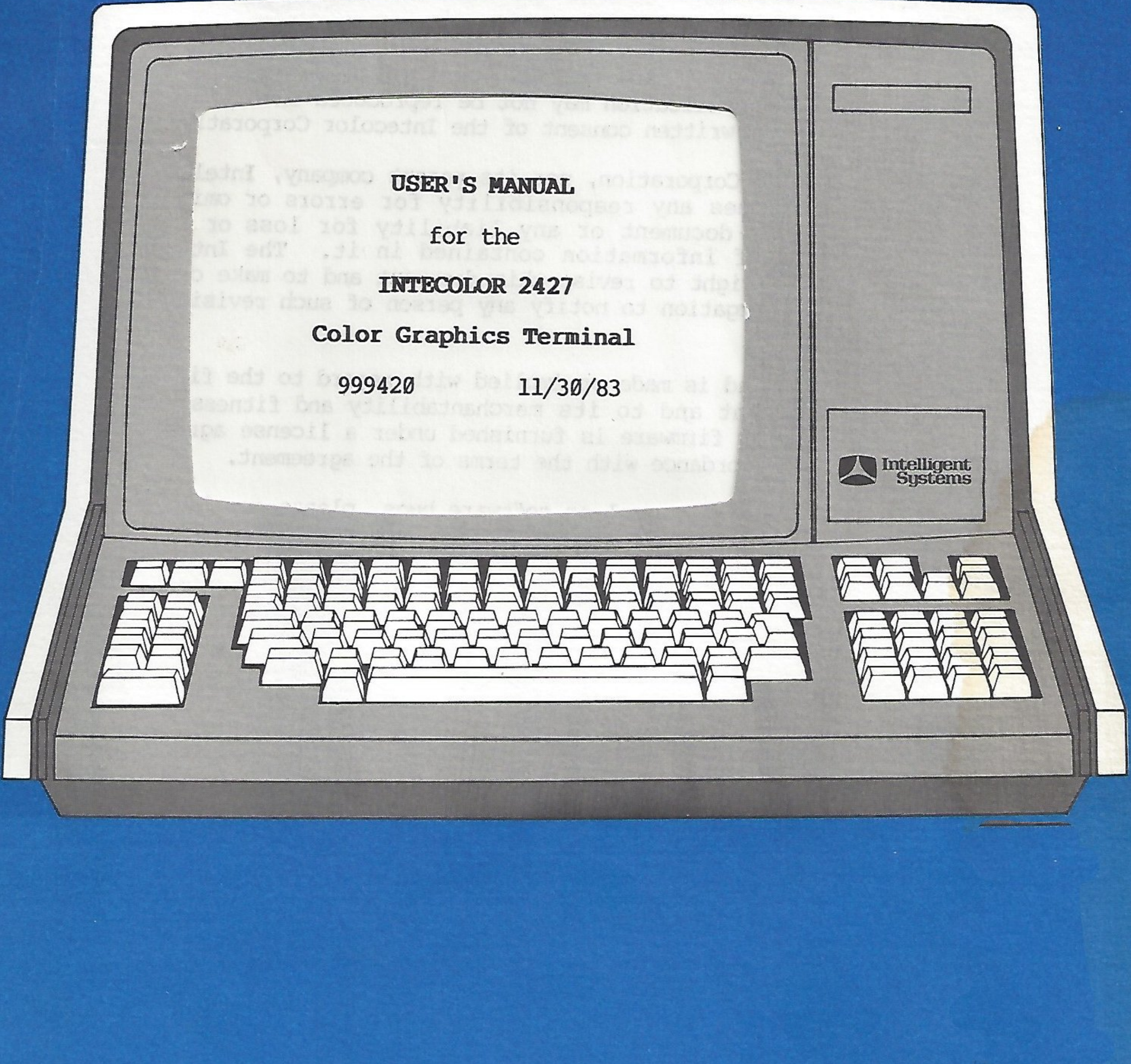


USER'S MANUAL

for the

INTECOLOR 2427

Color Graphics Terminal

999420 11/30/83

Intecolor®
AN INTELLIGENT SYSTEMS COMPANY

225 Technology Park/Atlanta • Norcross, GA 300921 • Telephone 404/449-5961 • TWX 810-766-1581

USER'S MANUAL

for the

INTECOLOR 2427

Color Graphics Terminal

999420 11/30/83

WARNING

This equipment generates, uses, and can radiate radio frequency energy and may cause interference to radio communications if not installed and used in accordance with the instructions in this manual. The equipment's radio frequency emissions have been measured and found to be within the limits established in FCC Rules, Part 15, Subpart J, for Class A computing devices. These rules are designed to limit interference by such devices to levels considered reasonable for commercial environments. If the user operates this equipment in a residential environment he will be likely to cause interference for which he can be required to take corrective action at his own expense.

CCR

Intecolor Corporation
225 Technology Park/Atlanta
Norcross, Georgia 30092
Telephone: 404/449-5961
TWX: 810/766-1581

NOTICE

The contents of this publication may not be reproduced in any form by any means without the prior written consent of the Intecolor Corporation.

Neither the Intecolor Corporation, nor its parent company, Intelligent Systems Corporation assumes any responsibility for errors or omissions which may appear in this document or any liability for loss or damage resulting from the use of information contained in it. The Intecolor Corporation reserves the right to revise this document and to make changes in its content without obligation to notify any person of such revisions or changes.

No warranty of any kind is made or implied with regard to the firmware described in this document and to its merchantability and fitness for a particular purpose. This firmware is furnished under a license agreement and may be used only in accordance with the terms of the agreement.

To report errors in the manual or software bugs, please complete and return the Software Problem Report form at the back of the manual.

This printing of the 2427 Manual has been revised to reflect changes included in V2.8 terminal software.

The Tektronix^{TM} 4027 Color Graphics Terminal and the Tektronix^{TM} 4010 Computer Graphics Terminal are products of Tektronix, Inc., Beaverton, Oregon.

Retro-Graphics^{TM} is a trademark of Digital Engineering, Sacramento, California.

VT52^{TM} is a trademark of Digital Equipment Corp., Maynard, Massachusetts.

PREFACE: THE PURPOSE AND SCOPE OF THIS MANUAL

This document contains information needed to install and operate the Intecolor 2427 terminal. It also contains information needed to write host programs which can control the production of high resolution graphics and other displays at the terminal.

The document is intended for use as a training and reference manual by programmers, applications personnel, and terminal operators. The reader need not be familiar with the Tektronix™ terminals which are emulated by the 2427; nor does he need extensive knowledge of computer hardware.

Organization of Information

Part One provides an overview of terminal operation and a description of the terminal's keyboard, controls and indicators.

Part Two covers installation, interconnection and power-up.

Part Three covers operation of the terminal in its basic local modes. These modes are utilized when setting up for on-line operation and when checking terminal status. Methods for switching among various on-line modes are also described.

On-line modes of operation are described in the following three parts. Part Four describes operation of the terminal as an emulator of the Tektronix 4Ø27 terminal. Part Five describes operation of the terminal as an emulator of the Tektronix 4Ø1Ø. Part Six covers operation of the terminal in its Transparent Mode, in which it functions like an Intecolor 24Ø5.

Several appendices, giving summaries of 2427 commands, have been included as convenient reference tables.

Related Documents

Information pertinent to the alignment and maintenance of the 2427 can be found in the Intecolor 24ØØ Series Maintenance Manual (ISC stock #999379).

CONVENTIONS USED IN THIS MANUAL TO REPRESENT COMMANDS

A specific set of commands is associated with each of the terminal's on-line modes of operation. The command sets are fully described in Parts Four, Five and Six of the text. In addition, they are summarized in the appendices. The following conventions have been adopted for representing all commands used with the 2427:

- Commands are represented as sequences of ASCII characters and control symbols

- ASCII control symbols are represented by the standard abbreviations of their names, in upper case characters, enclosed between the signs for "less than" and "greater than". For example,

  <**CR**> = Carriage return (code 13 decimal, ØD hex)

  <**ESC**> = Escape (code 27 decimal, 1B hex)

- Command keywords, when used, are spelled out. Upper case characters are used to distinguish the elements of the keywords which are significant to the machine from those which are not. For example,

  VECtor = A command keyword used in 4Ø27 mode. Only the first three letters in the word are significant.

- In the representation of commands which must or can include a string of parameter values, necessary parameters are indicated by words or symbols which stand alone in the command. Optional parameters are indicated by words enclosed by brackets. For example,

  lines = A parameter for which a value must be specified

  {**lines**} = A parameter for which a value need not be specified

- Spaces are sometimes used between the elements of command sequences to improve legibility. These spaces do not always indicate required delimiters. However, the use of delimiters where spaces are given is always permissible.

- For keyboard-entered control sequences, two or more keys pressed simultaneously are shown within parentheses; for example

  (Control B) = The keyboard operation to transmit
  <STX> (Hex Ø2)

- A raised asterisk, "*", is used where necessary to distinguish a lower case L from the numeral one. For example,

  1^* = A lower case L
  1 = The numeral one

TABLE OF CONTENTS

APPENDICES

PART ONE

PART ONE: INTRODUCTION TO THE 2427

OVERVIEW

The Intecolor 2427 is a data terminal, a device which an operator uses to communicate with a host computer and to control auxiliary equipment like printers and plotters.

Though they are physically unified, the 2427's components have two logically distinct functions. The keyboard acts as an input device for the host computer, while the display screen acts as an output device for the host computer. These two devices can function in a simultaneous and independent fashion. Data can be transmitted from the keyboard to the host while different data are transmitted from the host to the display. This type of operation is called full duplex.

The 2427 belongs among those terminals which are popularly characterized as "intelligent" because they have computing capabilities of their own. These capabilities are provided by microelectronic devices, in particular by large scale integrated circuits which function as processors and memories. In general, the presence of such capabilities at the terminal is doubly advantageous: first, because the host's workload can be minimized (especially important in time-sharing situations); second, because host programming can be considerably simplified.

The 2427's computing capabilities allow it to respond to commands which are very simple in form, yet require very complex operations for their execution. The operations of which the terminal is capable include all those needed to construct multicolored displays of text and/or graphs from data supplied according to any one of four protocols. These protocols include the Tektronix 4010 and 4027 protocols, which were developed to facilitate the creation of high resolution graphics displays, the device independent and otherwise very flexible ANSI Standard X3.64 protocol, and the widely used DEC VT52 protocol.

The terminal's ability to respond to Tektronix 4010 and 4027 commands is complemented by its high resolution display capabilities. Accordingly, it can emulate the operation of the Tektronix terminals themselves and is especially useful for graphics work. On the other hand, it has additional standard resolution display capabilities which are utilized when the terminal responds to ANSI X3.64 or DEC VT52 type commands.

