APPLIED DAT

ELECTRONIC INST

APPLIED DATA SYSTEMS, INC.
ELECTRONIC INSTRUMENTATION AND CONTROLS

ROBERT KONDNER
ELECTRONIC ENGINEER

9811 MALLARD DR. SUITE 213
LAUREL, MD. 20708

(301) 953-9326

VectorScan 512

Programmer's Manual

Version 1.3

APPLIED DATA SYSTEMS, INC.

ELECTRONIC INSTRUMENTATION AND CONTROLS

9811 MALLARD DRIVE, SUITE 213
LAUREL, MARYLAND 20708

(301) 953-9326

VectorScan 512

Programmer's Manual

Version 1.3

TABLE OF CONTENTS

1. INTRODUCTION....2
1.1 SCOPE....2
1.2 MANUAL OVERVIEW....2
2. OVERVIEW....3
3. INSTALLATION....4
4. OPERATION....7
4.1 SETUP....7
4.2 OPERATIONAL MODES....7
4.3 CONTROL COMMAND SEQUENCES....8
4.4 GRAPHICS COMMANDS....9
4.4.1 INTRODUCTION TO GRAPHICS COMMANDS....9
4.4.2 SET PLOT LEVEL (LEV)....11
4.4.3 PLOT POINT (P)....11
4.4.4 DRAW VECTOR (VEC)....12
4.4.5 DRAW ARC (ARC)....12
4.4.6 DRAW CIRCLE (CIR)....12
4.4.7 FILL SCREEN (FIL)....13
4.4.8 FILL AREA (PFL)....13
4.4.9 CHANGE LEVELS (CHG)....14
4.4.10 GRAPHICS CURSOR ON (GCN)....15
4.4.11 GRAPHIC'S CURSOR OFF (GCF)....15
4.4.12 GRAPHIC'S CURSOR POSITION (GCP)....15
4.4.13 ENTER TEXT (TXT)....15
4.4.14 SCALE CHARACTERS (SCL)....16
4.4.15 CHARACTER CURSOR POSITION (CRS)....16
4.4.16 PRINTER SPOOLER MODE (PSM)....17
4.4.17 PRINT HARD COPY (PHC)....17
4.4.18 DEFINE SCAN LEVELS (DSL)....17
4.4.19 DEFINE COLOR SEQUENCE (DCS)....18
4.4.20 SYNCHRONOUS MODE ON (SON)....18
4.4.21 SYNCHRONOUS MODE OFF (SOF)....18
4.4.22 SET SCREEN BOTTOM (SSB)....19
4.4.23 INTERLACE ON (ION)....19
4.4.24 INTERLACE OFF (IOF)....19
4.4.25 SET USER CHARACTER (SUC)....19
4.4.26 DRAW USER CHARACTER (DUC)....19
4.4.27 ENABLE XON/XOFF HANDSHAKE (EXH)....20
4.4.28 DISABLE XON/XOFF HANDSHAKE (DXH)....20
4.4.29 PRINT STRING DIRECT (PSD)....20
4.4.30 LOAD MAP REGISTER (LMR)....21
4.4.31 BINARY POINT PLOT (BPP)....21
4.4.32 BINARY FILL SCREEN (BFS)....21
4.4.33 PRINT HORIZONTAL (PH)....22
4.4.34 PRINT VERTICAL (PH)....22
4.4.35 DOTTED LINE (DOT)....22
4.4.36 AND/XOR MODE (AXM)....23
4.4.37 CHANGE CONTROL CHARACTER (CCC)....23
4.4.38 LOAD VIDEO CONTROLLER (LVC)....23
APPENDIX 1- COMMAND SUMMARY....24
APPENDIX 2- ACCESS TO JUMPERS AND FUSE....25
APPENDIX 3- SERIAL CONFIGURATION JUMPER SETTINGS....26
APPENDIX 4- SUPPORTED GRAPHICS PRINTERS....28
APPENDIX 5- DOWNLOADABLE CHARACTER FORMAT....29
APPENDIX 6- CONTROL COMMANDS ('@' COMMANDS)....32
APPENDIX 7- CONNECTOR PINOUTS....34
APPENDIX 8- COLOR AND GRAY SCALE REPRESENTATIONS....37

Date of last revision: 2/23/84

SECTION 1 - INTRODUCTION

1.1 SCOPE

This document describes the characteristics, operation and maintenance procedures for the VectorScan 512.

1.2 MANUAL OVERVIEW

This manual consists of 4 major sections. Section 1 introduces the manual. An overview of the VectorScan is provided in section 2. Section 3 provides information on how the external hardware is connected to the various rear panel connectors. Section 4 describes the details of both the software and hardware interface. Also in section 4 are the detailed descriptions of the VectorScan commands.

Several appendices have been added for a quick and easy reference to VectorScan commands and configuration jumper settings. Other topics such as control commands and programmable shape formats are included.

Additional copies of this manual can be obtained from Applied Data Systems Inc.

SECTION 2 - OVERVIEW

The VectorScan 512 is a low cost, high resolution color graphics controller which operates under the supervision of a host computer. It is an intelligent, microprocessor based peripheral that accepts high level graphics commands to produce images on an RGB color monitor. The VectorScan connects easily to any computer with an RS232 serial port. A second port is provided so that the VectorScan may be connected "between" a computer and terminal. In this configuration, the RS232 port already available for the user console can be used for graphics. The VectorScan will be dormant until activated by a special control sequence.

This simple interface allows the VectorScan to be connected to almost any computer. With the high level graphic commands and a RS232 interface, the VectorScan is well suited for use over low speed telephone lines with a modem. Since all commands are transfered via the RS232 port in a printable ASCII format, both software and hardware can be transfered from one computer to another. Application programs can be written in any high level language without the use of special machine dependent subroutines.

The display is 512 pixels wide by 480 pixels high and is available in 4, 8, and 16 colors or gray scales. The colors are selectable from a palette of 16 color/intensity combinations via a mapping register. Sixteen gray scales are also selectable using the mapping register. The VectorScan provides both an RGB/sync interface for a color monitor and a composite video/sync interface for a monochrome monitor.

Also standard on the VectorScan is a Centronics parallel interface which may be connected to a low cost dot matrix printer. The VectorScan contains software to drive a number of common printers to provide a hardcopy output of the graphics display. A special printer spooler mode is provided to allow the 64k, 96k, or 128k byte memory to be used as a printer queue.

Commands for drawing points, lines, circles, and arcs are provided as well as screen and area fill commands. Text may be displayed in either a horizontal or vertical format using the built in ASCII character set. Text may be positioned anywhere on the screen and scaled in both the x and y dimensions. A downloadable shape table can be used to provide a user programmable character set. Each command is described in detail in section 4, OPERATION, and a concise command summary is contained in Appendix 1.

The VectorScan is controlled internally by an 8085 microprocessor and can accomodate up to 16k bytes of ROM and 4k bytes of RAM. This program memory is separate from the 64k to 128k bytes of video RAM. Arrangements for custom firmware or application programs for the VectorScan can be made by contacting Applied Data Systems Incorporated.

SECTION 3 - INSTALLATION

The VectorScan 512 has six connectors located on the rear panel. They are power, terminal, computer, color video, monochrome video, and printer. Figures 3-1 and 3-2 show two typical configurations of a system incorporating a VectorScan.

The VectorScan should be connected to a 110 to 120 VAC 50/60 Hz power source via the equipment cord supplied.

The rear panel connector labeled "COMPUTER" should be connected to a standard RS232 port on a computer or modem. The baud rate, number of stop bits, number of data bits, and parity may be selected via jumpers located under the top panel. Appendix 2 describes how to access these jumpers and appendix 3 describes the jumper settings.

The rear panel connector labeled "TERMINAL" may be connected to a terminal, but this connection is not crucial to the operation of the VectorScan. It is provided so that the VectorScan may be inserted "between" the computer and a terminal, thus allowing the VectorScan to be added to systems without an extra RS232 port. This connection is also useful when using the VectorScan at a remote location via a modem. The baud rate, stop bits and parity for this port are the same as for the "COMPUTER" port described above.

The nine pin "D" connector provides the RGB/sync signals for driving an RGB color monitor. The signals provided are Red, Green, Blue, Intensity, Horizontal sync, Vertical sync, and Ground. All signals are positive TTL levels. Appendix 7 lists the pinout of this connector.

The BNC video output jack on the rear panel provides a 2.0 volt RS170 composite video signal for operating a monochrome monitor. This should be connected to a standard RS170 monochrome monitor. While the VectorScan will accomodate any standard RS170 monitor, it is recommended that a high persistence phosphor (eg. P 39) be used to reduce the effects of interlace flicker.

Finally, the port labeled "PRINTER" should be connected to the parallel interface of the printer to be used. The printers which are supported (in the graphics mode) by the VectorScan are listed in appendix 4. If your printer is not listed in this appendix then its graphics structure should be checked to see if it matches any of the printers listed. Consult Applied Data Sytems Inc. if the printer you have is not supported by the standard VectorScan 512. The printers listed in appendix 4 are supported in the graphics hardcopy mode, however the VectorScan supports the spooler mode for all printers conforming to the Centronics interface standard.

