 Switching to VSV11 Mode

To load the VSV11 emulation software, you use the VCP command:

```
LOAD device__spec VS11EM
```

4.2.2 VSV11 Summary

Full details of the VS11 register structure, device driver, and display list commands are given in the *VS11 Software Driver Guide (AA-K793C-TE)*. The following is a summary of VSV11 features.

4.2.2.1 Device Registers – The VSV11 has four registers available to an application through the device driver. These registers are:

- Display program counter
- Display status register
- Display X register
- Display Y register.

4.2.2.2 Device Driver Commands – The following commands to the device driver are available.

```
IO$START   (main address, length, [display start], [auxiliary address, length])

IO$TOUT    (seconds)

IO$STOP
```

IO$RESUME (main address, length, [display start],
[auxiliary address, length])

IO$READSTATUS (buffer)

IO$WAITSWITCH (seconds)

4.2.2.3 Display List Contents – The following VSV11 display list functions are available:

Set CHARACTER mode [and load pixel data]
Set SHORT VECTOR mode [and load pixel data]
Set LONG VECTOR mode [and load pixel data]
Set ABSOLUTE POINT mode [and load pixel data]
Set GRAPH/HISTOGRAM X mode [and load pixel data]

Set GRAPH/HISTOGRAM Y mode [and load pixel data]
Set RELATIVE POINT mode [and load pixel data]
BIT-MAP-0 (4-bit pixel data)
BIT-MAP-0 (8-bit pixel data)
BIT-MAP-1 (4-bit pixel data)

BIT-MAP-1 (8-bit pixel data)
Set RUN-LENGTH mode
JOYSTICK STATUS
Set HISTOGRAM BASE
Set CHARACTER BASE

JUMP
JUMP-TO-SUBROUTINE
DMA PIXEL READBACK
Display NO-OP
MARKER NO-OP

LOAD STATUS A
LOAD STATUS A__STOP
CLEAR PIXEL DATA INHIBIT
SET PIXEL DATA INHIBIT
LOAD GRAPH/HISTOGRAM ENVIRONMENT

LOAD CURSOR COORDINATES
LOAD EXTENDED JOYSTICK
LOAD STATUS C

INDEX

VSV21 Version 2.0
User Guide
AZ-FV70B-TC

READER'S COMMENTS

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

1. **Which of the following most closely describes your job?**

(a) Administrative support (d) Scientist/Engineer (g) Educator/Trainer
(b) Programmer/Analyst (e) Systems Manager (h) Computer Operator
(c) Software support (f) Sales (i) Other ______________

1 ⊂a⊃ ⊂b⊃ ⊂c⊃ ⊂d⊃ ⊂e⊃ ⊂f⊃ ⊂g⊃ ⊂h⊃ ⊂i⊃

2. **How many years of experience do you have with computers?**

(a) Less than 1 (b) 1 to 3 (c) 4 to 6 (d) 7 to 9 (e) 10 or more

2 ⊂a⊃ ⊂b⊃ ⊂c⊃ ⊂d⊃ ⊂e⊃

3. **What did you like *most* about this manual?**

4. **What did you like *least* about this manual?**

5. **How do you rate this manual?**

Indicate your opinion of the quality of the manual. For each aspect of quality, darken your response on the five-point scale, where (1) = POOR and (5) = EXCELLENT

(a) Accuracy	⊂1⊃	⊂2⊃	⊂3⊃	⊂4⊃	⊂5⊃
(b) Completeness	⊂1⊃	⊂2⊃	⊂3⊃	⊂4⊃	⊂5⊃
(c) Usefulness of Examples/Figures	⊂1⊃	⊂2⊃	⊂3⊃	⊂4⊃	⊂5⊃
(d) Clearness of Language	⊂1⊃	⊂2⊃	⊂3⊃	⊂4⊃	⊂5⊃
(e) Helpfulness of Index/Table of Contents	⊂1⊃	⊂2⊃	⊂3⊃	⊂4⊃	⊂5⊃
(f) Consistency in Presenting Information	⊂1⊃	⊂2⊃	⊂3⊃	⊂4⊃	⊂5⊃
(g) Logical Organization	⊂1⊃	⊂2⊃	⊂3⊃	⊂4⊃	⊂5⊃
(h) Visual Appeal	⊂1⊃	⊂2⊃	⊂3⊃	⊂4⊃	⊂5⊃
(i) Relevance of Information	⊂1⊃	⊂2⊃	⊂3⊃	⊂4⊃	⊂5⊃
(j) Ease of Learning	⊂1⊃	⊂2⊃	⊂3⊃	⊂4⊃	⊂5⊃
(k) Ease of Use	⊂1⊃	⊂2⊃	⊂3⊃	⊂4⊃	⊂5⊃
(l) YOUR OVERALL IMPRESSION	⊂1⊃	⊂2⊃	⊂3⊃	⊂4⊃	⊂5⊃
(m) Quality Relative to Other Digital Manuals	⊂1⊃	⊂2⊃	⊂3⊃	⊂4⊃	⊂5⊃
(n) Quality Relative to Other Companies' Manuals	⊂1⊃	⊂2⊃	⊂3⊃	⊂4⊃	⊂5⊃

6. **List any errors you found in the manual. (Reference page, table, or figure numbers.)**

7. **Do you have any additional comments?**

Name ______________________ Company ______________________

Title ______________________ Department ______________________

Street ______________ City ____________ State/Country ____________ Zip ____________

Telephone No. □ Date □

Site Code: RE

FOLD HERE AND TAPE

digital™

No Postage
Necessary
if Mailed in the
United States

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 33 MAYNARD, MA.

POSTAGE WILL BE PAID BY ADDRESSEE

DIGITAL EQUIPMENT CORPORATION
Educational Services/Quality Assurance
12 Crosby Drive BUO/E08
Bedford, MA 01730

FOLD HERE

VSV21 USER GUIDE ADDENDUM

AD-FV70B-T1

PAGE SECTION

6 1.4.3.1

When the VSV21 is running the VIVID interpreter (see the VSV21 Programmers Guide for more information), it can draw in either Display Priority or Drawing Priority.

Display Priority is the default mode. In this mode, the VSV21 gives priority to the display of information, and drawing into the pixel memory occurs at a lower priority.

The VIVID instruction SCREEN_BLANK can be used to set the priority. If the screen is blanked, then the VSV21 switches to Drawing Priority. This allows the VSV21 full access to the pixel memory for drawing, but the screen is blanked whilst the VSV21 is in this mode. The effect of this is that drawing occurs up to about 3 times faster on a high resolution screen, depending on the instructions used, in Drawing Priority. There is no throughput advantage with a low resolution system.

This mode would be used typically to redraw an entire screen rapidly, by blanking the screen, which selects Drawing Priority, drawing a new screen and then unblanking the screen which selects Display Priority. Since Drawing Priority blanks the screen it is not recommended to use it when redrawing small areas of the screen, as the user would notice the screen flash to black momentarily.

22 3.2.5

Refer to the VSV21 Programmers Guide for an explanation of segmentation in VIVID.

The NOTE: at the end of this section should be ignored.

23 3.4

VCP commands may be abbreviated to the shortest unambiguous string, for example the command:

```
VCP> LOAD KERNEL
```

may be abbreviated to the command:

```
VCP> L K
```

31	3.7.1.2	When setting colors, the parameter gval is a string of from 1 to 24 hexadecimal characters. The number of characters should be a multiple of 6, not 18 as stated.