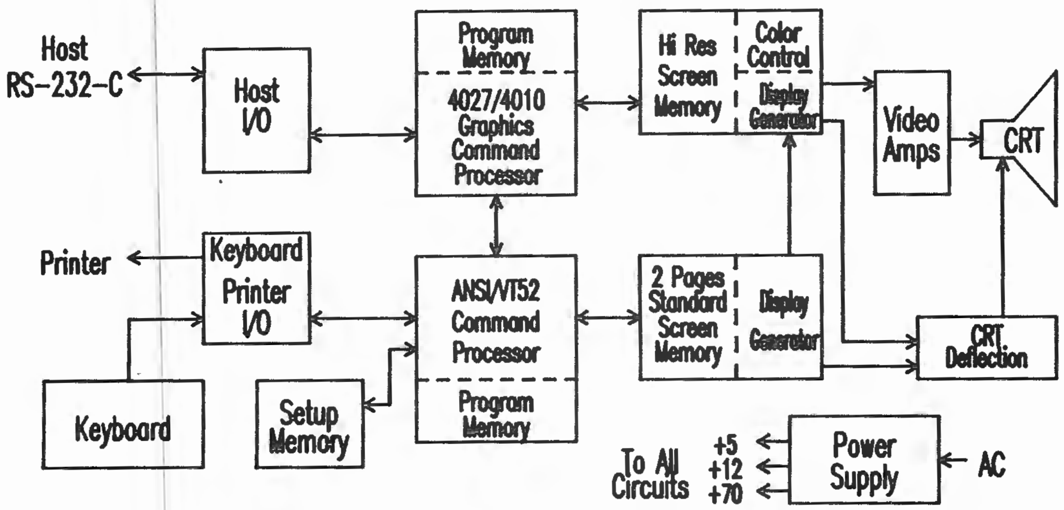
The 2427's high resolution color graphics capabilities are provided by logic circuitry developed by Digital Engineering, Inc. and marketed under the name Retro-Graphics^{TM}. Its additional capabilities are provided by circuitry originally developed by the Intecolor Corporation for the Model 2405 Color Graphics Terminal.

The 2427's main functional elements are represented in the diagram below (Figure 1.1). Note that data enter and leave the terminal in serial form through two bidirectional ports. One of these ports is used for the keyboard and for a printer. The other is used for the host.

Within the terminal, data are handled by two command processors. Each processor has its own program memory, but the processor programs are linked in such a way that the graphics processor has overall control of host communications. A Setup memory, used by both processors, stores the values defining terminal operating conditions such as baud rate, type of parity, etc. This memory is equipped with a battery which permits it to remember settings between operating sessions. (For more information on the Setup memories, see Part Three of this manual.)

The processors can read or write data at two display memories. One display memory is used for high resolution graphic displays and is addressed directly by the graphics processor. The other is used for standard resolution displays; it is addressed directly by the ANSI/VT52 command processor and indirectly by the graphics processor. Data from these memories are used by two display generators to construct video and sync signals. The video signals control the operation of three electron guns in the CRT. The sync signals control the production of signals which deflect the electron beams produced by the guns to produce a raster scan of the display screen. (Further information about display capabilities is provided in subsequent sections of the manual.)

2427 FUNCTIONAL BLOCK DIAGRAM



(Figure 1.1)

MODES OF OPERATION

- Setup Mode

  A local mode of operation, entered from the keyboard. Used to select the values of parameters governing host/terminal communications (baud rate, parity, etc.) and other operating conditions, and to check status.

  The terminal's display unit provides a Setup Menu representing current conditions. Menu items are selected from the keyboard.

- 4027 Mode

  An on-line mode, with selectable local echo, entered from the Setup Mode or from another on-line mode upon receipt of an appropriate command from the host or the keyboard.

  The terminal displays textual messages from the host or keyboard on the lower part of the screen (monitor) with character cell matrix techniques and high resolution graphs on the upper part of the screen (workspace) with dot matrix techniques.

  The terminal accepts inputs formed according to Tektronix 4027 protocol.

- 4010 Mode

  An on-line mode, with selectable local echo, entered after setting up for display in 4027 Mode. Entry may be made from Setup Mode or from 4027 Mode upon receipt of an appropriate command issued by the host or from the keyboard.

  The terminal's entire display area is devoted to high resolution graphics using dot matrix techniques.

  The terminal accepts inputs formed according to Tektronix 4010 protocol.

- Transparent Mode

  An on-line mode, with selectable local echo, entered from the Setup Mode or from another on-line mode upon receipt of an appropriate command from the keyboard or the host.

  The terminal operates as an Intecolor 24Ø5. Text and graphs are displayed with character cell matrix techniques.

  Communications with the host are handled either according to ANSI X3.64 protocol or according to the DEC VT52 protocol.

- Self-test Mode

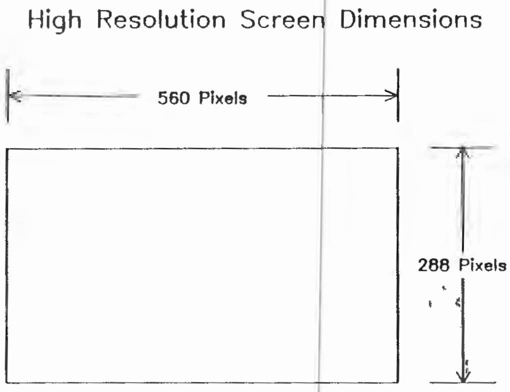
  A local mode, entered automatically at terminal power-up and upon the receipt of an appropriate command from the keyboard or host.

  The screen is used to display test patterns and error messages. When no error is found, the terminal reverts automatically to one of its on-line modes.

Operation of the terminal in the Setup Mode, the Key Mode and the Self-test Mode is described fully in Part Two of this manual. 4Ø27 Mode, 4Ø1Ø Mode and Transparent Mode operations are described in Parts Three, Four and Five.

are 56Ø pixels horizontally and 288 pixels vertically. Displays are built up on this matrix dot by dot; accordingly, the screen is said to be dot addressable.

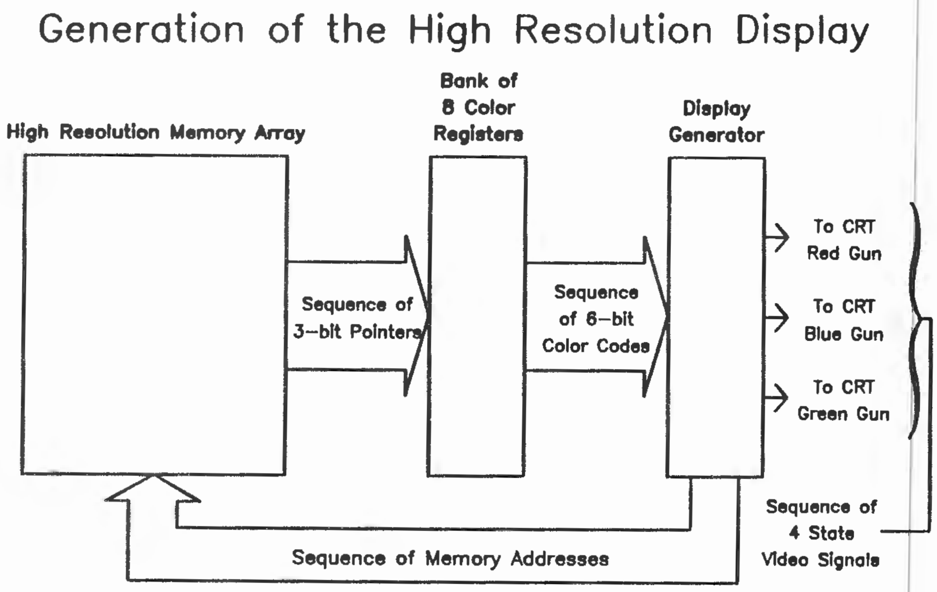
A colored dot is produced when beams of electrons, generated by some combination of the cathode ray tube's three guns, are directed upon a group of light-emitting phosphors, which are situated on the inside surface of the screen and constitute a pixel. The phosphors in a group are of three types (red, blue and green), each of which is excited by one of the beams.



(Figure 1.2)

The three beams are made to sweep over the whole matrix of pixels in a regular manner. As they are moved from pixel to pixel, they are turned on or off, and dots of various colors are produced. These dots constitute a pattern which seems to endure because the sweep is repeated over and over.

The dot pattern is defined by the contents of a large array of memory registers. A separate register can be assigned to each of the 161,28Ø pixels on the screen. The address of each assigned register belongs to a sequence which corresponds to the sequence in which the pixels are scanned by the cathode ray tube's electron guns to produce the display. Accordingly, when addressed by the display generator, the memory array can provide a sequence of codes in step with the sweep. The memory is said to represent a "bit map" (a sort of blueprint formed by a pattern of binary digits) which the display generator reads in order to construct the display.


Generation of the High Resolution Display
High Resolution Memory Array
Bank of 8 Color Registers
Display Generator
Sequence of 3-bit Pointers
Sequence of 6-bit Color Codes
To CRT Red Gun
To CRT Blue Gun
To CRT Green Gun
Sequence of 4 State Video Signals
Sequence of Memory Addresses

(Figure 1.3)

Each memory register can hold a three-bit code which "points" to one of eight color registers in the display generator. Each of these color registers holds a six-bit color code. Since a six-bit binary code can represent any one of 64 arguements, the contents of each color register define one of 64 displayable colors.

Each time a color register is selected by a pointer from the memory, the display generator uses the color code held by that register to produce a set of three video signals, one for each of the CRT's electron guns. The six-bit color code is broken down into three groups of two bits. Each two-bit group is then decoded to produce a video signal with one of four possible states: off, 25% on, 50% on and 100% on. Thus, while the sweep is covering the area of a single pixel, one or more of the guns can be switched on, and each gun can be on for 25%, 50% or 100% of the time the sweep is over that pixel. Accordingly, the phosphors in the pixel can be excited in one of 4^3 or 64 ways to produce one of 64 colors.

Since the contents of the color registers can be changed independently of the contents of the memory, the colors used in the high resolution display can be changed without changing the dot pattern, and vice versa.

In order to construct a display, the graphics command processor must convert data supplied to it from the host or keyboard into display memory and color register information. It must then write this information into the appropriate registers. The data supplied to the processor can include representations of coordinate pairs (in the form of decimal integers) in an (x,y) system. The dimensions of the coordinate system can vary within limits defined by the terminal's mode of operation. In all cases, the origin of the system corresponds to the lower left corner of the graphics display area, and the graphics processor converts coordinate pairs into the addresses of pixels in this area. Depending on the command given, these pixels represent plotted points, the beginning or end points of vectors, etc. which the processor "draws". The processor can "draw" both graphs and text on the high resolution screen. Special routines stored in the processor's program memory are used to construct alphanumeric characters on subdivisions of the display area which measure 7 x 8 pixels.

Standard Resolution Displays

The method used to create standard resolution displays involves the electronic division of the display area on the screen to form a matrix of cells in which characters or plot blocks can be represented.

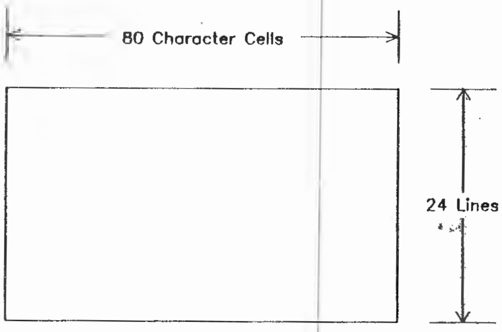
If the entire screen is used for standard resolution displays, the dimensions of the character cell matrix are 80 character cells horizontally and 24 character cells vertically. Accordingly, the display area can represent a page of text having 24 lines, each line having up to 80 characters.