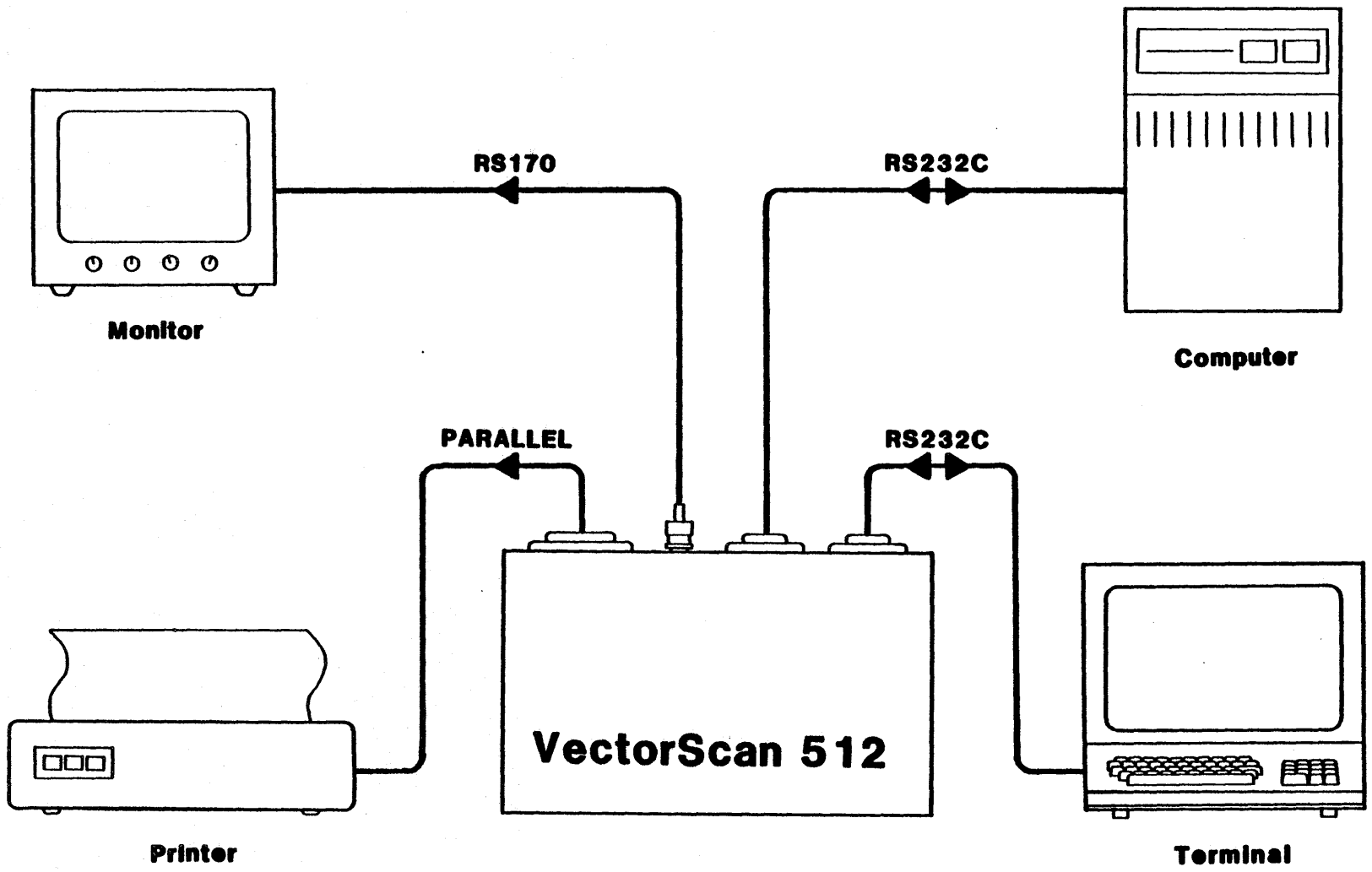

RS170
RS232C
Monitor
Computer
PARALLEL
RS232C
VectorScan 512
Printer
Terminal

Figure 3-1

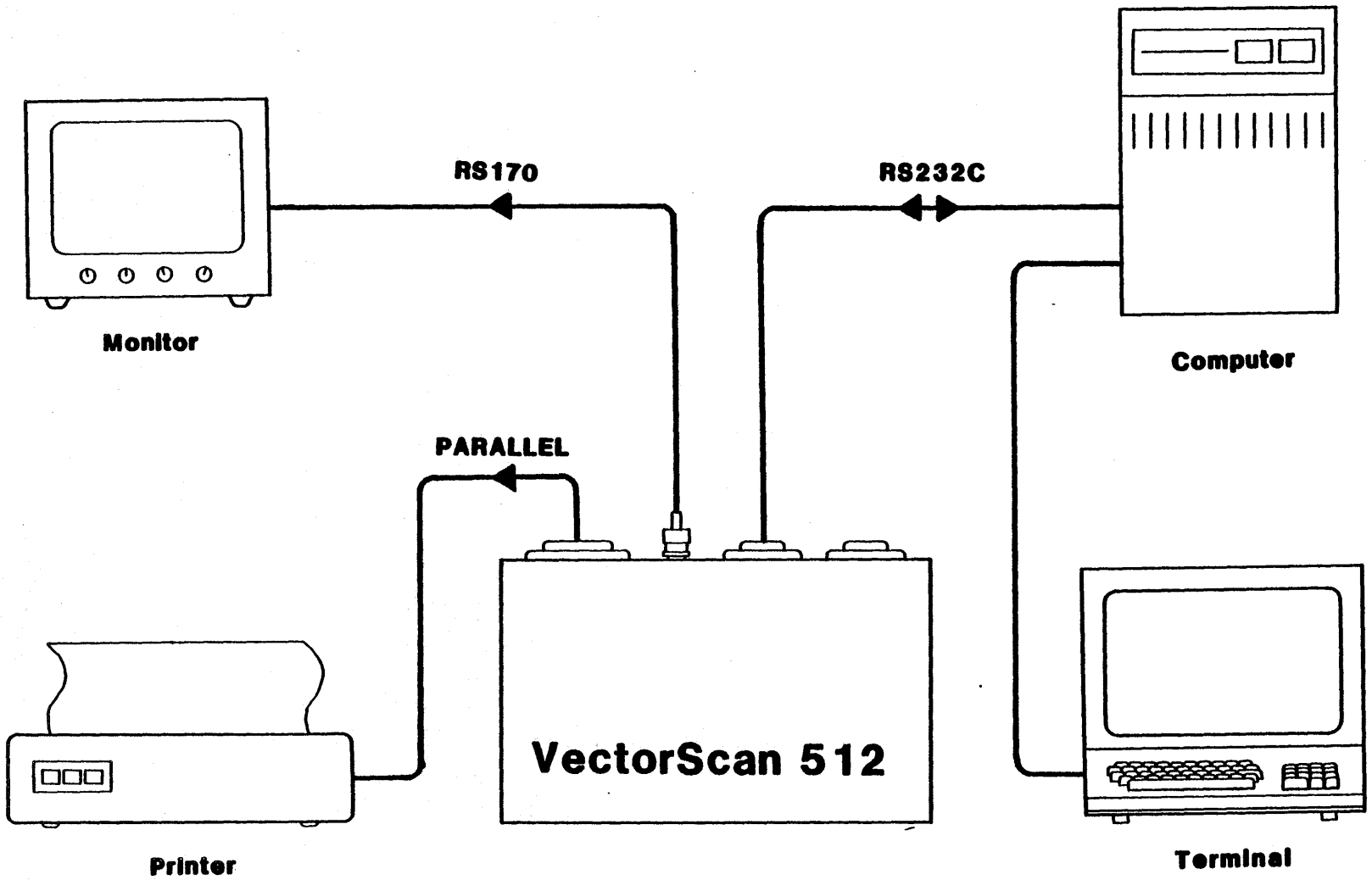

RS170
RS232C
Monitor
Computer
PARALLEL
VectorScan 512
Printer
Terminal

Figure 3-2

SECTION 4 - OPERATION

4.1 SET UP

The VectorScan can be connected in either of the two configurations shown in figures 3-1 and 3-2. The configuration depicted in figure 3-1 is for using the VectorScan "between" a terminal and the computer. In that configuration the VectorScan normally passes all data transparently between the computer and terminal. Special control commands can be issued to activate the VectorScan and/or to inhibit data from being passed to the terminal. Hence, in this configuration, the VectorScan may be added to a system without the need for an extra serial port or it may be used to provide graphics at a remote location. In the configuration shown in figure 3-2 the VectorScan is connected to a dedicated port and no terminal need be connected. A terminal can, however, be added in this configuration for debug purposes. (Note that the VectorScan does not respond to input from the terminal port.) If the VectorScan is connected to the printer port of the computer then, when it is not being used for graphics, it can be used in the spooler mode to speed listings. The special control sequences for activating the VectorScan and inhibiting the terminal are applicable in either configuration. It is important to note that the terminal data input is ORed with the VectorScan serial output to the computer. Hence care should be taken not to use the terminal while the VectorScan is actively producing serial output data. The VectorScan uses two protocols for throttling the computer commands. The XON/XOFF (control Q control S) protocol is available for software handshaking and the DTR (Data Terminal Ready) line can be used to provide a hardware handshake. The DTR handshaking is always enabled. The XON/XOFF software handshaking may be enabled or disabled with VectorScan commands.

4.2 OPERATIONAL MODES

The VectorScan operates in one of several modes depending on the commands it receives. These are the transparent, graphics, hardcopy, and spooler modes. In each of these modes the VectorScan may be operated in half or full duplex and the terminal port may be enabled or disabled.

The VectorScan defaults to the transparent mode at power up or can be placed in the transparent mode by an "@D" command sequence. In the transparent mode, the VectorScan examines the data being received via the "COMPUTER" connector and searches for control command sequences ignoring everything else. The VectorScan will remain in the transparent mode until it receives an "@C" command sequence.

When the VectorScan receives an "@C" command sequence it enters the graphics mode. In this mode the VectorScan interprets the

data stream being received via the "COMPUTER" port as graphics commands. This data is also sent to the terminal port unless that port has been inhibited. If the VectorScan is in full duplex while in the graphics mode then it will echo all characters it receives and return error indications for invalid commands. The VectorScan has a 128 character input FIFO to buffer incoming data. If that FIFO becomes full, the VectorScan will throttle the host computer by either forcing low the DTR line on the serial interface (Pin 20 "COMPUTER" connector) or by sending an XOFF (control S), depending on the configuration selected. As the FIFO empties, the VectorScan will again allow serial input by either sending an XON (control Q) or asserting the DTR line. From the graphics mode the VectorScan can be placed in the transparent mode by receiving the command sequence "@D". Any commands in the input FIFO when the "@D" is received will be executed. The VectorScan does not place the "@" commands in the input FIFO but, instead, interprets them immediatley.

The hardcopy mode is used to copy the image in screen memory to the attached graphics printer. The VectorScan can be placed in the hardcopy mode only from the graphics mode and will return to the graphics mode when the hardcopy is finished. While in the hardcopy mode, the VectorScan's command FIFO can be filled with graphics commands.

The printer spooler mode of the VectorScan is used to allow a listing to be queued in screen memory. This mode can only be entered from the graphics mode. While in the spooler mode, the VectorScan uses screen memory as a buffer thus destroying any image that may have been there.

4.3 CONTROL COMMAND SEQUENCES

The VectorScan accepts commands received over the "COMPUTER" port. These commands fall into two main catagories - operating commands, described in subsequent sections, and control commands which are described here. The format of the control commands is an "@" followed immediatly by the command character. This two character command must be the first sequence of characters on a line in order for it to be interpreted as a control command. (Note that a new line is defined by a carriage return followed by any number of line feeds or nulls. This control command syntax was chosen because it is unlikely to appear in normal text and because some systems will trap nonprinting control codes.) A list of the control command sequences accepted by the VectorScan and their meanings is contained in appendix 6. All the control sequences are passed to the attached terminal unless it has been inhibited. Control sequences are never echoed to the host computer.

The VectorScan may be enabled (graphics mode) or disabled (transparent mode) by the control commands "@C" and "@D" respectively. The commands "@E" and "@F" are used to enable or

disable VectorScan output to the computer when the VectorScan is in the graphics mode. When the VectorScan receives an "@F" it ceases to put data for the computer in its output FIFO, however the current contents of the output FIFO are sent. When the VectorScan output is enabled, via an "@E", the VectorScan will begin putting data in the output buffer to send to the computer. Any commands that have already been received but not yet processed (ie. in the input buffer) will produce output. Note that the XON/XOFF input control will be used by the VectorScan whether or not the VectorScan's output is enabled (if this handshaking protocol has been selected).

The output data to the terminal can be enabled or inhibited by using the "@A" and "@B" commands respectively. The VectorScan defaults to a terminal enabled ("@A") condition. Note that the terminal input through the VectorScan to the computer is always enabled.

Finally, the "@R" command has been provided to reset the VectorScan regardless of its current state. This command forces the VectorScan to execute its power up procedures which initializes all variables and modes to their defualts. The current command is aborted immediatly (including hardcopy and printer spooling operations) and the unit is placed in its transparent mode. The screen is cleared and the sign on message is displayed. Note that the downloadable user shape table is not affected.

4.4 GRAPHICS COMMANDS

This section describes the graphics commands offered by the VectorScan 512.

4.4.1 INTRODUCTION TO GRAPHICS COMMANDS

The VectorScan 512 provides a mechanism for creating raster graphics on a 512x480 four, eight, or 16 color/gray scale display. Both RGB/sync, color and composite video/sync, monochrome signals are provided. The colors are selectable from a palette of 16 color/intensity combinations via a mapping register. The 16 gray scales are also selectable using the mapping register. The VectorScan may be equipped to provide four, eight, or sixteen simultaneous colors or gray scales. Throughout the rest of this manual the terms "gray scale", "color", and "level" will be used interchangeably. The commands described in subsequent sections, control the setting of pixels or groups of pixels on the display. The image memory is logically presented as an X-Y coordinate system with dimension 512x512. The horizontal dimension is X and the vertical dimension is Y. The notation for specifying a pixel will always be to specify the X coordinate first followed by the Y

coordinate. The pixel in the lower left corner of screen memory is 0,0 and the pixel located in the upper right corner of screen memory is 511,511. Note that screen memory is 512x512 but that only 512x480 pixels are displayed. A command is provided which specifies which Y coordinate will be on the lowest line of the display. Note that the X,Y coordinate system is referenced to the video memory not the screen display. Hence, if the Y coordinate of the lowest screen line is not 0, then the point 0,0 will not be in the lower left corner of the screen. Care has been taken in the design of the VectorScan to provide a one-to-one aspect ratio on the screen so that no special scaling will be required of the host computer.

If any command is issued which would cause the VectorScan to compute a pixel coordinate outside of the 0 to 511 range available, the VectorScan will plot that pixel modulo 512. This results in a "wrap around" effect. An example of a command that would cause wrap around is drawing a circle with a radius of 1000.

The VectorScan is capable of displaying a pixel in one of sixteen colors/gray scales. The gray scales are numbered in order of increasing intensity, with zero the darkest and fifteen the most intense. Most of the plotting and drawing commands are uni-color. (That is they use only one color when executed). The plot color may be set or changed at any time and remains in effect until a new plot color is selected.

With the exception of the control command sequences (ie the "@" commands), all VectorScan commands have the following format. Each command must start on a new line and may be preceded by any number of spaces. The command token is from one to three upper case letters and no imbedded spaces are allowed. The command token terminates when either a character other than an upper case letter or the third upper case letter is received. The command token is then immediatley evaluated to determine if the command is valid. If the command is not valid then the VectorScan will indicate this by responding with a "?" and a bell. (Of course if the VectorScan's output has been disabled by an "@F" command sequence, the VectorScan will not return the error message.) If the command token is valid then the VectorScan will parse the rest of the line for parameters. If an invalid parameter is received or if the wrong number of parameters is received the VectorScan will issue the error response just described. Any number of spaces (or tabs) between tokens (ie. command tokens and parameters) is allowed. Numeric parameters are specified in decimal and do not require leading zeros. Numbers must be specified as integers and must not include a sign or decimal point. Commands that require the address (x,y coordinate) of a point may use a "+" to indicate that the coordinate of the graphics cursor. (The exceptions to this are the binary mode commands. These commands are used to reduce the number of bytes required to specify a command.)

In the descriptions which follow examples of each command will be provided. The command tokens will be specified exactly as

they should be sent to the VectorScan (ie. as a sequence of upper case letters). The parameters, if any are required, will be shown as lower case letters. The actual parameters sent to the VectorScan should be numbers (or the appropriate ASCII character in the case of binary mode commands)

4.4.2 SET PLOT LEVEL (LEV)

As described above, all uni-color commands (ie. commands requiring only one plot color) will use the level last selected. The color (level) selected using this command determines the value of pixels to be plotted in video memory and is independent of the map register setting. The sixteen levels are numbered from 0 to 15. Another level (16) may be specified which will put the VectorScan in a special highlight mode. In this mode, existing pixel levels are increased by 8 (modulo 16) instead of being written over. This feature allows a drawn shape (eg. circle or line) to be erased by simply re-drawing that same shape. (Any level plus 16 is equal to the original level.) Drawing a shape in this special manner allows that shape to be visible in any background. (Note that the filling commands do not use this highlighting feature. The fill commands will use the last absolute level specified. See section 4.4.8 for more information of area filling.)

The "LEV" command will select the plot level to be used for subsequent uni-color commands. The format of the "LEV" command is

```
LEV z
```

where z is a number between zero and sixteen, with sixteen used to activate the highlight mode. Note that whenever the LEV command is used any previous AXM commands will over-written. The new AND mask will be zero, the XOR mask will be set to z. (See section 4.4.36 for AXM command description.)

4.4.3 PLOT POINT (P)

The "P" command is used to plot a point in the 512x512 image memory. The point will be plotted using the plot level set with the last "LEV" command or the mask set specified with an AXM command. The format of the plot command is

```
P x y
```

where x is the x coordinate of the point and
y is the y coordinate

Recall that the lower left corner of image memory is 0,0 and the upper right corner is 511,511. Also note that the plot

command plots points relative to 0,0 in image memory (which may not be the lower left corner of the screen display depending on the currently selected bottom screen coordinate).

4.4.4 DRAW VECTOR (VEC)

The "VEC" command is used to draw a vector (a line), in the current plot level, between two specified endpoints. The format of the vector command is

```
VEC x1 y1 x2 y2
```

where x1 is the x coordinate of the first endpoint
y1 is the y coordinate of the first endpoint
x2 is the x coordinate of the second endpoint and
y2 is the y coordinate of the second endpoint

4.4.5 DRAW ARC (ARC)

The "ARC" command is used to draw an arc in the current plot level. The format of the "ARC" command is

```
ARC x y r p1 p2
```

where

x is the x coordinate of the center of the arc
y is the y coordinate of the center of the arc
r is the radius of the circle containing the arc
p1 is the start angle of the arc and
p2 is the end angle of the arc

The start and end angles (p1 and p2) may be specified in one degree increments from 0 to 360 degrees. Note that the order of p1 and p2 is important when specifying an arc. If p1 (the start angle) is 45 and p2 (the end angle) is 90 then a 45 degree arc will be drawn. However if p1 is 90 and p2 is 45 then a 315 degree arc will be drawn. The arc is drawn counter clockwise from the start angle to the end angle.

4.4.6 DRAW CIRCLE (CIR)

The "CIR" command is used to draw a circle in the current plot level. The circle command is identical to an "ARC" command with a start angle of 0 and an end angle 360 degrees. The format of the circle command is

```
CIR x y r
```

where x is the x coordinate of the circle's center
y is the y coordinate of the circle's center and
r is the circle's radius

4.4.7 FILL SCREEN (FIL)

The "FIL" command fills the entire display memory with the current plot level. The command may be used to clear the screen memory or set it to any given level. The format of the fill command is

FIL

Note that the fill command accepts no parameters. Also note that the special highlight mode does not apply to this command. The "FIL" command will use the last absolute plot level specified.

4.4.8 FILL AREA (PFL)

The "PFL" command is used to fill an area of display memory. The area is defined by a boundary in display memory. The boundary and the fill level are both specified by the current plot level or AND/XOR mask.

PFL x y (amask)

where x is the x coordinate of a point inside the boundary and
y is the y coordinate of the point inside the boundary
(amask) is an optional AND mask for selecting which bits in a pixel are to be tested as a boundary condition.

The area fill command will fill very complex regions including regions with holes and barriers. To understand what will be filled when the command is invoked it is helpful to imagine filling that area with water. That is, if water were poured into the area at the point specified in the fill command, the region filled by the water is the same region that the "PFL" command would fill. The fill level will "leak out" of small holes in the boundary and "flow around" obstacles within the area. A very intelligent and efficient filling algorithm has been employed in the VectorScan to allow filling of extremely complex shapes, however there is a limit to the complexity of the shape which can be filled. This is not a limit in the class of shapes which can be filled (eg. simple polygon, star shaped, anulus etc.), rather it is limit to the number of irregularities that may be in the border (eg. the number of cusps, obstacles etc.). If the point specified in the fill command lies outside a boundary then all points outside the boundary will be filled.