Each character cell resolves to a 6 x 12 pixel matrix, in which characters and plot blocks are represented as a pattern of colored dots. However, the display is not dot addressable. Each character cell as a whole is treated as an element of the display. A memory register is assigned to each character cell, whose content is defined by a data word written into the associated register.

The address of each memory register corresponds to the position of a character cell in the display area, and the memory furnishes the standard resolution display generator with a sequence of data words in step with the progress of the sweep of the CRT's beams from cell to cell over the screen.

The data words stored in the standard resolution memory array define the contents of the character cells in two ways. On the one hand, they single out characters from the terminal's character set or plot block set (up to eight plot blocks can be displayed in a character cell, and there are 256 possible combinations of blocks).

Standard Resolution Screen Dimensions



(Figure 1.4)

On the other hand, they determine a background and foreground color combination for the display of the selected character or plot block(s) and the blink (if any) of the character. Eight background and eight foreground colors are displayable.

The data words held in memory do not represent directly the dot patterns and dot colors which represent characters and plot block groups. Instead, they point to registers in read-only memories where a code for each displayable character and color combination is permanently stored. Accordingly, when the contents of the standary resolution display memory are changed, the content of an entire character cell, i.e., a character and/or color, is changed.

The standard resolution display is under the direct control of the VT52/ANSI X3.64 command processor. The graphics command processor can control the standard resolution display indirectly, by sending appropriate commands to the other processor.

The 2427 contains two memory arrays which are dedicated to character cell graphics; each array can hold enough data to define one complete display. When the terminal is operated in the Transparent Mode, commands can be given by the keyboard or host to switch between arrays in order to change the page on display.

Screen Saver

A CRT (cathode ray tube) such as that used in the 2427 is subject to damage by the prolonged display of a fixed image on its screen. Such display tends to "burn" the image into the screen's phosphors.

The 2427 has automatic equipment which is designed to prevent this kind of damage and thereby prolong the life of the CRT by limiting the time a given image can be displayed on the screen. A display which is not changed within a period of ten minutes is automatically turned off. The display returns as soon as there is any activity from the host or keyboard.

The screen saver equipment may be enabled or disabled by means of Setup Mode commands. See Part Three for details.

THE KEYBOARD

The 2427's detachable keyboard is connected to the terminal with a coiled cable terminated with a standard telephone type plug. To insure proper operation, the cable should be securely plugged into the terminal <u>before</u> power is applied. If the cable plugged in while the terminal is under power the keyboard might not operate properly.

Three versions of the keyboard are available. These versions differ only with respect to the number of programmable function keys installed: none, twelve, or twenty-four. The function keys are labeled FØ through F23 and are situated across the top of the main keypad when installed.

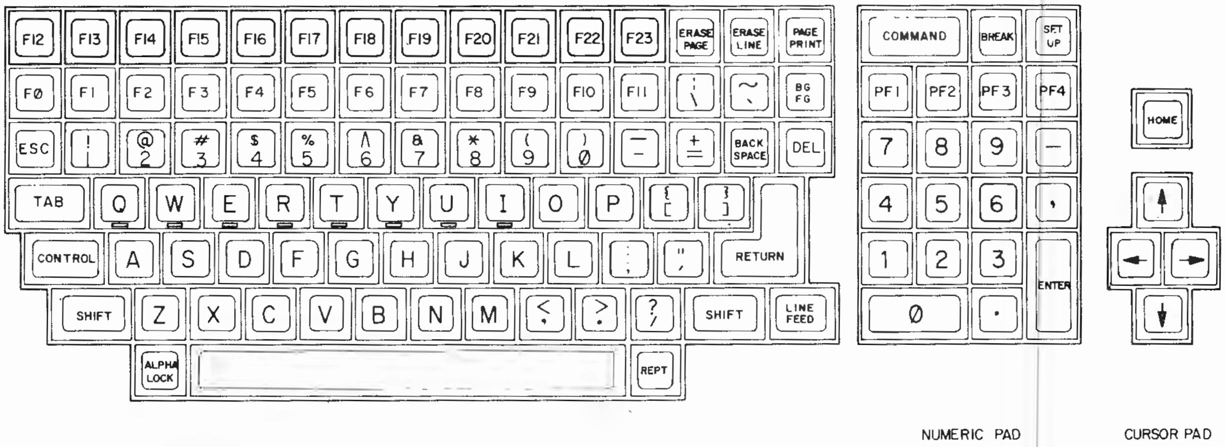
The keys are arranged in three major groups. The group of keys to the operator's right is used to control the display cursor, a blinking line or other visible sign which marks the position at which the next display operation will begin.

The keys grouped immediately to the left of the cursor pad are arranged in six rows. The top two rows contain special purpose keys whose functions are discussed below. The bottom four rows constitute a numeric keypad similar to those found on standard office machines.

The main keypad includes alphabetic and numberic keys organized like the keys on a standard typewriter. These keys have grey caps. Many of the remaining keys have special functions which are described below.

The keyboard is used to generate codes which are transmitted either to the host computer alone (on-line operation with no local echo), to the display unit alone (off-line or local operation), or to both host and display unit (on-line operation with local echo). These codes are ASCII binary equivalents of characters and control symbols or of groups of these characters and control symbols.

2427 Keyboard Layout



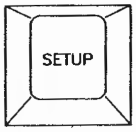
(Figure 1.5)

Within the terminal, the keyboard is assigned a buffer where keystrokes are recorded until the proper codes can be generated and sent to the host computer. If the processing of keystrokes is slowed down or held up while the operator continues typing, the buffer can fill up. If this occurs, an audible indication is given each time a new key is pressed and the keystroke is ignored. When the terminal has had time to process some of the keystrokes in the buffer new keystrokes are again accepted. A buffer overflow is very unlikely to occur unless the Repeat key is held down.

The functions of the various keys on the keyboard are described on the following pages in detail under the following headings:

- Special Purpose Keys
- The Alphanumeric Keys in the Main Keypad
- Keys Which Generate ASCII Control Codes
- Function Keys (FØ - F23)
- PF (Private Function) Keys
- Numeric Keypad Keys
- Cursor Control Keys

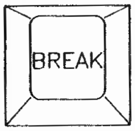
Special Purpose Keys



The Setup key is operated while holding down either of the Shift keys in order to place the terminal in its Setup Mode or to take it out of this mode. (See Part Two of this manual for details about Setup Mode.)

This key is used by itself to enable operation of the keyboard when it has been disabled by a command from the host or by the terminal's Self-test program.

This key never transmits a code to the host.

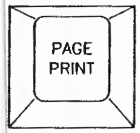


When the Break key is operated, the transmission line to the host is pulled to a logical low state for Ø.2333 seconds (with a ten-percent variation) and any operation being performed by the terminal processor (scrolling the display, for example) is interrupted.

The operation of this key has no effect when the keyboard has been disabled.

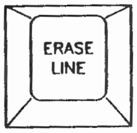
The Command Key is used in conjunction the alphanumeric keys to generate Escape sequences that may be assigned functions by the host program. The key is used by holding it down while striking another key. When operated by itself it generates no code.

The sequences generated when this key is used begin with <ESC>O. For example, the combination COMMAND P gives the following sequence: <ESC>OP



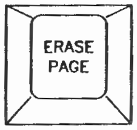
The Page Print key is used to transfer the contents of a standard resolution display to a printer without altering the display. It sends the following Media Copy (MC) command sequence: <ESC>[Øi

This key can be used when the terminal is operating in any mode.



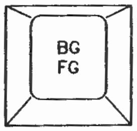
The Erase Line key is used to erase the contents of the display from the cursor position to the end of the line. It sends the following Erase Line (EL) command sequence: <ESC>[ØK

This key can be used only when the terminal is operating in the Transparent Mode (ANSI protocol).



The Erase Page key is used to erase the standard resolution display from the cursor position to the end of the page. The key generates the following Erase Display (ED) command sequence: <ESC>[ØJ

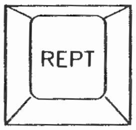
This key can be used only when the terminal is operating in the Transparent Mode (ANSI protocol).



The FG/BG key is used in the terminal's Transparent Mode only in order to change the colors with which characters and plot blocks are displayed. Color change is accomplished by striking this key and then one of the following alphabetic keys: Q,W,E,R,T,Y,U,I. These keys represent the colors black, red, green, yellow, blue, magenta, cyan, and white respectively.

Only characters displayed subsequently, beginning at the current cursor position are affected.

Used unshifted, the FG/BG generates a foreground color control sequence. Shifted, FG/BG generates a background color control sequence.



The Repeat key may be held down together with another key in order to transmit repeatedly the code or code sequence generated by that key.

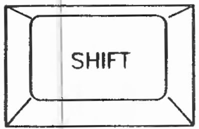
If this key is used to generate a very long string of codes while the terminal is operating at a low baud rate the keyboard buffer could overflow. Therefore, the terminal is equipped with a repeat rate controller, which lowers the repeat rate automatically if the buffer fills to a certain point. (See the PRKC commands in Part Six.)

The Alphanumeric Keys in the Main Keypad

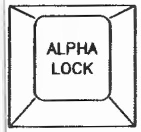
The alphanumeric keys (including the space bar), with grey caps, generate the ASCII codes for the uppercase and lowercase alphabetic characters, numerals and punctuation marks. When these codes are returned to the terminal, either locally or by the host, they can cause the display of the characters. In certain modes of operation, these keys can be used to display special graphics characters. (See Part Six of this manual for further information.)

The keys Q, W, E, R, T, Y, U and I are marked with colored bars. These keys can be used in conjunction with the FG/BG key to change the colors of the display when the terminal is operated in the Transparent Mode.

Operated by themselves, the alphanumeric keys generate the lowercase ASCII codes. They are commonly used in conjunction with three other keys, the two Shift keys and the Alpha Lock key, to produce the uppercase codes.



The two Shift keys are used interchangeably in conjunction with other keys to generate the uppercase ASCII codes. All the alphanumeric keys except for the Space Bar and DEL are affected by the Shift keys. The Shift keys are used by holding either of them down while operating another key. By themselves they generate no code.



The Alpha Lock key, when engaged, makes all the alphabetic keys uppercase. It does not affect numeric and special character keys.

The codes generated with the alphabetic keys are given in the table below as hexadecimal numbers.

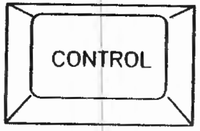
Key	Shifted	Unshifted	Key	Shifted	Unshifted
A	41	61	N	4E	6E
B	42	62	O	4F	6F
C	43	63	P	50	70
D	44	64	Q	51	71
E	45	65	R	52	72
F	46	66	S	53	73
G	47	67	T	54	74
H	48	68	U	55	75
I	49	69	V	56	76
J	4A	6A	W	57	77
K	4B	6B	X	58	78
L	4C	6C	Y	59	79
M	4D	6D	Z	5A	7A

The codes generated with the numeric and punctuation keys are given in the table below as hexadecimal numbers.