The third optional parameter amask can be used to allow more than one pixel value to be detected as a boundary condition. If

amask is not specified it will default to 15. (All bits are checked). Whenever a particular bit plane is not being checked for a boundary condition, then that bit plane can be either forced to zero, forced to one, complemented, or not affected. Selection of the above operation for each bit plane is done with bit settings in the AND and XOR masks. (See section 4.4.36 for AXM command description.) The bits in each pixel corresponding to bits in amask that are set to one, are compared to the corresponding bits in the XOR mask to test for the boundary. For pixels inside the boundary, those bits are set to the corresponding XOR mask bit values.

AND mask bit	XOR mask bit	PFL bit operation
0	0	No operation
0	1	Complement
1	0	Force to Zero
1	1	Force to One

Note that the above table applies only to bits not being checked for a boundary condition. The bits being checked are set to the target values.

4.4.9 CHANGE LEVELS (CHG)

The "CHG" command is used to change the level of all pixels in display memory. For each of the sixteen levels a new target level is specified. The target levels need not be unique but if more than one source level is mapped into the same target level then the distinction between the pixels of those source levels will be lost. For example if both levels one and two are changed to level three then there will be no way to determine which of the new level three pixels were previously level one and which were level two. The format of the change command is

CHG I0 I1 I2 I3 I4 I5 I6 I7 I8 I9 I10 I11 I12 I13 I14 I15

where I0 specifies the level to replace the level 0 pixels
I1 specifies the level to replace the level 1 pixels
I2 specifies the level to replace the level 2 pixels
I3 specifies the level to replace the level 3 pixels.
I4 specifies the level to replace the level 4 pixels
I5 specifies the level to replace the level 5 pixels
I6 specifies the level to replace the level 6 pixels
I7 specifies the level to replace the level 7 pixels
I8 specifies the level to replace the level 8 pixels
I9 specifies the level to replace the level 9 pixels
I10 specifies the level to replace the level 10 pixels
I11 specifies the level to replace the level 11 pixels
I12 specifies the level to replace the level 12 pixels
I13 specifies the level to replace the level 13 pixels
I14 specifies the level to replace the level 14 pixels
I15 specifies the level to replace the level 15 pixels

Note that a positional notation is used so that all parameters

must be specified even if some of the levels will not be changed. The "CHG" command actualy changes the values of the pixels in display memory. This is different than the load map register (LMR) command which affects the mapping of pixel levels in video memory to levels in the displayed image.

4.4.10 GRAPHICS CURSOR ON (GCN)

The "GCN" command is used to display the graphics cursor. The graphics cursor is a 5 pixel by 5 pixel blinking "crosshair" constructed as follows. The center of the cross hair is the pixel location specified by the graphics cursor positioning command ("GCP"). Eight is added to the value of all pixels comprising the crosshair. These pixels are the two pixels above the center, the two pixels below the center, the two pixels to the right of the center, and the two pixels to the left of the center pixel. (Note that the "wrap around" feature also applies to the cursor.) Because eight is added (modulo sixteen) to each pixel in the crosshair, the cursor can be seen in any background. The cursor is non-destructive. When the cursor is turned off or moved the pixels forming the crosshair are restored. Also, when any command is issued that plots pixels, the cursor is turned off while the command is executed and then turned on again. This ensures that displaying the cursor will not interfere with any of the other VectorScan functions.

4.4.11 GRAPHICS CURSOR OFF (GCF)

This command disables the graphics cursor which is described in 4.4.10.

```
GCF
```

4.4.12 GRAPHICS CURSOR POSITION (GCP)

This command is used to position the graphics cursor in display memory. The graphics cursor position will be up-dated whether or not the cursor is currently enabled. (See section 4.4.10 for a description of the graphics cursor.)

```
GCP x y
```

4.4.13 ENTER TEXT (TXT)

This command uses an internal character set to draw ASCII characters in video memory. Both the size and placement of these characters can be controlled using the SCALE (SCL) and CURSOR (CRS) commands respectively.

The format of this command is as follows:

TXT d<STRING>d

Once the TXT command is detected, the VectorScan uses the first character, other than a space (20 Hex), tab (09 Hex), or Null (00 Hex), as a delimiter. In the above example the delimiting character is "d". As each character in the string is drawn, the cursor position is adjusted so that subsequent characters are properly positioned. When the delimiting character is detected the TXT command terminates and other VectorScan commands can commence. As an example,

TXT $This is a test.$

will display the message "This is a test." on the video monitor. Note that the ASCII character "$" is used as the delimiting character.

4.4.14 SCALE CHARACTERS (SCL)

This command is used to vary the size of characters that are drawn with TXT or DUC (user character) commands. Characters can be scaled in both the X and Y dimensions independently. Characters that are to large for the current cursor position will be wrapped around on both horizontal and vertical axis.

The built in character set used with the TXT command provides a 5 X 7 dot character. Replication of this basic dot pattern is used to accomplish the required amount of character scaling. When a downloaded character is drawn, the end points of the stored vectors are multiplied by the X and Y scale factors. (See appendix 5.)

4.4.15 CHARACTER CURSOR POSITION (CRS)

This command is used to set the starting location for the drawing of ASCII characters. The x-y coordinates define the upper left hand corner of the character. The character cursor is not visible. Each time a character is drawn, the character cursor is updated to the next character position which depends on the scale currently in effect. When a carriage return (ASCII code 13 decimal or 0D hexadecimal) is received in the text mode, the x coordinate of the cursor is set to 0. A line feed (ASCII code 10 decimal or 0A hexadecimal) received in the text mode causes the cursor to be moved down 1 character line which depends on the scale currently in effect. The format of this command is

CRS x y

4.4.16 PRINTER SPOOLER MODE (PSM)

The PSM command is used to place the VectorScan into a mode in which serial data is buffered in screen memory and transfered to the printer port. As the data is buffered in screen memory any existing graphic image will be erased. As in the TXT command the first character other than a SPACE, TAB or NULL is not printed. This first character is used for detecting the end of the string or file that is being sent to the printer.

PSM d<Text string or file to be sent to the printer.>d

The string can contain any ASCII characters including control or escape sequences. Care must be taken to choose a delimiting character (shown as "d" in the example) that will never appear in the file being sent to the printer. Any such occurence would cause the premature termination of the PSM command.

4.4.17 PRINT HARD COPY (PHC)

This command is used to send a copy of the video image to a graphics printer. In order to allow color mixing and over striking to generate hard copy gray levels, multiple print head scans are allowed. The number of head scans and a printer type identifier are specified as follows.

PHC type scans

Both "type" and "scans" are integer numbers that define the type of printer being used and the number of scans to be processed. The mapping of VectorScan levels into over strikes or colors is defined in the Define Scan Levels command (DSL). See appendix 4 for printer type numbers. The number of allowed scans is 1 through 4.

4.4.18 DEFINE SCAN LEVELS (DSL)

This command is used to define which pixel level or levels will cause the printer to print a dot. In this manner various pixel levels can be combined into a single hardcopy image. By specifing the same level to be printed in separate scans, that level will cause oversrikes resulting in darker pixels on the hardcopy. This mechanism can be used to generate gray scales on the printer or (in the case of the color printers) to map the levels into colors.

Since there can be up to four separate scans with different level specifications, each scan is defined along with the levels to be used for that scan.

DSL scan# I0 I1 I2 I3 ... I15

In this command "scan#" is a integer between 1 and 4 that identifies which scan is being specified. The varibles I0 through I15 are optional. Each variable is an integer between 0 and 15 and, if included, will cause pixels of that value to print during the scan specified in "scan#".

DSC 2 1 3

This command defines scan 2 to print whenever pixels of levels 1 or 3 are detected.

4.4.19 DEFINE COLOR SEQUENCE (DCS)

The DCS command in only applicable to hardcopy using a color printer. In order to mix ribbon colors the DCS command can be used to define which colors are printed on each scan. By specifing which pixel level will cause the printer to print (the DSL command), and which ribbon color is used for each scan, pixel levels can be mapped into any individual or combination of colors.

DCS c1 c2 c3 c4

In the above example c1 through c4 are digits that define which colors are to be used for scans 1 through 4 respectivly. Consult the printer manual to determine which color corresponds to each number.

DCS 2 1 1 4

The above example would cause color 2 to be printed in scan 1, color 1 would be printed on scans 2 and 3, and color 4 will print on scan 4.

4.4.20 SYNCHRONOUS MODE ON (SON)

This command will cause all further video memory accesses to be synchronized with the retrace of the video monitor. Since video data is blanked during transfers to and from video memory, use of this command will eliminate the "snow" from the video image as the VectorScan executes drawing commands.

SON

4.4.21 SYNCHRONOUS MODE OFF (SOF)

In order to allow the most rapid drawing and screen control the synchronization of video memory accesses to video retrace can be disabled with the SOF command.

SOF

4.4.22 SET SCREEN BOTTOM (SSB)

This command allows the user to specify which horizontal line is displayed at the bottom of the video screen. In this manner the image can be made to scroll vertically. The bottom line must be an even numbered Y coordinate. If an odd number is specified, then the next lowest even number will be used.

SSB y

4.4.23 INTERLACE ON (ION)

This command allows the user to control whether or not the video disply is interlaced. The interlaced mode provides 480 displayed lines, and non-interlaced allows 240 lines. In either mode there are 512 lines in video memory. When using a short persistence phosphor (P 31 or P 4) in the interlaced mode, horizontal lines will flicker.

4.4.24 INTERLACE OFF (IOF)

This command allows the user to operate the video monitor in a non-interlaced mode. This mode allows 240 displayed lines.

4.4.25 SET USER CHARACTER (SUC)

This command allows the user to specify their own set of special characters or shapes. Characters and shapes are stored as a series of bytes that are encoded to draw a set of vectors. This byte format is shown in appendix 5. The standard VectorScan has 1024 bytes of memory available for use in the shape table. Another 2048 bytes can be added to this with the addition of another 2048x8 RAM. This option can be installed by the factory.

SUC start# byte1 byte2 byte3 ... byteN

In the above sequence SUC initiates the command, start# positions the shape array (byte0 through byteN) within the shape table. The first byte of the table is numbered 0.

4.4.26 DRAW USER CHARACTER (DUC)

This command is used to draw a user shape which has been specified with a SUC command. The shape is drawn relative to a

point specified. The character scale currently in effect will be used to scale the vectors which define the shape.

DUC start# x y

DUC defines the command, start# points to a set of bytes that define the shape to be drawn, x and y define where the shape is to be drawn. See appendix 5 for user shape byte formats.