Key	Shifted	Unshifted	Key	Shifted	Unshifted
! 1	21	31	_ -	5F	2D
@ 2	40	32	+ =	2B	3D
# 3	23	33	| \	7C	5C
$ 4	24	34	~ `	7E	60
% 5	25	35	{ [	7B	5B
^ 6	5E	36	}]	7D	5D
& 7	26	37	: ;	3A	3B
* 8	2A	38	" '	22	27
(9	28	39	< ,	3C	2C
) 0	29	30	> .	3E	2E
Space Bar	20	20	? /	3F	2F
DEL	7F	7F			

The Keys which Generate ASCII Control Codes

The alphabetic keys and certain other keys can be used in conjunction with the Control key to generate the ASCII control codes. The control codes are distinguished from the character codes because they are not normally associated with displayable characters and usually signal special operations within the system.



The Control key is used by holding it down while operating another key or keys. It generates no code of its own.

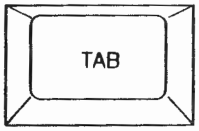
The table below shows how the various control codes are generated. The position of the Alpha Lock has no bearing on the generation of the control codes. However, certain keys must be Shifted, as noted.

Control +			Control +		
Key	Mnemonic	Hex	Key	Mnemonic	Hex
@*	NUL	00	P	DLE	10
A	SOH	01	Q	DC1 (XON)	11
B	STX	02	R	DC2	12
C	ETX	03	S	DC3 (XOFF)	13
D	EOT	04	T	DC4	14
E	ENQ	05	U	NAK	15
F	ACK	06	V	SYN	16
G	BEL	07	W	ETB	17
H	BS	08	X	CAN	18
I	HT	09	Y	EM	19
J	LF	0A	Z	SUB	1A
K	VT	0B	[	ESC	1B
L	FF	0C	\	FS	1C
M	CR	0D	]	GS	1D
N	SO	0E	^*	RS	1E
O	SI	0F	_*	US	1F

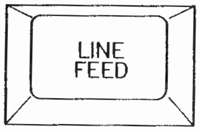
*Shifted Key

When the terminal is operated in the Transparent Mode (ANSI protocol) control symbols can be represented on the screen following the issuance of special commands. (See Part Six, especially the section on the CRM commands, for further information.)

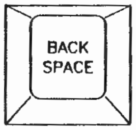
Special keys are included for generating certain very frequently used control codes by means of single key operations.



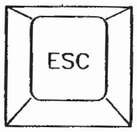
This key can be used instead of the (Control I) combination. It sends the ASCII horizontal tab code <HT>.



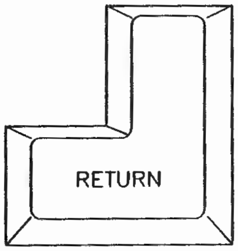
This key is used instead of the (Control J) combination to send the line feed code <LF>. When held down, it sends the code repeatedly.



This key is used instead of (Control H) to send the ASCII back space code <BS>. When held down it sends the code repeatedly.



This key is used instead of (Control [) to send the ASCII Escape code <ESC>. This code is often used in command and control sequences. Such sequences can be formed by striking the Escape key and then the other key or keys needed.



This key is used instead of the (Control M) combination to send the ASCII carriage return code <CR>.

Function Keys

Each of the optional keys labeled FØ through F23 can be programmed by the user to send sequences of up to 4Ø ASCII character codes. Programming of these keys can be done from the keyboard when the terminal is in the Setup Mode and from the Host when the terminal is in the Transparent Mode. See Part Three and Part Six (PPFN commands) for further information.

When these keys are not specially programmed by the user, they are assigned default sequences as shown in the table on the next page.

Default Assignments for Function Keys

Key	Unshifted	Shifted	Control
FØ	<ESC>O	<ESC>OØ	<ESC>O@
F1	<ESC>O!	<ESC>O1	<ESC>OA
F2	<ESC>O"	<ESC>O2	<ESC>OB
F3	<ESC>O#	<ESC>O3	<ESC>OC
F4	<ESC>O$	<ESC>O4	<ESC>OD
F5	<ESC>O%	<ESC>O5	<ESC>OE
F6	<ESC>O&	<ESC>O6	<ESC>OF
F7	<ESC>O'	<ESC>O7	<ESC>OG
F8	<ESC>O(	<ESC>O8	<ESC>OH
F9	<ESC>O)	<ESC>O9	<ESC>OI
F1Ø	<ESC>O*	<ESC>O:	<ESC>OJ
F11	<ESC>O+	<ESC>O;	<ESC>OK
F12	<ESC>O,	<ESC>O<	<ESC>OL
F13	<ESC>O-	<ESC>O=	<ESC>OM
F14	<ESC>O.	<ESC>O>	<ESC>ON
F15	<ESC>O/	<ESC>O?	<ESC>OO
F16	<ESC>OP	<ESC>O`	<ESC>Op
F17	<ESC>OQ	<ESC>Oa	<ESC>Oq
F18	<ESC>OR	<ESC>Ob	<ESC>Or
F19	<ESC>OS	<ESC>Oc	<ESC>Os
F2Ø	<ESC>OT	<ESC>Od	<ESC>Ot
F21	<ESC>OU	<ESC>Oe	<ESC>Ou
F22	<ESC>OV	<ESC>Of	<ESC>Ov
F23	<ESC>OW	<ESC>Og	<ESC>Ow

PF (Private Function) Keys

The PF keys generate private (machine specific) code sequences that may be sent to the host.

The codes generated by these keys and their normal uses are indicated below.

Key	Code
PF1	<ESC>OP
PF2	<ESC>OQ
PF3	<ESC>OR
PF4	<ESC>OS

Numeric Keypad Keys

The numeric keypad consists of the block of keys below the PF keys. The keys on the numeric keypad normally transmit the ASCII codes for the numerals, decimal point, minus sign (hyphen) and comma. The ENTER key transmits the carriage return code. (PKPN Mode.) Thus they simply duplicate the functions of the corresponding keys in the main keypad. These keys are intended mainly as a convenience for users who must enter large amounts of numeric data.

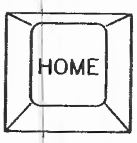
When the terminal is operated in the Transparent Mode (ANSI protocol) it may be commanded to assign special codes to these keys. See Part Six, especially the section on the PKPA (Keypad Application Mode) commands, for further details.

Codes Generated by the Numeric Keypad Keys

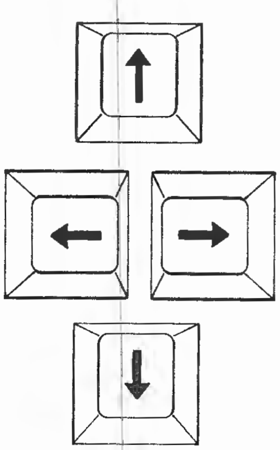
Key	Code in PKPN Mode	Code in PKPA Mode
Ø	Ø	<ESC>Op
1	1	<ESC>Oq
2	2	<ESC>Or
3	3	<ESC>Os
4	4	<ESC>Ot
5	5	<ESC>Ou
6	6	<ESC>Ov
7	7	<ESC>Ow
8	8	<ESC>Ox
9	9	<ESC>Oy
-	-	<ESC>Om
,	,	<ESC>Ol*
.	.	<ESC>On
ENTER	<CR>	<ESC>OM

Cursor Control Keys

The five keys grouped together at the right of the keyboard, including the Home key and the Arrow keys, are used primarily for the control of the display cursor. Cursor control with these keys requires that the codes generated when they are are struck be echoed back to the terminal, either by the host or within the terminal itself (local echo).



This key generates a control sequence that moves the cursor to the home position on the display. The home position is the upper left-hand corner of the display area.



These keys generate codes that move the cursor in the direction of the arrows on the keys, one character cell per keystroke. When these keys are held down their codes are generated repeatedly.

The code sequences generated by these keys are given in the table below. Assignments of codes are changeable in Transparent Mode with ANSI PCKM sequences. Default assignments are those of PCKM reset.

Codes Normally Generated by the Cursor Keys

	Code with	Code with
	PCKM set	PCKM reset
Key	(Special)	(Default - Cursor)
Up Arrow	<ESC>OA	<ESC>[A
Down Arrow	<ESC>OB	<ESC>[B
Right Arrow	<ESC>OC	<ESC>[C
Left Arrow	<ESC>OD	<ESC>[D
Home	<ESC>OH	<ESC>[H

When the terminal is operated in the Transparent Mode using the VT52 protocol, these keys are automatically assigned VT52 cursor codes. See pages 6.41 - ff.

TERMINAL CONTROLS AND INDICATORS

The Power Switch and Line Fuse

The Power Switch for the Intecolor 2427 is in the back of the terminal on the left-hand side as you face the keyboard. The terminal should not be turned off and then back on immediately. A few seconds should be allowed between power-down and power-up to permit the terminal's power supplies to discharge. Otherwise the power supply control circuits will shut off power within the unit. In addition, reset might not occur.

A line fuse holder is installed on the rear panel next to the power switch. The fuse sometimes blows on account of a momentary surge on the line. It should always be replaced with another fuse of the proper rating. The rating depends on the line voltage as shown:

22ØV	115V
Type 3AG	Type 3AG
Slo-Blo	Slo-Blo
.75A/25ØV	1.5A/25ØV

If the fuse blows repeatedly, a properly qualified service technician should be called in to determine the problem.

Audible Indicators

The Intecolor 2427 produces two audible indicators: a click and a bell.

- Click — The keyclick is an audible indication to the operator that a keystroke has been received by the terminal. The click can be disabled or enabled in the Setup Mode. The Shift, Control and Command keys do not produce clicks, and none of the keys produce clicks when the host has locked the keyboard.

- Bell — The bell can sound for a variety of reasons. It can be sounded when the host transmits the ASCII bell code <BEL>, or when some particular condition in exists in the terminal, when, for example the keyboard buffer is full. The bell also sounds on Power-up to indicate that reset has occurred.

PART TWO

PART TWO:

INSTALLATION, INTERCONNECTION AND POWER-UP

UNPACKING AND INSPECTING A NEW TERMINAL

The 2427 is shipped from the factory in a reinforced cardboard carton packed with plastic foam. When received, the carton should be inspected for damage **before** the unit is unpacked. If damage is discovered, the shipper and Intecolor Corporation should be notified.

After unpacking and **before** installation and initial power-up, the terminal itself should be subjected to a careful visual inspection. It is a good idea to remove the cabinet cover and to verify that all cables within the unit are properly connected and that the circuit boards and CRT are secure. (The cover is held by two screws situated at sides of the cabinet.) Any damage should be reported to Intelligent Systems Corp.

While the cover is off, the setting of SW2, the line voltage selector switch mounted on the main circuit board, should be checked. This switch must be set for either 115V or 22ØV, in accordance with the line voltage at the site of installation.

> **WARNING:** Work within the console should be performed only by persons who are aware of the possible hazards and are trained and qualified for this type of work.
>
> Hazardous voltages may exist within the unit when power has been applied.

THE OPERATING ENVIRONMENT

When installing the 2427, it is important to take account of environmental factors at the site which could affect the terminal's performance.

The unit's **ambient temperature** should be maintained between Ø° and 4Ø° Centigrade. It is especially important that the specified maximum temperature not be exceeded, because the 2427 relies heavily on convection for heat dissipation while in operation. Temperatures below Ø° C can cause damage to electrolytic capacitors in the unit.

The terminal should not be situated within strong **magnetic fields,** such as are present around large transformers. Magnetic fields can cause distortion of the display on the CRT screen.

Dust and smoke particles can cause problems since they are attracted by the unit's high voltage components and can collect at ventilating holes.

In especially **dry environments,** where electrostatic charges are easily created, it may be necessary for the operator to take precautions lest his handling of the equipment result in a discharge which could damage an integrated circuit. In such situations the operator can generally prevent damage by "grounding" himself before touching the keyboard.

The 2427's display is visible under normal indoor **lighting** conditions, and its brightness control compensates for some variation from such conditions. Accordingly, the area in which the 2427 is installed normally need not be darkened. However, the screen should be kept out of the glare of especially bright artificial light or sunlight.

IMPORTANT NOTE: Radio frequency emissions from the 2427 do not exceed the limits set by the FCC (in its Rules, Part 15, Subpart J) for class A computing devices. Nevertheless, the unit does radiate some radio frequency energy and accordingly can interfere with communications equipment (including but not limited to television receivers) and certain types of electronic instruments operating in the immediate vicinity. Precautions should be taken where appropriate.

The inside surfaces of the 2427's cabinet are coated with a conductive paint which serves as a shield for the equipment. The cabinet cover must be in place if RF emissions are to be minimized.

POWER-UP AND PRELIMINARY TESTING

WARNING: Before applying power to the terminal, verify that the line voltage selector switch, SW2, on the main circuit board is properly set and that the proper fuse is installed in the fuse holder on the rear panel. The fuse should be

1.5 Ampere 25ØV "Slo-Blo", type 3AG, for 115V operation, or

.75 Ampere 25ØV "Slo-Blo", type 3AG, for 22ØV operation.

For continued protection against fire and equipment damage, the fuse should be replaced only with a fuse of the proper type and rating.

WARNING: Operate the unit from a single-phase AC source only. The power supply protection circuit is not designed for two-phase sources. Do not defeat the purpose of the three-prong power plug (safety) by cutting or otherwise disabling the ground prong.

WARNING: The upper line voltage limit (25Ø VAC) must not be exceeded. The line filter, fuse, fuseholder and switch are rated at 25Ø VAC maximum.

After the terminal has been inspected it may be connected to an AC source, switched on and tested. Initial tests should be made with the host and any peripheral devices disconnected, but with the keyboard connected.

Note: The keyboard should always be connnected before power is applied.

Before attaching the power cord, verify that the lower edge of the white power switch on the rear panel is depressed (the off position). Insert the power cord into its socket on the rear panel and then connect it to a source of single-phase AC power, 5Ø Hz or 6Ø Hz.

Turn the power switch to the on position (upper edge depressed).

Note: After applying power, it may be necessary to adjust the brightness control, a potentiometer mounted on the rear panel, to make the display visible.

The 2427 automatically goes through a self-test routine immediately after power is applied. If the self-test detects a problem, an error character (@, A, C, G, or O) is displayed in red on the screen. (For information on the error codes see Part Three of this manual.) Otherwise, upon completion of the test routine, the terminal begins executing its initialization routine. Initialization involves communication between the ANSI/VT52 processor and the 4Ø27/4Ø1Ø graphics processor within the terminal. Data stored in the terminal's permanent Setup memory registers are read and used to set initial operating conditions in both processors.

During this time (a few seconds) the terminal is unable to communicate with a host. Then the terminal displays the following multicolored header message at the top of the screen:

```
INTECOLOR 2400 V2.8
```

This message gives the version of the software installed for the ANSI/VT52 command processor (V2.8). Early versions of the 2427 have V2.7 software. (See note on page 2.6.)

The appearance of this message indicates that initialization is complete and that the terminal is operating in its default power-up or entry mode (normally 4027 Mode). Certain default operating conditions (baud rate, etc.) are now established and the terminal is ready for communication. It will accept input from the keyboard.

<u>Note</u>: It is possible for the terminal to power-up displaying a Setup Menu instead of the header messages described above. The appearance of the Setup Menu at power-up indicates that no data are stored permanently in the Setup memory which would allow initialization to be completed.

In this case, certain Setup conditions must be established by the operator to make the terminal fully operational. Setup is fully discussed in Part Three of this manual. For purposes of initial testing, the following Setup selections should be made:

Main Menu:		
	NEW LINE	YES
	WRAP	YES
	PARITY	OFF
	ECHO	NO
	LOCAL/LINE	LINE
	HOST RATE	9600
Graphics Submenu:		
	ENTRY MODE	4027
	GRAPHICS LOCAL ECHO	NO
	TMODE CHAR	CAN

Then SAVE SETUP, on the Main Menu, should be selected. (The settings of other Setup Menu items are not important at this time.)

After these Setup selections have been made, the terminal should be powered down and, after a few seconds, powered up again. The header message described above should appear.

If the terminal repeatedly goes into Setup Mode at power-up, the Setup memory's battery back-up is probably faulty and should be checked by a competent technician.

Assuming the header message appears as described, the terminal can be tested to insure that it is properly adjusted. A testing proceedure is described on the following pages.

Initial Test

Must be in TRANS mode.

The terminal should be placed in an off-line condition to check for display alignment and proper character and color generation. This is done as follows:

1) While holding down a "Shift" key, strike the "Setup" key. A Set-Up Menu should appear.

2) Observe the setting of the Local/Line option as given in the menu. If necessary, depress the "J" key to select "Local".

3) Next, strike the "Set-up" key with the "Shift" key. The Setup Menu should disappear and the following header message should appear:

```
INTECOLOR 2400 V2.8
```

The following tests can now be performed:

1) Color generation and can be checked by creating rasters in each of the eight colors. This is done using the following sequence of keystrokes:

   "Home", "Shift" together with "BG FG", "(one of the color keys, Q-I)", "Erase Page"

   Each color should be pure and distinct.

   A white raster is good for checking the straightness of the edges of the display. Edges with pronounced curves indicate a need for "pincushion" adjustment.

   Note the size and position of the white raster. It should be 9.5 inches wide, 7.25 inches high and centered on the screen. Vertical and horizontal size and centering adjustments are indicated if these specifications are not met.

2) The character memory can be checked after selecting a background color (using the keystrokes described above) and a different foreground color, using the sequence of keystrokes

   "Home", "BG FG" (unshifted), "(color key)".

   The keyboard can then be used to put all the characters on the screen.

To check the special characters, strike the "N" key while holding down the "Control" key. Subsequently striking the unshifted alpha keys will then produce the special characters.

To shift back to the standard characters hold down the "Control" key and strike the "O" key.

Note: The control representation characters and the plotting blocks are not examined in this test.

3) The linearity of the display can be checked by filling the screen with L's or T's. This is done using the following sequence of keystrokes:

"Home", "Erase Page", "(L or T)", "ESC", "#", "8"

The vertical and horizontal components of the character should be aligned everywhere on the display.

These tests check deflection circuit adjustments and basic Transparent Mode display generation only. Further tests are described in Part Three of this Manual.

If these preliminary tests indicate the need for adjustments, the Maintenance Manual for the Intecolor 2400 Series terminals should be consulted for adjustment proceedures. Adjustments should be performed only by properly qualified service technicians.

Once the terminal is found to be operational and properly adjusted, it may be connected to the host and any peripheral equipment and placed in service.

Note: The tests described above are performed following the selection of Local Mode operation. In early terminals (V2.7 software), once Local Mode has been invoked, the terminal may not be returned to on-line operation merely by reseting Item J on the Main Setup Menu to "Line". It is necessary to power-down and subsequently power-up in order to reestablish communication between the terminal's two processing units. The "Line" selection should always be saved in the unit's permanent Setup Memory to permit the power-down/power-up sequence to have the proper effect. Terminals with V2.8 software may be returned to on-line operation simply by changing the setting of item J on the Setup Menu to "Line" and exiting Setup Mode.

Setup for on-line operation is described fully in Part Three.

INTERCONNECTION

Three connectors are provided at the rear panel of the 2427 for connection of the terminal to other equipment.

J4 is an 8 pin modular connector for the **keyboard.** Since the keyboard and its interconnecting cable are part of the 2427, the technician will not ordinarily be concerned with J4's pin assignments when he is installing the unit. However, they are given here for reference:

1 - Ground (cable shield)
2 - Serial Data Input from keyboard (TTL levels)
3 - Clear-to-Send (active high) to keyboard

> Note: This signal is high at all times when jumper W3 on the logic board is installed. For a Clear-to-Send signal under software control W3 must be removed and W4 installed. Normally, W3 is used. See the schematic diagram, drawing number 1Ø1959, sheet 2, for details.

4 - Ground
5 - +5 V to keyboard
6 - Ground
7 - +5 V to keyboard
8 - Ground

J5 is a 25 pin female connector for a **printer.** Its pin assignments are:

1 - Ground
2 - Serial Data Output to printer
5 - Clear-to-Send (active high) from printer
7 - Ground
2Ø- Data-Terminal-Ready (high) to printer

> Note: The terminal only transmits and does not receive data at this connector. Only the pins listed are used.
>
> Terminals in production as this edition of the manual goes to press (11/22/83) are not equipped with software to support use of the printer port for connection of graphics printers. Standard text printers may be connected when the terminal is operated in transparent mode.

J6 is a 25 pin female connector used to connect the terminal to a **host** or other device, either directly or by way of a modem or current loop. Its pin assignments are:

1 - Ground
2 - Serial Data Output (RS-232 levels) to host
3 - Serial Data Input (RS-232 levels) from host
4 - Request-to-Send (active high) to host
5 - Clear-to-Send (active high) from host
6 - Data Set Ready (active high) from host
7 - Ground
8 - Data Carrier Detect (active high) from host
9 - NC
1Ø- NC
11- Serial Data Input (current loop/+) from host
12- NC
13- NC
14- NC
15- External Clock Input

> Note: Used for synchronous transmission to host. Requires jumper W13 on the logic board and the reprogramming of the USART.

16- NC
17- External Clock Input

> Note: Used for synchronous reception of data from host. Requires installation of jumper W12 on the logic board and the reprogramming of the USART.

18- Serial Data Input (current loop/-) from host
19- NC
2Ø- Data Terminal Ready (active high) to host
21- Serial Data Output (current loop/+) to host
22- NC
23- NC
24- External Transmit Clock Output
25- Serial Data Output (current loop/-) to host

PART THREE

PART THREE: SETUP, TESTING AND ON-LINE

OPERATION

SETTING UP THE TERMINAL

The Intecolor 2427 has a Setup Mode of operation which allows the operator to set the terminal's operating conditions (baud rates, display format, etc.) from the keyboard while referring to a plain language menu. This feature eliminates the need for switches which are present on some terminals. Several settings must be made before the terminal can be operated on-line. Others are optional.