4.4.27 ENABLE XON/XOFF HANDSHAKE (EXH)

As described earlier, the VectorScan provides two methods for throttling the computer output. One method is the RS232 Data Terminal Ready (DTR) signal which is a hardware handshaking mechanism. The other method is the XON/XOFF software handshaking protocol. This protocol uses an XOFF (control S) to suspend output from the computer and uses XON (control Q) to resume output. The "EXH" command is used to enable the XON/XOFF handshaking. The XON/XOFF is enabled on power up and need not be specified unless the "DXH" command (4.4.28) has been issued. The format of this command is

EXH

4.4.28 DISABLE XON/XOFF HANDSHAKE (DXH)

The "DXH" command is used to disable the XON/XOFF software handshaking protocol (see "EXH" in section 4.4.27 above). The format of this command is

DXH

4.4.29 PRINT STRING DIRECT (PSD)

The "PSD" command is used to send data to the attached printer. It differs from the Printer Spooler Mode command (PSM) in that it does not buffer the data in image memory. This command is useful for sending short strings to the printer when there is valuable data in the display memory. The format of this command is

PSD d<string>d

The <string> to be printed may contain any characters. The first non-blank character after the "PSD" (d in this case) is used as the delimiter for the string. When the delimiter is again encountered, the command terminates and characters are no longer sent to the printer.

4.4.30 LOAD MAP REGISTER (LMR)

The "LMR" command is used to load a value into the color/gray scale mapping register. Pixel levels in video memory are mapped through (looked up in) a mapping register to produce the desired output color or gray scale on the screen display. The "LMR" command accepts the source and output levels as parameters and loads the map register accordingly. The map register is loaded during vertical retrace so that a "split screen" condition will be avoided. The format of the load map register command is

LMR s o

The parameter s specifies the source level from the video memory and the parameter o specifies the output value to be sent to the screen. Appendix 8 described how the output numbers relate to colors and gray scales.

4.4.31 BINARY PLOT POINT (BPP)

The "BPP" command allows plotting points in the current plot level. This command uses a special binary encoding of the pixel coordinates. Exactly three bytes are required to specify a point and they are defined as follows. Each byte contains six bits of the pixel coordinate. The byte is encoded as the 6 bit number + 33 (21 hex or ASCII "!"). The first byte contains the low 6 bits of the X coordinate. Byte 2 has the three high X bits and the 3 high Y bits. The 3 Y bits are in bits 0, 1 and 2, with the X bits in bits 3, 4 and 5. (Bit 0 being the least significant.) Byte 3 contains the low 6 Y coordinate bits. The "BPP" command will continue accepting (and plotting) pixel coordinates until a carriage return (13) is sent. A carriage return sent at any time during the "BPP" command will terminate the command even if it is sent before a complete coordinate has been specified (that point will not be plotted). Any tabs or spaces that occur are ignored. (Note that binary encoding format is such that only printing ASCII characters are used.)

BPP xzyxzyxzy

4.4.32 BINARY FILL SCREEN (BFS)

The "BFS" command is used to fill image memory with specified data. The "BFS" command sets pixels to the levels specified starting from the point given in the command and moving in increasing X then increasing Y. Each level is specified using one byte with the binary format described for the "BPP" command. That is, the level (between 0 and 15) is added to 33 (20 hex or ASCII "!") to form the byte. Spaces and tabs are ignored and the command terminates when a carriage return is received.

BFS x y IIIIIIIIIIIIIIIIII...

The parameters, x and y, specify where the fill is to start while the parameters I specify the pixel levels to plot.

4.4.33 PRINT HORIZONTAL (PH)

The "PH" command will cause all further TXT commands to place text on the screen in a horizontal manner. This is the default mode whenever the VectorScan is powered on.

4.4.34 PRINT VERTICAL (PV)

The "PV" command will cause all further TXT commands to place text on the screen in a vertical direction. This mode is useful in placing labels on a vertical axis. This command will also rotate shapes that that are drawn from the programable shape table.

4.4.35 DOTTED LINE (DOT)

The "DOT" command is used to draw lines and user shapes in a dotted line format. Two parameters are passed to determine the length of a dash (drawn section of line) and the space between successive dash space pairs. A dash is one or more consecutive pixels that together form a segment of a dotted line.

DOT on length

Parameter "on" determines how many pixels will be set to the current level creating a dashed section of the line. Parameter "length" determines the length of the dash and spaced sequence of the line. The length of the space sequence is "length" minus "on". The range of both "on" and "length" is 0 through 255.

For example-

DOT 2 6

would result in 2 pixels out of every 6 being drawn. The dotted line would consist of 6 dot sequences, 2 on and 4 off. The dotted line mode is terminated by setting the "on" and "length" both to zero.

4.4.36 AND XOR MODE (AXM)

The "AXM" command is used to specify a logical AND mask and a logical XOR mask that allow the user to operate the VectorScan in a bitplane mode. In this mode up to four independent images can be drawn in video memory. With use of the LMR command any single plane, or combination of planes, can be displayed in any color (including black) or combinations of colors. The format for the "AXM" command is

```
AXM  amask  xmask
```

Parameter "amask" sets the logical AND mask, "xmask" sets the logical XOR mask. The range for both "amask" and "xmask" is 0 through 15. The bitplane mode is terminated by the first LEV command. Operations performed using the AXM mode allow graphics planes to be set, cleared, complemented or left undisturbed.

The values accessing the four single bit planes are listed below to serve as examples.

```
AXM 14 1        set bit plane 0
AXM 13 2        set bit plane 1
AXM 11 3        set bit plane 2
AXM 7  8        set bit plane 3
AXM 14 0        reset bit plane 0
AXM 13 0        reset bit plane 1
AXM 11 0        reset bit plane 2
AXM 7  0        reset bit plane 3
```

4.4.37 CHANGE CONTROL CHARACTER (CCC)

This command allows the user to change the control escape character ("@") to any other character. This is useful if the host system also uses "@" as an escape or control character.

```
CCCnewchar
```

4.4.38 LOAD VIDEO CONTROLLER (LVC)

This command gives the user access to the 6845 CRT controller registers. The command is not normally used and the user should consult the data sheets and the factory before using it.

```
LVC reg# value
```

APPENDIX 1 - COMMAND SUMMARY

COMMAND		FUNCTION	SECTION
ARC	x y r a1 a2	Draw arc	4.4.5
AXM	amask xmask	AND/XOR mode	4.4.36
BFS	x y I I I I ...	Binary fill screen	4.4.32
BPP	xzy xzy xzy...	Binary point plot	4.4.31
CCC	newchar	Change Control Character	4.4.37
CHG	I0 I1 I2 I3	Change levels	4.4.9
CIR	x y r	Draw circle	4.4.6
CRS	x y	Postion text cursor	4.4.15
DCS	c1 c2 c3 c4	Define color sequence	4.4.19
DOT	on length	Dotted line	4.4.35
DSL	s# I0 I1 I2 I3	Define scan levels	4.4.18
DUC	start# x y	Draw user character	4.4.26
DXH		Disable XON/XOFF	4.4.28
EXH		Enable XON/XOFF	4.4.27
FIL		Fill screen	4.4.7
GCF		Graphics cursor off	4.4.11
GCN		Graphics cursor on	4.4.10
GCP	x y	Position graphics cursor	4.4.12
IOF		Interlace off	4.4.24
ION		Interlace on	4.4.23
LEV	l	Set plot level	4.4.2
LMR	s o	Load map register	4.4.30
LVC	reg# value	Load Video Controller	4.4.38
P	x y	Plot point	4.4.3
PH		Print Horizontal	4.4.33
PV		Print Vertical	4.4.34
PFL	x y (amask)	Polygon fill	4.4.8
PHC	type scans	Print hard copy	4.4.17
PSD	d<string>d	Print string direct	4.4.29
PSM	d<string>d	Printer spooler mode	4.4.16
SCL	x y	Scale character	4.4.14
SOF		Synchronous mode off	4.4.20
SON		Synchronous mode on	4.4.21
SSB	start	Set screen bottom	4.4.22
SUC	start# <data>	Set user character	4.4.25
TXT	d<string>d	Enter text	4.4.13
VEC	x1 y1 x2 y2	Draw vector	4.4.4

APPENDIX 2 - ACCESS TO COMMUNICATION JUMPERS

REMOVE POWER CORD

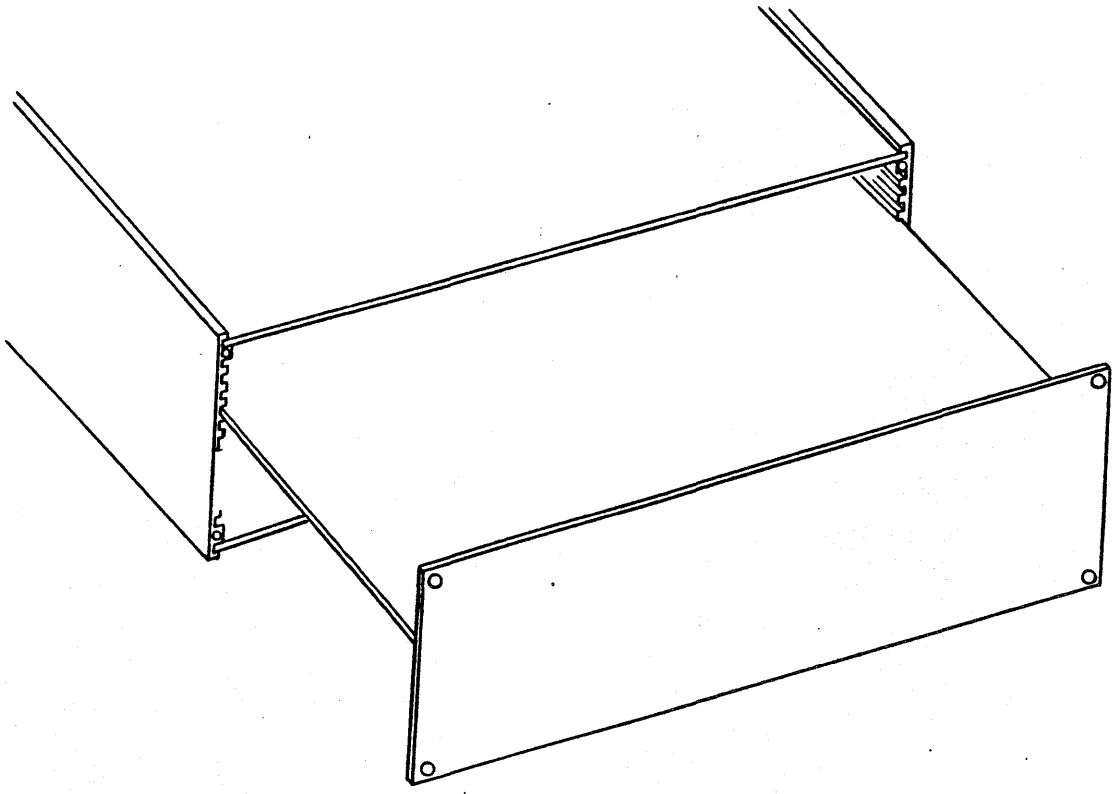
To access the communication control jumpers:

1. ********** REMOVE THE POWER CORD!!! **********
2. Remove all other cables from the VectorScan
3. Remove four corner screws.
4. Slide rear panel and circuit board away from the chassis.