The terminal has two memory register sets dedicated to Setup data. One register set is operative at all times, even when the terminal is disconnected from the power line. The other is operative only when AC power is applied to the terminal. When the terminal is placed in operation, the contents of the first register set are automatically loaded into the second (temporary) set and the terminal is set up according to the contents of the second set. The data in this register set can be read and changed by the operator when he places the terminal in the Setup Mode.

> **NOTE:** Changing the data in the temporary Setup registers results in a new setup for the current operating session. However, it can have no effect on the way the terminal will come up in subsequent operating sessions unless a Store Setup command is issued while the terminal is in the Setup Mode. (See below.) When the terminal receives this command it adjusts the contents of the first (permanent) register set to agree with the contents of the temporary register set.
>
> If the permanent Setup registers are empty at power-up, the terminal automatically enters the Setup Mode. This happens very rarely, usually because the memory's battery has failed. (The battery has a life expectancy of about five years.)

The terminal enters or exits the Setup Mode when the Setup key is operated while a Shift key is held down. (An exit from Setup Mode may also be initiated by striking the ESC key.) This Mode may be entered from any other mode of operation except the Self-Test Mode. If Setup Mode is entered when the terminal is on-line and receiving data from a host, the host is automatically signaled to stop sending until the operator returns the terminal to on-line operation. If a standard resolution display is present when Setup Mode is entered, this display disappears until the terminal leaves Setup Mode, at which time the original display returns. (The second page of the standard resolution display memory is used in Setup Mode.) If a high resolution display is present, this display remains visible while the terminal is in Setup Mode.

When the Setup Mode is entered a series of menus is displayed on the terminal screen, showing the terminal's current communications settings and other current conditions. An example of the main Setup Menu is given below.

```
      SETUP FUNCTION - To exit SETUP MODE, press SHIFT + SETUP key

          A)    XON/XOFF              YES

          B)    MARGIN BELL           NO

          C)    KEYCLICK              NO

          D)    WRAP                  NO

          E)    NEW LINE              YES

          F)    SCREEN SAVER          NO

          G)    PARITY                OFF

          H)    CURSOR TYPE           _

          I)    ECHO                  NO

          J)    LOCAL/LINE            LINE

          K)    HOST RATE             192ØØ

          L)    PRINTER RATE          3ØØ

          M)    COLOR PRINTER         NO

          N)    ANSI 3.64             YES

          O)    TABS

          P)    ANSWERBACK

          Q)    FUNCTION KEYS

          R)    STORE SETUP

          S)    GRAPHICS SUBMENU

INTECOLOR 24ØØ V2.8 (C) Copyright 1983 by Intelligent Systems Corp.
Color Retro-Graphics V1.3 (C) Copyright 1983 by Digital Engineering, Inc.
```

Down the left-hand side of the menu are letters which indicate the keys to be used to change settings or to call up other menus. The middle column contains the names of the functions, parameters, etc. which are in question. The right-hand column shows the current settings.

Most of the items on the menu have only two possible settings. The keys associated with these items are used to toggle between settings. When an item (Host Rate, for example) has more than two possible settings, striking its associated key results in the display of a new menu showing a set of possible settings and the keys to be used to select them.

There is no provision in Setup Mode for adjusting the terminal's display generators to the power line frequency (5ØHz for most 22ØV single phase power lines throughout the world, 6ØHz for 115V single phase lines in the United States) because this is done automatically by the logic circuitry.

Also, there is no provision for setting the number of stop bits used in host/terminal communications. The number of stop bits is fixed at 1.

Items on the Main Setup Menu

XON/XOFF

Menu key: **A**
Settings: YES, NO

The 2427 is capable of automatically sending XON and XOFF codes to the host computer and responding to these codes when received from the host (software handshaking). It only uses this capability when the XON/XOFF option is enabled. When this option is enabled the terminal sends an XOFF when any of the following occur:

1) The terminal is placed in Setup Mode.

2) (Control S) is entered at the keyboard.

3) The receiving buffer becomes half-full.

An XON is sent only after an XOFF has been sent and (Control Q) is entered at the keyboard or the receiving buffer drops below one-fourth full.

MARGIN BELL

Menu key: **B**
Settings: YES, NO

When the Margin Bell is enabled the 2427 sounds a bell when the cursor is eight characters from the end of the current line. This feature is much like the margin bell on a typewriter.

KEYCLICK

Menu key: **C**
Settings: YES, NO

The keyclick is a sound generated by the terminal every time a key other than CONTROL or SHIFT is pressed. Most typists find the keyclick helpful for more accurate typing. If the host computer locks the keyboard, the keys will not generate a click.

WRAP

Menu key: **D**
Settings: YES, NO

This option, when enabled, causes the cursor to be automatically repositioned at the beginning of the next text line when a given line has been filled. If the option is not enabled, the cursor must be repositioned with commands inserted into the text stream; otherwise, characters are placed at successive positions as received until the end of a line is reached, at which point subsequent characters are all placed in the last character cell of that line.

NEW LINE

Menu key: **E**
Settings: YES, NO

This option, when enabled, causes line feed codes received to be treated as carriage return and line feed codes.

SCREEN SAVER

Menu Key: **F**
Settings: YES, NO

The function of the screen saver is described in the Introduction. When the F key is struck the current setting for the standard resolution screen saver is changed (YES = screen saver on). There is a separate screen saver for the high resolution screen which is set from the Graphics Submenu.

PARITY

Menu key: **G**
Settings: OFF, EVEN, ODD

When the G key is struck, a Parity Menu appears on the screen. The menu has three items, as shown:

A) OFF
B) EVEN
C) ODD

A selection is made by striking the key whose letter is associated with the desired parity setting. If the operator does not wish to change the current selection, he uses the Shift-Setup key combination to return to the Main Menu. When a new selection has been made, the main Setup Menu reappears on the screen, and the selection appears beside Main Menu item G.

When parity is enabled, the terminal tests for errors in received data stream according to the function selected; it also modifies data to be transmitted to the host to permit parity checking by the host. When an error is detected by the terminal, it displays an error character in place of the character which was improperly received.

Parity checking requires the use of the high order bit in each data byte sent or received. Accordingly, when parity is enabled there are seven data bits per data word and the seven-bit ASCII code can be used. However, special eight-bit codes cannot be used unless parity checking is disabled to make the eight bit in each word available for data.

CURSOR TYPE

Menu key: **H**
Settings: BLK _, _, BLOCK, BLK BLOCK, NONE

When the H key is struck a Cursor submenu appears.

The menu contains five items, as follows:

A)	BLK _	(Blinking Underscore)
B)	_	(Underscore)
C)	BLOCK	(The block cursor occupies one entire character cell.)
D)	BLK BLOCK	(Blinking block)
E)	NONE	

A selection is made by striking the key whose letter is associated with the desired cursor type. After a selection is made, the main menu reappears showing the cursor type selected. Only the Transparent Mode cursor and 4Ø27 Mode monitor cursor are affected by a selection.

ECHO

Menu key: I
Settings: YES, NO

This option allows the operator to determine whether or not the keyboard input being routed to the host will also be routed directly to the display. Local echo is selected by setting item I to "YES". Since many host computers echo everything they receive, they make the use of local echo unneccessary and superfluous (characters echoed by terminal and host would be displayed twice). Accordingly, the option can be disabled by selecting "NO".

This option refers to all on-line operating modes. (In early versions of the 2427, this option affects only echo to the standard resolution displays and a separate echo function for high resolution displays is available on the Graphics Submenu.)

Note that the use of the keyboard for display control usually requires that some type of echo be enabled.

HOST RATE

Menu key: **K**
Settings: (see Table 3.1)

When the K key is pressed, a menu of baud rate options is displayed. To select a new baud rate, press the key beside the rate to be used. Table 3.1 shows the various communication rates that the 2427 can use and their corresponding selection keys.

Table 3.1 - Baud Rates

A)	50	I)	1050
B)	75	J)	1200
C)	110	K)	1800
D)	134.5	L)	2000
E)	150	M)	2400
F)	200	N)	4800
G)	300	O)	9600
H)	600	P)	19200

PRINTER RATE

Menu key: **L**
Settings: (see Table 3.1)

This item is much like the host rate except that it selects the communication rate for an auxiliary device connected at the terminal's serial output port (printer).

COLOR PRINTER

Menu key: **M**
Settings: YES, NO

This option allows the user to signal the terminal that the auxiliary device connected at the printer port can accept and process standard ANSI command sequences for color changes and format control. If this option is enabled, the terminal passes these command sequences on to the auxilliary device. If the option is disabled, these sequences are suppressed. (See FETM in Part Six.)

ANSI 3.64

Menu key: **N**
Settings: YES, NO

This option allows the user to specify that the terminal respond to standard ANSI command sequences or to an alternate set of command sequences when operating in Transparent Mode. If this option is enabled, the terminal responds to the ANSI and private command sequences specified in Part Six. If this option is disabled, the terminal responds to VT52 type command sequences (see PATM in Part Six.)

TABS

Menu key: **O**
Settings: (See Figure Below)

This option permits setting the horizontal tabulation stops to be used with Transparent Mode and 4027 Mode Monitor tab commands and with the <HT> terminal control code. (See TAB in Command Group F, Part Five, and CHT and CBT in Part Six.) The 4027 Mode workspace tab stops are set with a 4027 Mode command (STOps).

In Transparent Mode, the 2427 has two horizontal tab modes, line-tab mode and page-tab mode. In line-tab mode, each line has tabulation stops in the same columns. In page-tab mode, the horizontal tabulation stops are set on specific line and column positions, which means that one line might have three tabulation stops while the next line might have five tabulation stops at different positions from the previous line. Page-tab mode tabulation stops can only be set by special Transparent Mode control sequences. (see TSM and CTC in Part Six.)

When the M key is pressed, the terminal is placed in line-tab mode, and default horizontal tab stops are set at intervals of eight columns. A ruler is displayed showing the tab stops that are set (see Figure 3.2). A tabulation stop is denoted by the letter T under the column on the ruler where the tabulation stop is set. The cursor is placed on the ruler. The Space bar, the Tab key, the Return key, and the left and right Arrow keys can be used to move the cursor over the ruler. To clear a tabulation stop, move the cursor to the tabulation stop to be cleared and then strike the C key. To set a tabulation stop, move the cursor to the column that at which the tabulation stop is to be set and then strike the T key. The tab indicators will reflect any changes made to the tabulation stops.

Figure 3.2

```
....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
     T   T     T       T       T       T       T       T       T
```

Pressing the SETUP key while holding the Shift key down stores the new tab settings and restores the main Setup menu.

ANSWERBACK MESSAGE

Menu Key: P
Settings: Answerback response message defined by operator

This option permits the definition of an identifying message of up to twenty characters to be transmitted by the terminal to the host computer when the host issues an enquiry <ENQ>. This response to a host computer inquiry will take place without the knowledge of the operator and without altering the data on the screen.

To review or define the terminal's answerback message, press the N key. The current answerback message is displayed on the screen. If no changes are desired, the main Setup menu can be restored by striking the Setup key while holding a Shift key down. To define the answerback message strike the FØ key. The terminal is then ready to accept up to twenty characters, entered at the keyboard, as the answerback message. To terminate the new answerback message, press the F1 key. The terminal then stores the new answerback message and restores the main Setup menu.