To reassemble the VectorScan:

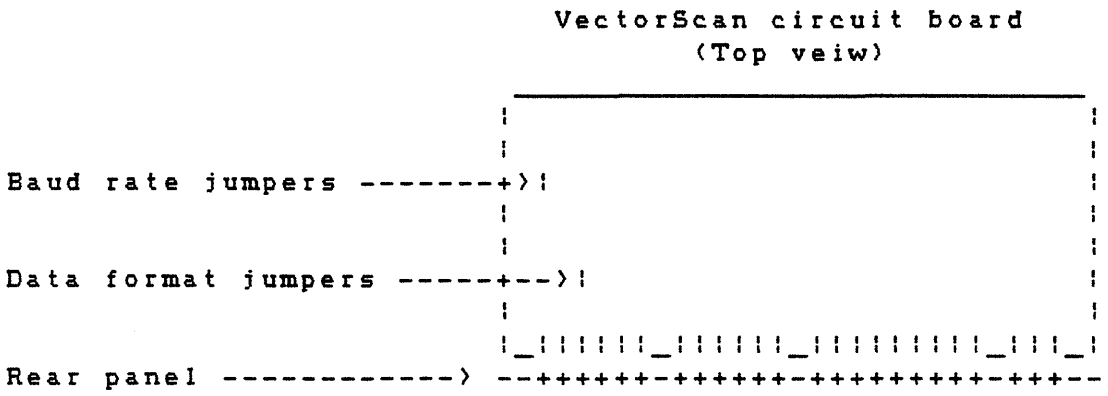
1. Locate the ninth small groove from the chassis top. (This groove is 1.7 inches from the top edge of the side panel.)
2. Slide the rear panel and circuit board assembly into the chassis on the ninth small groove.
3. The four rear panel holes should line up with the chassis holes. (If not re-count the groves.)
4. Replace the four corner screws.

FIGURE AP2-1
REMOVAL OF BACKPLANE AND BOARD



APPENDIX 3 - SERIAL CONFIGURATION JUMPERS

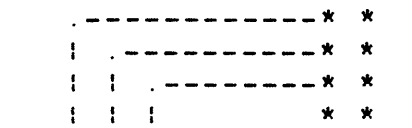
The serial channel confiuration may be selected via jumpers located on the VectorScan printed circuit card. To gain access to these, refer to the procedures outlined in appendix 2. Two sets of jumpers are provided - one for setting the baud rate and the other for determining the data format. The set of 8 jumper posts nearest the left side of the circuit board (when veiwed from the back) are for selecting the baud rate. Only one jumper should be installed here. The remaining set of 4 jumper posts, located directly behind the "TERMINAL" connector are for selecting the data format.(Only 3 of these jumpers are used, the fourth jumper is ignored.) The location and settings of the jumpers are shown on the following page.



Baud rate jumper selection.

JUMPER	BAUD RATE	
* *	19200	(Jumper close to MC6850 UART)
* *	9600	
* *	4800	
* *	2400	
* *	1200	
* *	600	
* *	300	
* *	150	(Jumper close to backpanel)

Data format jumper selection



JUMPER	IN/OUT	DATA BITS	PARITY	STOP BITS
I	I	8	ODD	1
I	O	7	ODD	1
O	I	8	EVEN	1
O	O	7	EVEN	1
I	I	8	NONE	1
I	O	7	ODD	2
O	I	8	NONE	2
O	O	7	EVEN	2

APPENDIX 4 - SUPPORTED GRAPHICS PRINTERS

Printer Type	PHC Varable
OKIDATA 83 and 84	1
OKIDATA 82	2
PRISM 80/132 COLOR	3
EPSON MX-80	4
EPSON MX-100	5
PRISM 132 EXPANDED MODE	6

If your printer is not included in the list of supported printers, then check to see if its graphics structure is the same as one of the printers in the list. If not then please contact Applied Data Systems. Support for printers not contained in the list may have been provided after the printing of this manual.

APPENDIX 5 - DOWNLOADABLE CHARACTER FORMAT

A downloadable user shape facility is provided by the VectorScan. The built in character set includes the printable ASCII characters which are formed in an 8x8 dot matrix. Since the primary application of this device is graphics (not text) the downloadable characters (shapes) are not specified in a dot matrix format. Instead, they are specified as a sequence of vectors. Each vector is specified by either one or two bytes as described later. The first endpoint of each vector in the shape is defined as the second endpoint of the last vector plotted for that shape. The second endpoint of each vector is specified by a byte (or two bytes) in the shape table and is relative to the first endpoint. The very first vector of a shape uses the position specified by the Draw User Character ("DUC") command for its first endpoint. Hence the shapes may be placed anywhere in the display memory. Figure AP5-1 illustrates the above description. A flag associated with each byte (or byte pair) in the shape table specifies whether the vector should be drawn or the endpoint just updated. This allows disconnected shapes to be specified. If the vector is drawn, the current plot level will be used including the highlight mode, if selected. The x-y character scale ("SCL") in effect at the time the shape is drawn will be used to scale the shape.

The format of the byte or byte pair which specifies a vector is shown in figures AP5-2 and AP5-3. First the single byte format will be described then the two byte format. In the single byte format, the high order bit (bit 7) is used to specify whether the vector will be drawn or not. If bit 7 is "1" (ie the value of the byte is greater than 127) then the vector will be drawn. The next bit (bit 6) is the byte extension bit and specifies whether the two byte format is being used or not. For the one byte format, bit 6 is "0". Bits 3,4, and 5 specify the x offset of the vector's second enpoint and bits 0,1, and 2 specify the y offset. Both the x and y offsets (in the single byte format) range from -4 to 3 and are represented as 3 bit 2's complement numbers.

The two byte vector specification format is used to extend the range of the endpoint offset. Bit 7 of the first byte indicates whether or not the vector should be drawn just as in the one byte format. Bit 6, the extension bit, will be a "1" in the two byte format to indicate that another byte will follow. Bits 3,4, and 5 represent the three high order bits of the x offset and bits 4,5,6 and 7 of the second byte represent the four low order bits of the x offset. Similarly, bits 0,1, and 2 represent the three high bits of the y offset while bits 0,1,2, and 3 are the four low bits of the y offset. This format is shown in figure AP5-4. The x and y offsets, using this format, range from -64 to 63 and are represented as 7 bit 2's complement numbers.

The number of vectors which specify each shape is variable. A shape terminates when the next byte in the shape table is zero. This would correspond to not moving the endpoint and not drawing the point. Single points can be specified by drawing a vector of

length zero.

The vector specification format was chosen to allow complex shapes to be defined using a minimal number of bytes in the shape table. Many intricate shapes can be defined using the one byte format and, wherever possible, this format should be used to reduce the length of the shape table. The two byte format is provided for those shapes which require relatively long vectors. The formats may be mixed when specifying a shape.

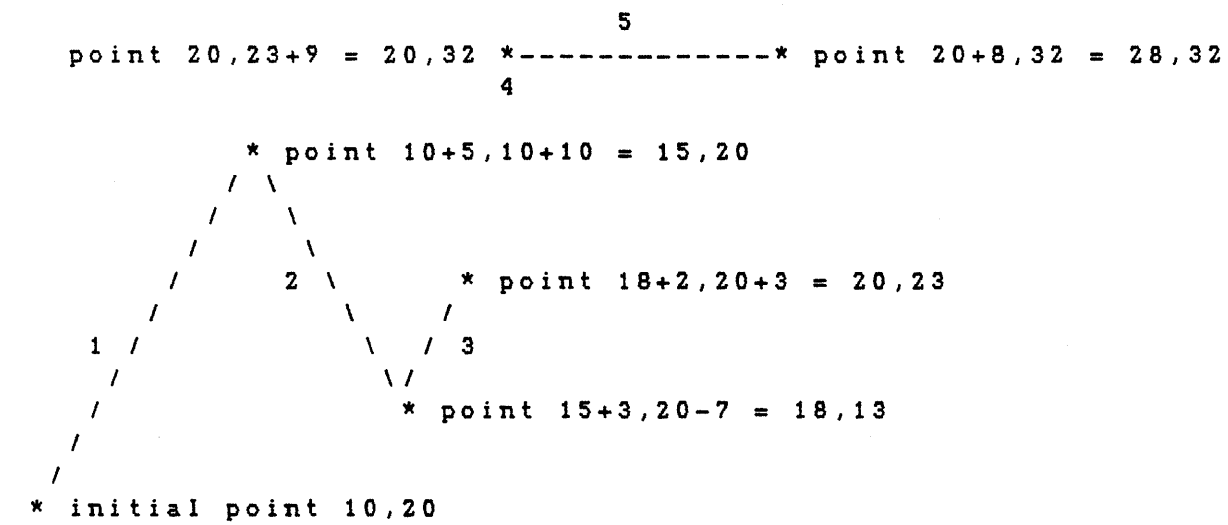


Figure AP5-1

The above shape was specicfied using the following sequence of vectors.

VECTOR	DRAW	OFFSET X	OFFSET Y	BYTE(S)	REMARKS
1	YES	5	10	192 90	Starts at point specified in 'DUC'
2	YES	3	-7	199 57	Second vector. Starts at vector 1 end
3	YES	2	3	147	Third vector. Short enough for 1 byte
4	NO	0	9	64 10	Moves point but does not draw
5	YES	9	0	192 128	Last vector to draw
	NO	0	0	0	Value of 0 terminates the shape

The actual command line to specify this shape starting at location 22 in the shape table is

```
SUC 22 192 90 199 57 147 64 10 192 128 0
```

The shape was drawn at location 10,20 with the command

```
DUC 22 10 20
```

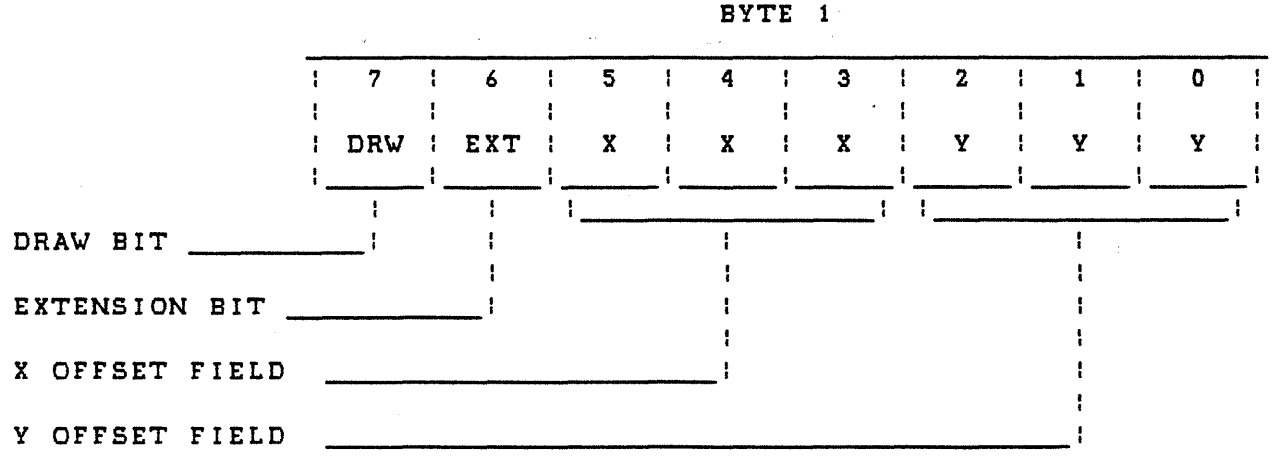


FIGURE AP5-2
USER CHARACTER BYTE 1 FORMAT

If the EXTENSION bit = 0 then only the first byte is used.

If the EXTENSION bit = 1 then a second byte is used to extend the range of the x and y offsets. The format is shown below.

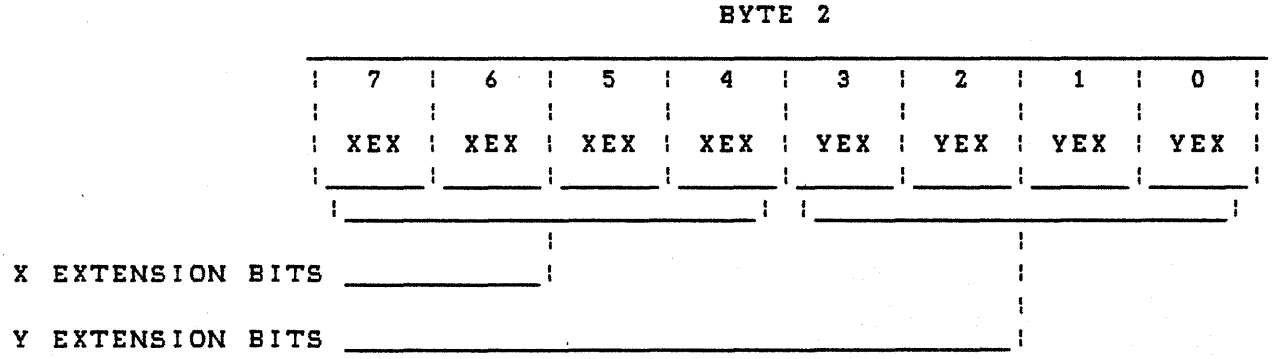


FIGURE AP5-3
USER CHARACTER BYTE 2 FORMAT

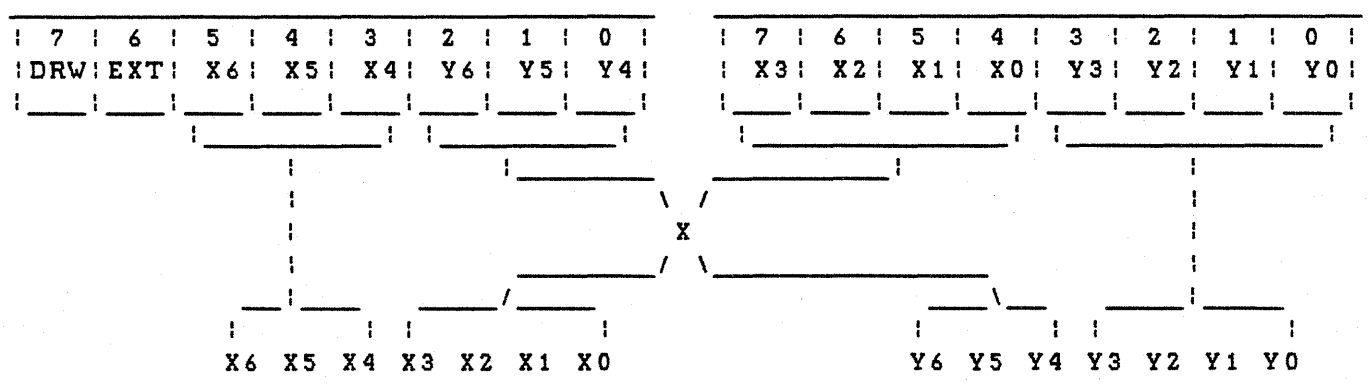


FIGURE AP5-4
FORMAT OF EXTENDED OFFSETS

APPENDIX 6 - CONTROL COMMAND SUMMARY

@A TERMINAL ENABLE

This command enables the serial data stream from the computer through the VectorScan to the terminal. This is a power on default mode. If the data stream is already enabled it will remain enabled.

@B TERMINAL DISABLE

This command will inhibit the data stream from the computer through the VectorScan to the terminal. In this manner graphic drawing command can be prevented from being displayed on the user terminal.

@C VECTORSCAN ENABLE

This command will cause the VectorScan to evalualte the incoming data as graphics commands. Data will be scanned until an @D command is recieved.

@D VECTORSCAN DISABLE

This command will inhibit graphics command execution."@" commands will continue to be decoded and processed. This is a power on default mode.

@E ECHO ENABLE

This command will echo all characters back to the computer connector. When echo is enabled, errors in graphic commands will result in question marks being sent to the computer port. This mode is very useful since it allows the VectorScan to be used with a simple RS-232 terminal. (Note that the terminal must be connected to the computer connector through a DCE-DTE swap cable.)

@F ECHO DISABLE

This command will inhibit the sending of received characters or question marks that result from graphic command errors. This is a power on default mode. When in the echo disable mode, the VectorScan will produce no output except for the XON/XOFF (control Q and S) for handshaking if this handshaking mode is selected.

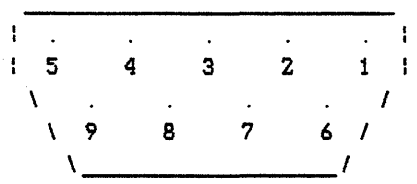
@R RESET THE VECTORSCAN

This command is used to reset the VectorScan. It will cause the unit to execute its power up procedures and is idential to a power up reset except that the downloadable user shape table is no affected. This command is useful for aborting hardcopy and long listings or for putting the VectorScan in a known state.

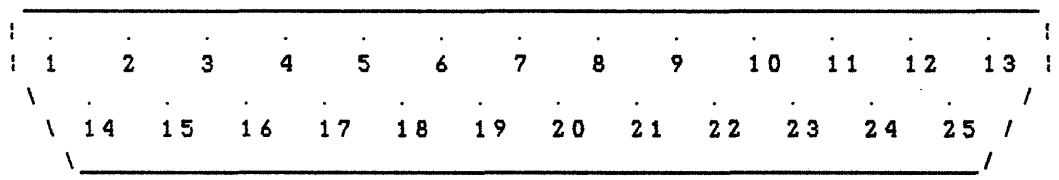
APPENDIX 7- CONNECTOR PINOUTS

This appendix lists the pinouts of the four "D" connectors on the VectorScan. The four connectors are labeled "TERMINAL", "COMPUTER", "PRINTER", and "COLOR VIDEO". The pin numbering for the three types of connectors is shown below.

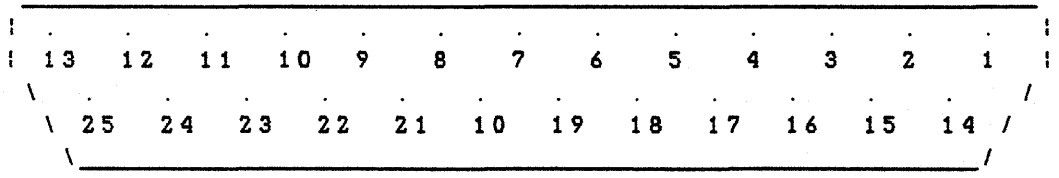
9 pin "D" connector pin numbering



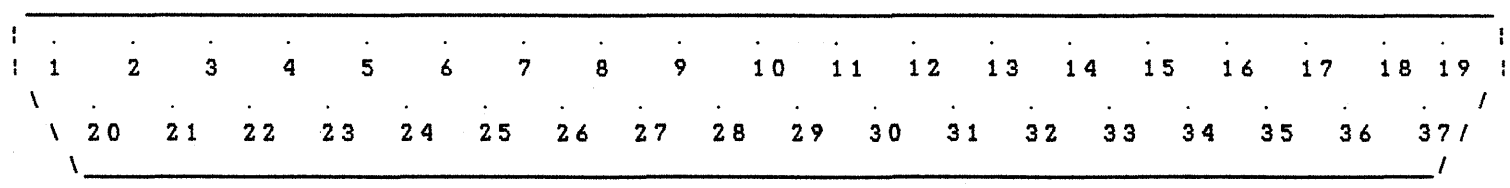
25 pin "D" COMPUTER connector pin numbering