Note: The answerback message can contain the characters generated by any key except the function keys themselves and the SETUP key.

Note: To correct an error made while defining the answerback message, hold down the Command key while striking the Back Space key.

FUNCTION KEYS

Menu Keys: Q
Settings: Function key definitions

The function keys are keys that can be programmed to generate sequences of up to 4Ø characters when struck. Each function key can generate three different sequences, one for the unshifted key, one for the Shifted key, and one for when the key is used with the Control key.

To review or define a character sequence for a function key, first press the P key while the terminal is in Setup Mode. Next, press the key (by itself or with the Shift or Control key) whose sequence is to be reviewed or changed. The terminal then displays the characters assigned to that funcion key and prompts the operator to indicate that the assignment is to be changed or is to remain in effect. The operator responds by pressing any key except the Y key if he wants to make no changes. The terminal then restores the main Setup menu. He responds by pressing the Y key if he wishes to make a new assignment. The terminal then accepts up to 4Ø characters entered at the keyboard as the new assignment. When the assignment is complete, the key being defined is pressed. The terminal then stores the new assignment in memory and restores the main Setup menu.

Note: The function key assignment can contain the characters generated by any keys except the function keys themselves and the SETUP key.

Note: To correct an error made while defining a function key assignment, hold down the Command key while striking the Back Space key.

Note: Function Key assignments are not retained from operating session to operating session unless a Save Setup command is issued following their definitions. (See below.)

Note: If the function keys are not defined by the user, or if user definitions are not stored in the permanent Setup Memory, their assignments revert to the default values shown on page 1.16 of this manual.

STORE SETUP

Menu Key: **R**

This option allows you to store the current set up configuration in permanent memory. When this option is selected the current set up features become those which will be enabled subsequently at power-up or in response to an Transparent Mode RIS command.

GRAPHICS SUBMENU

Menu Key: S

When the S key is struck, a new menu appears on the screen. The options represented permit selection of operating conditions of the associated with 4Ø27 Mode and 4Ø1Ø Mode. These options are described in detail on the following pages.

Once the Submenu is on display, the operator may recall the Main Setup Menu by striking the Return key. The Main Menu must be recalled in order to store Setup selections in permanent memory and to take the terminal out of Setup Mode.

An example of the Graphics Submenu appears below.

```
                         DEI GRAPHICS MENU

                         TYPE <CR> TO EXIT

A)    CURRENT MODE                          4Ø27

B)    ENTRY MODE                            4Ø27

C)    TMODE CHAR                            CAN

D)    GRAPHICS INPUT DEVICE                 CROSSHAIR

E)    SCREEN SAVER ENABLED                  YES

F)    GRAPHICS TRAILER1                     CR

G)    GRAPHICS TRAILER2                     NUL

H)    PRINTER OPTIONS                       N/A

I)    TRANSMIT DELAY COUNT                  Ø

J)    ERASE GRAPHICS
```

Items on the Graphics Submenu

The items on the graphics submenu are parameters which define the graphic command processor's mode of operation.

CURRENT MODE

Menu Key: **A**
Settings: TRANS, 4027, 4010

This menu item shows the currently selected on-line operating mode. The operator can use the A key to change the current mode setting. Changing the current mode setting does not change the terminal's entry mode, the on-line mode entered at power-up.

When the A key is struck the the following options are displayed:

A) TRANS B) 4027 C) 4010

The operator uses the A, B or C key to select the desired mode. The selection is then shown to the right of item A on the Graphics Submenu. When the terminal is returned from Setup Mode to on-line operation, the selection goes into effect.

ENTRY MODE

Menu Key: **B**
Settings: TRANS, 4027, 4010

This menu item shows the on-line mode into which the terminal goes following its power-up sequence. The entry mode may be changed by striking the B key and then selecting the desired entry mode from the following list with the A, B or C key:

A) TRANS B) 4027 C) 4010

The selection made is displayed to the right of item B on the Graphics Submenu. After other Graphics Submenu selections have been made and the operator has recalled the Main Setup Menu by striking the carriage return key, a STORE SETUP selection must be made to record the Entry Mode selection in the terminal's permanent Setup memory. Following the issuance of the STORE SETUP command, the operator should cause the terminal to exit Setup Mode by using the (Shift Setup) key combination before powering down.

TMODE CHAR

Menu Key: C
Settings: (see Table 3.3)

The TMode character is the ASCII control character which is used to cause the terminal to enter Transparent Mode from another on-line mode. This character is usually sent by the host, but it may be sent from the keyboard if some sort of echo is enabled.

The operator is permitted to select any ASCII control character as the TMode character. However, many of the control characters are already assigned terminal control functions or 4Ø1Ø mode functions, and these characters should not be used. Normally, the <CAN> code is used.

Selection of a TMode character is made by striking the D key to cause the terminal to display a list of control characters. The control characters are numbered according to their ASCII decimal equivalents. (The list is shown as Table 3.3.) The operator then enters the ADE of the desired character and follows it with a carriage return. The Graphics Submenu reappears showing the selection made.

Table 3.3 - Control Characters with their ADEs

Ø)	NUL
1)	SOH
2)	STX
3)	ETX
4)	EOT
5)	ENQ
6)	ACK
7)	BEL
8)	BS
9)	HT
1Ø)	LF
11)	VT
12)	FF
13)	CR
14)	SO
15)	SI
16)	DLE
17)	DC1
18)	DC2
19)	DC3
2Ø)	DC4
21)	NAK
22)	SYN
23)	ETB
24)	CAN
25)	EM
26)	SUB
27)	ESC
28)	FS
29)	GS
3Ø)	RS
31)	US

GRAPHICS INPUT DEVICE

Menu Key: **D**
Settings: CROSSHAIR, LIGHTPEN

The graphics input device is the means used to position the graphics beam or to send graphics coordinates to a host when the terminal is operating in Graphics Input (GIN) Mode, a submode of 4Ø27 or 4Ø1Ø Mode operation.

If the crosshair cursor is selected as the input device, it is displayed upon entry into GIN mode. If the light pen is selected, it is enabled to operate upon entry into GIN mode. The light pen is an optional accessory to the terminal and should not be selected unless it is actually installed.

The selection of the graphics input device is changed merely by striking the E key. The setting is then switched.

SCREEN SAVER ENABLED

Menu Key: **E**
Settings: YES, NO

The purpose of the screen saver is discussed in the Introduction. This menu item refers to the high resolution screen only. The standard resolution screen saver is enabled and disabled independently with settings made from the Main Setup Menu.

Striking the F key causes the setting to be switched.

GRAPHICS TRAILER1

Menu Key: **F**
Settings: (see Table 3.3)

Trailer codes are ASCII control codes appended to special messages sent by the terminal to the host computer (GIN messages, for example). The terminal responds to certain commands by sending these special messages and then going into a bypass condition, in which it ignores incoming information until a control code is received. Trailer codes are sent at the end of these messages in order to be echoed back to the terminal by the host and cancel the bypass condition. If the trailer codes are terminal control codes (or, in 4Ø1Ø Mode operation, 4Ø1Ø Mode command characters), they are acted upon as such when echoed back.

Two trailer codes must be specified. A <CR><LF> trailer is often used. Menu Key G is used to specify the first trailer code. When this key is struck, a menu of ASCII control codes is displayed. The operator specifies the desired code by typing in its ADE, followed by a carriage return.

GRAPHICS TRAILER2

Menu Key: **G**
Settings: (see Table 3.3)

See the explanation of trailer codes in the description of Menu Key G above. The H key is used to specify the second of two trailer codes appended to special messages to the host.

When the H key is struck a table of ASCII control codes is displayed. The operator specifies the desired code by typing in its ADE, followed by a carriage return.

PRINTER OPTIONS

Menu Key: **H**
Settings: N/A

Printer utilities are not presently included in the graphics processor's program. This item is included on the Graphics Submenu with the idea that it will be ulitized in future versions of the 2427.

TRANSMIT DELAY COUNT

Menu Key: I
Settings: Ø-31

This item refers to a delay between bytes transmitted by the terminal to the host. This delay is independent of baud rate. The delay is set by specifying a number of delay periods (Ø-31). One delay period is the period of one cycle of the terminal's AC input, either 1/6Ø or 1/5Ø of a second.

The operator sets the transmission delay by striking the J key and then typing in the number of delay periods (Ø-31), followed by a carriage return.

ERASE GRAPHICS

Menu Key: **J**

Striking Menu Key K results in the erasure of the contents of the graphics region of the display. The same effect can be had by giving the 4Ø27 Mode command !ERA G<CR>, or by using the (Command - Erase Page) key combination.

Only the high resolution display is affected. Workspace and graphics region definitions remain in effect.

Summary of Terminal Setup Options

Setup Menu	Settable by Host			Normal or Default
Item	4Ø27 Mode	4Ø1Ø Mode	Transparent	Setting
XON/XOFF	no	no	no	YES
Margin Bell	no	no	no	optional
Wrap	no	no	PAWM	YES
New Line	no	no	LMN	YES
CRT Saver	no	no	no	optional
Parity	no	no	no	NO
Cursor Type	no	no	no	optional
Keyclick	no	no	no	optional
Echo	no	no	SRM	NO
Line/Local	no	no	no	LINE
Host Rate	no	no	no	96ØØ
Printer Rate	no	no	no	optional
Color Printer	no	no	FETM	optional
ANSI 3.64	no	no	PATM	YES
Tabs	TAB	no	CTC, TSM	optional
Answerback	no	no	no	optional
Function Keys	no	no	PPFN	optional
Store Setup	no	no	no	(defaults stored)
Graphics Submenu				
Current Mode	yes*	yes*	yes*	optional
Entry Mode	no	no	no	4Ø27
TMode Char	no	no	no	<CAN>
GIN Device	no	no	no	CROSSHAIR
Screen Saver	no	no	no	optional
Trailer 1	no	no	no	<CR>
Trailer 2	no	no	no	<LF>
Printer Optn	(LPR)	no	no	N/A
Trans Delay	no	no	no	Ø
Erase	ERA	no	no	

*See the Section on Terminal Control Codes below.

HOST-TERMINAL COMMUNICATIONS

Software Handshaking

The 2427 processes the incoming data from the host computer through a first in/first out 255 character buffer. If the rate of arriving characters exceeds the rate at which the terminal can process characters, this buffer will begin to fill up. To avoid having the buffer overflow, the terminal has a XON/XOFF signaling capability. If the Auto XON/XOFF feature is enabled, the terminal transmits an XOFF code (<DC3> or 13 hex) to the host computer when the character count of the buffer exceeds 152 characters. In response, the host computer should cease transmitting data until the terminal sends an XON code (<DC1> or 11 Hex). If the host does not follow this convention, the buffer may continue to fill and overflow, which could result in the loss of data. The 2427 processes the characters in the buffer until the character count drops below 76, at which time the terminal transmits the XON code. The use of the XON and XOFF codes ensures that all data are processed by the terminal correctly.

The XON and XOFF codes can also be transmitted from the keyboard. To transmit an XOFF code, press (Control S). To transmit an XON code, press (Control Q). These keys will send the same codes as the convention described above.