25 pin "D" TERMINAL connector pin numbering



37 pin "D" connector pin numbering



Color video output connector pinouts

PIN	SIGNAL
1	GROUND
2	GROUND
3	RED (OUT)
4	GREEN (OUT)
5	BLUE (OUT)
6	INTENSITY (OUT)
7	NOT CONNECTED
8	HORIZONTAL SYNC (OUT)
9	VERTICAL SYNC (OUT)

Printer output connector pinouts

PIN	SIGNAL
1	/DATA STROBE (OUT)
2	DATA BIT 1 (LSB) (OUT)
3	DATA BIT 2 (OUT)
4	DATA BIT 3 (OUT)
5	DATA BIT 4 (OUT)
6	DATA BIT 5 (OUT)
7	DATA BIT 6 (OUT)
8	DATA BIT 7 (OUT)
9	DATA BIT 8 (MSB) (OUT)
10	/ACKNOWLEDGE (IN)
11	BUSY (IN)
12	NOT USED
13	NOT USED
14	SIGNAL GROUND
15	NOT USED
16	NOT USED
17	CHASSIS GROUND
18	NOT USED
19	NOT USED
20	SIGNAL GROUND
21	SIGNAL GROUND
22	SIGNAL GROUND
23	SIGNAL GROUND
24	SIGNAL GROUND
25	SIGNAL GROUND
26	SIGNAL GROUND
27	SIGNAL GROUND
28	SIGNAL GROUND
29	SIGNAL GROUND
30	SIGNAL GROUND
31	NOT USED
32	NOT USED
33	SIGNAL GROUND
34	NOT USED
35	NOT USED
36	NOT USED
37	NOT USED

Computer connector pinouts

PIN	SIGNAL
1	PROTECTIVE GROUND
2	TRANSMIT DATA (OUT)
3	RECEIVE DATA (IN)
4	CONNECTED TO TERMINAL PIN 4
5	CONNECTED TO TERMINAL PIN 5
6	CONNECTED TO TERMINAL PIN 6
7	SIGNAL GROUND
8	CONNECTED TO TERMINAL PIN 8
9	CONNECTED TO TERMINAL PIN 9
10	CONNECTED TO TERMINAL PIN 10
11	CONNECTED TO TERMINAL PIN 11
12	CONNECTED TO TERMINAL PIN 12
13	CONNECTED TO TERMINAL PIN 13
14	CONNECTED TO TERMINAL PIN 14
15	CONNECTED TO TERMINAL PIN 15
16	CONNECTED TO TERMINAL PIN 16
17	CONNECTED TO TERMINAL PIN 17
18	CONNECTED TO TERMINAL PIN 18
19	CONNECTED TO TERMINAL PIN 19
20	DATA TERMINAL READY (OUT)
21	CONNECTED TO TERMINAL PIN 21
22	CONNECTED TO TERMINAL PIN 22
23	CONNECTED TO TERMINAL PIN 23
24	CONNECTED TO TERMINAL PIN 24
25	CONNECTED TO TERMINAL PIN 25

Terminal connector pinouts

PIN	SIGNAL
1	PROTECTIVE GROUND
2	TERMINAL TRANSMIT DATA (IN)
3	TERMINAL RECEIVE DATA (OUT)
4	CONNECTED TO COMPUTER PIN 4
5	CONNECTED TO COMPUTER PIN 5
6	CONNECTED TO COMPUTER PIN 6
7	SIGNAL GROUND
8	CONNECTED TO COMPUTER PIN 8
9	CONNECTED TO COMPUTER PIN 9
10	CONNECTED TO COMPUTER PIN 10
11	CONNECTED TO COMPUTER PIN 11
12	CONNECTED TO COMPUTER PIN 12
13	CONNECTED TO COMPUTER PIN 13
14	CONNECTED TO COMPUTER PIN 14
15	CONNECTED TO COMPUTER PIN 15
16	CONNECTED TO COMPUTER PIN 16
17	CONNECTED TO COMPUTER PIN 17
18	CONNECTED TO COMPUTER PIN 18
19	CONNECTED TO COMPUTER PIN 19
20	DATA TERMINAL READY (IN)
21	CONNECTED TO COMPUTER PIN 21
22	CONNECTED TO COMPUTER PIN 22
23	CONNECTED TO COMPUTER PIN 23
24	CONNECTED TO COMPUTER PIN 24
25	CONNECTED TO COMPUTER PIN 25

APPENDIX 8- COLOR AND GRAY SCALE MAPPINGS

This appendix describes the correspondence between the available colors/gray levels and the mapping register outputs. The mapping register has a four bit output which is used to produce a palette of 16 color/intensity or gray scale combinations. The output bits have the following meaning.

OUTPUT BIT	COLOR MEANING	GRAY SCALE MEANING
3 (MSB)	RED	3 (MSB)
2	GREEN	2
1	BLUE	1
0 (LSB)	INTENSITY	0 (LSB)

Hence a map register output value of 10 would produce a low intensity magenta output by enabling the RED and BLUE outputs. The gray scale produced would be level 5 from 16 evenly spaced available levels. (Gray scale value 0 is the least intense.)

Level / Color Relationships

Level	=	Color	Level	=	Color
0	=	Black	8	=	Red
1	=	Grey	9	=	Bright Red
2	=	Blue	10	=	Magenta
3	=	Bright Blue	11	=	Bright Magenta
4	=	Green	12	=	Yellow
5	=	Bright Green	13	=	Bright Yellow
6	=	Cyan	14	=	White
7	=	Bright Cyan	15	=	Bright White

Release 1.4 of the VectorScan 512 firmware included a binary command to help improve the RS232 link utilization. In this mode, commands are invoked using a single character and numeric parameters are encoded in a one, two or three character format. Each character represents six bits by adding the six bit number to 33 (21 hex or ASCII !"). This keeps the character within the printable ASCII set. Each command is assigned a number as shown in the table. A fixed number of characters is expected for each parameter as follows.

1. X,Y coordinates - Encoded in three six bit words.

   WORD 1 - X5, X4, X3, X2, X1, X0
   WORD 2 - X8, X7, X6, Y8, Y7, Y6
   WORD 3 - Y5, Y4, Y3, Y2, Y1, Y0

2. Single character parameters - These are parameters with values less than 64. Levels, map registers, and AND masks are examples of single character parameters.

3. Two character parameters - These are parameters which can be greater than 63 such as angles, user characters, and scale factors. These are encoded as follows

   WORD 1 - V11, V10, V9, V8, V7, V6
   WORD 2 - V5, V4, V3, V2, V1, V0

As an example of how the binary format works, consider drawing a circle in the normal (ASCII) mode and in the binary mode. In the ASCII mode a circle of radius 53 centered at (100,200) is

```
CIR 100 200 53
```

In the binary mode the same command would be

```
#E,)!V
```

The # represents an encoded 2 which is the command number for CIR from the table. The E,5 and !V represent the circle center (100,200) and radius, respectively, as follows.

```
X = 100 = 001 100100 binary
Y = 200 = 011 001000 binary
WORD 1 = low 6 bit of X = 100100 binary = 36
WORD 2 = high 3 bits of X and Y = 001011 binary = 11
WORD 3 = low 6 bits of Y = 001000 binary = 8
Character 1 = 36 + 33 = 69 = E ASCII
Character 2 = 11 + 33 = 44 = , ASCII
Character 3 = 8 + 33 = 41 = ) ASCII

Radius = 53 = 000000110101 binary
WORD 1 = high 6 bits of radius = 000000 = 0
WORD 2 = low 6 bits of radius = 110101 = 53
Character 1 = 0 + 33 = 33 = ! ASCII
Character 2 = 53 + 33 = 86 = V ASCII
```

To enter the binary mode, use the BON (binary on) command. To exit the binary mode use the BOF (binary off) command. Note that this command is used from the binary mode and is encoded as I.

In addition to the binary mode offerred in release 1.4 of the VectorScan 512 firmware, certain enhancements were made. Most notable among these are the following.

Many of the commands can now function in a continuation mode. For example, the vector command (VEC) will continue to accept points after the first pair of coordinates using the last coordinate of the previous vector as the start point for the next vector. In the binary command table, those parameters enclosed in [ ] indicate the continuation parameters for each command.

A pair of enhancements were made to the FIL command which is used to erase the screen. The fill now uses the AND/XOR mask if the VectorScan is in that mode. This allows individual bit planes to be cleared. Further, optional boundaries may be specified for the fill so that a window of the screen may be erased. This boundary is specified by giving the coordinates of the diagonal corners of a rectangle. If no coordinates are specified, the entire screen is filled.

Note that the above enhancments are upward compatible with the firmware releases before 1.4.

The table that follows shows the command encoding for the binary mode. Those commands which require numeric parameters have the parameter types expected shown. Some of the commands have a continuation mode which will re-invoke the command at some point. The [ ] notation indicates at which point the continued parameters start. Optional parameters are shown in ( ).

xy => (x,y) coordinate. Three byte format
short => less than 64. One byte format
long => may be greater than 64. Two byte format

TABLE OF BINARY MODE COMMAND TRANSLATIONS

COMMAND	NUMBER	CHARACTER	PARAMETERS
ARC	10	+	xy long long long
AXM	37	F	long long
BFS	34	C	xy [short]
BOF	40	I	
BON	39	H	
BPP	33	B	[xy]
CCC	30	?	
CHG	9	*	short short short ... (16 times)
CIR	2	#	xy long
CRS	8	)	[xy]
DCS	21	6	short (short...)
DLY	38	G	long
DOT	36	E	short short
DSL	22	7	short short (short...)
DUC	19	4	long xy
DXH	28	=	
EXH	27	<	
FIL	1	"	(xy xy)
GCF	13	.	
GCN	12	-	
GCP	14	/	[xy]
IOF	23	8	
ION	24	9	
LEV	5	&	short
LMR	32	A	[short short]
LVC	35	D	[short long]
P	4	%	[xy]
PFL	3	$	xy (short)
PH	25	:	
PHC	20	5	short short
PSD	29	>	
PSM	11	,	
PV	26	;	
SCL	7	(	long long
SOF	16	1	
SON	15	0	
SSB	17	2	long
SUC	18	3	long [long]
TXT	6	'	
VEC	0	!	xy [xy]