The terminal will transmit the XOFF code if any characters are received from the host while the terminal is in the Self-test or Setup Mode. The terminal will send an XON code when it exits either of these modes and is able to process received characters.

The 2427 also recognizes the XON/XOFF conventions when they are sent by the host computer. If the host computer sends an XOFF command, the terminal will cease sending characters to the host computer. .Any keys typed at the keyboard will be stored in the keyboard buffer until it becomes full or until the host computer sends an XON command. If the keyboard buffer becomes full and another key is pressed, the bell will sound.

The 2427 can be used with a host computer which does not support the XON/XOFF conventions. The main requirement is that the baud rate be limited to 4800. It may be desirable to disable XON/OFF signaling when a host application program uses one of these control codes in a special way.

Terminal Control Codes

Several of ASCII control codes (represented by the values between 00 and 1F Hex and the value 7F Hex) are used either by themselves or as command sequence introducers for overall terminal control. These control codes form a group which is distinct from the command sets proper to the terminal's various modes of on-line operation.

When received, these codes cause the terminal to perform some function other than the production of a display element. On-line mode transitions and audible signaling are typical of the functions in question.

These codes can be sent from the keyboard if some type of echo is enabled.

Table 3.3 - Terminal Control Codes and Their Actions

Code	Selectable Recognition	Function
<BEL>		Causes terminal's audible indicator to sound
<BS>		Causes cursor to move one space to the left
<CAN>*		Causes the terminal to enter Transparent Mode when it is in 4010 Mode
<CR>		Causes the cursor to move to the left margin of the current line. If item D "New Line" is selected in Setup Mode, <CR> is interpreted as <CR><LF>.
<DC1>	Yes	Causes the terminal to resume sending to the host (same as XON)
<DC3>	Yes	Causes the terminal to stop sending to the host (same as XOFF)
<ESC> c**		Causes the terminal to be reset.
<ESC>"6g		Causes the terminal to enter 4027 Mode when it is in 4010 Mode. This sequence is normally sent by the host.
<ESC>"7g		Causes the terminal to enter 4010 Mode when it is in 4027 Mode. This sequence is normally sent by the host.
<ESC>"0g		Causes the terminal to enter Transparent Mode when in 4010 or 4027 Mode. This sequence is normally sent by the host.

*This code can be changed by means of selections made in Setup Mode (Graphics Submenu).

**This code is usable with terminals having V2.8 software only. It should not be used with early (V2.7) terminals.

<ENQ>	Yes	Causes the terminal to send its answerback message to the host.
<FF>		Same as <CR>
<GS>		Causes the terminal to enter 4010 Mode when it is in Transparent Mode
<HT>		Causes cursor to move one tab stop to the right
<LF>		Causes the cursor to move down one line
<SI>		Causes the terminal's alternate character set to be used for text display. When the terminal is in 4027 Mode, only the monitor display is affected. (See SCS in Part Six.)
<SO>		Causes the terminal to revert to its normal character set for text display (monitor only in 4027 Mode).
<VT>		Causes the cursor to move up one line

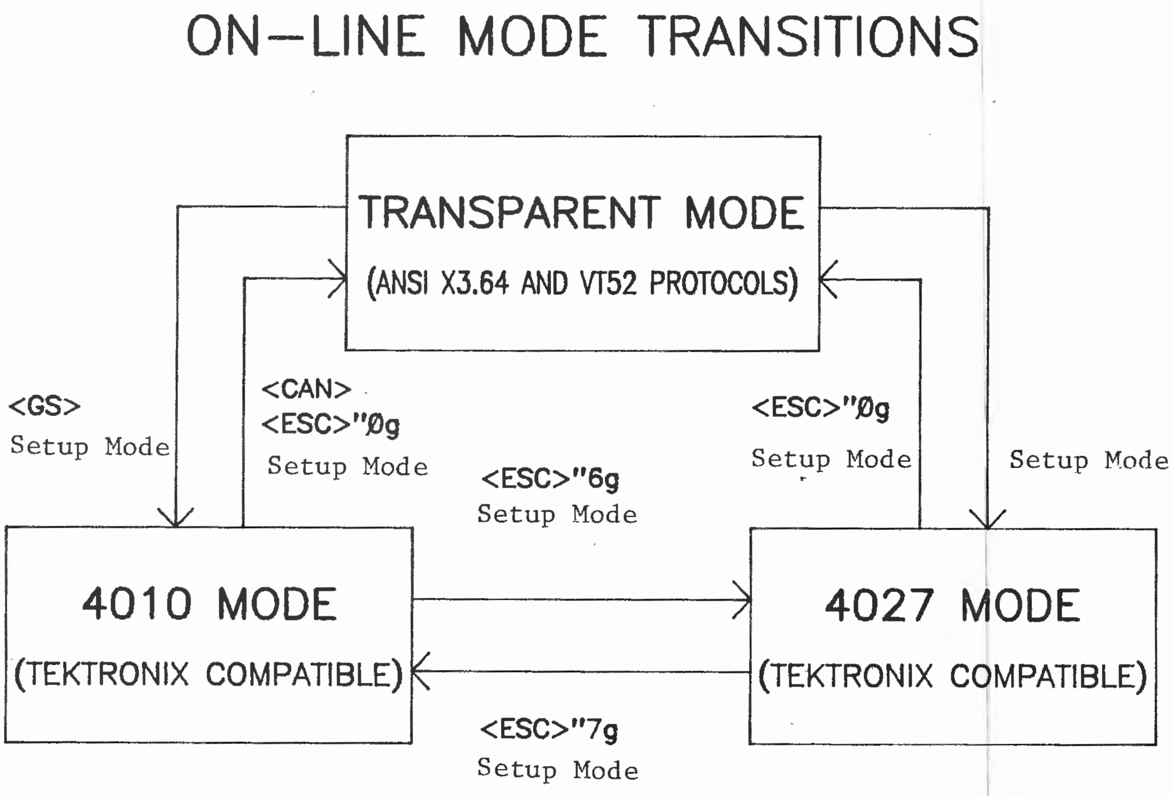
ON-LINE MODE TRANSITIONS

Once the terminal has been set up to communicate with a host, transitions from one on-line mode to another can be made by either of two methods:

- The operator can cause the terminal to enter the Setup Mode, call up the Graphics Submenu and then make a change by selecting item A, using the keyboard.
- The host can make a change by sending the appropriate control code.

Note: At the completion of the power-up sequence, the terminal is automatically placed in the on-line mode specified in Setup (Graphics Submenu). Normally, the 4Ø27 Mode is specified as the entry mode.

The chart below provides a summary of the methods available for switching among the terminal's three on-line modes of operation.



THE TERMINAL'S SELF-TEST CAPABILITIES

In addition to the automatic self-test which is part of the terminal's power-up sequence, there are several tests of terminal operation which can be invoked by commands sent from the host or the keyboard. These tests are described below.

Transparent Mode Confidence Test

The test routines which are automatically invoked at terminal power-up can also be invoked with an ANSI command sequence while the terminal is in Transparent Mode. These tests check the operation of the ANSI/VT52 processor's program memory and I/O devices, the Setup memory, and the standard resolution display memory.

The tests can discover bad memory data and other faults provided the terminal's power supplies, CRT deflection circuits and standard resolution display generator are functioning.

These tests are invoked with the following command sequence (PCFN):

<ESC> [Pn y

"Pn" stands for a decimal integer which can have the following values:

Pn	Test Selected
Ø or 1	ANSI/VT52 processor program memory Setup Memory Standard Resolution Display memory ANSI/VT52 and graphics processor I/O
2	ANSI/VT52 and graphics processor I/O
4	Keyboard/printer I/O

Pn can also be the sum of any of these values, in which case a combination of tests is selected. For example, if it is desired to test both I/O devices, but not the memories, Pn should be 6. When Pn has values from Ø to 7 one test cycle will be performed. If 128 is added to the value of Pn the selected test or tests is repeated until one of three things happens: 1) the Setup key is struck to abort testing, 2) the terminal is powered down, 3) a fault is discovered. If Pn is 255, all tests are repeated in a like manner.

The fact that a test is underway is not indicated by the display unit unless the display memory itself is under test or unless a fault is discovered. If a fault is discovered, a coded message is displayed:

Code	Fault
@	System program memory (UA6 and UA7, main logic board)
A	Setup memory (UA8, main logic board)
C	Display memory (UB4, UB5, UB6 and UB7, main logic board)
G	ANSI/VT52 and graphics processor I/O (UD1, main logic board)
O	Keyboard/printer I/O (UD2, main logic board)

Depending upon how the self-test is invoked and depending upon the results of the test, one of four possible events will occur.

If the test is invoked for one cycle and there are no problems, the test will conclude and the terminal will become active and ready for use.

If there are no hardware problems, but the self-test finds invalid data in the nonvolatile Setup Memory, the terminal will automatically enter Setup Mode and the default settings for all Setup options will be made automatically. In this case the operator should make any changes required in the setup and select the Store Setup option to save settings in the nonvolatile memory. He may then use the Shift and Setup keys to place the terminal on line.

> Note: If the terminal repeatedly enters Setup Mode at power-up, even after a Store Setup command has been given, the Setup memory's battery backup should be checked by a properly qualified technician. For further information, see the 2400 Series Maintenance Manuals.

4027 Mode Confidence Tests

When the terminal is operating in 4027 Mode, there are two tests which can be invoked by commands issued by the host or by the operator from the keyboard.

GTEst Command - Checking the Graphics Character Memories

Syntax: **!GTE**st<**CR**>

When this command is issued the terminal's high resolution screen memory and graphics character memories (ROM and RAM) are tested.

The terminal responds to the command by first displaying a full screen test pattern for a very brief time. This pattern indicates that the screen memory is being checked. The terminal then tests its graphics character memories. No test pattern appears on the screen in this case. Instead, the terminal prints out the test results on the monitor at intervals of a few seconds. Normally, the operator will see the following messages:

```
RAM OK

ROM OK
```

When the test has been completed, the terminal is ready to receive other 4027 Mode commands.

TESt Command - Testing the High Resolution Graphics Display Memory

Syntax: **!TES**t<**CR**>

When this command is issued, a high resolution test display is created on the screen. The test display includes all the standard graphics characters and all the default drawing colors. The test routine automatically creates a workspace and a graphics region and automatically scales coordinates properly. Accordingly, no supplementary commands need be given to make the test display visible, and the test may be run immediately upon entry into 4027 Mode.

When the test display has been completed, the terminal screen may be cleared using the Setup Mode Erase function, the (Command - Erase Page) key combination, or the 4027 Mode Erase Command. Normal 4027 Mode operation may then begin.

PART FOUR

PART FOUR: OPERATION IN THE 4Ø27 MODE

When operating in the 4Ø27 Mode, the 2427 terminal accepts all commands which can be given to the Tektronix 4Ø27 Color Graphics Terminal^{TM}. The 2427 acts on most of these commands and ignores the remainder. It also accepts and acts on some commands which are not part of the Tektronix 4Ø27 command set. All the commands actually acted on by the 2427 while in this mode are described in this part of the manual.

THE 4Ø27 MODE DISPLAY

Workspace and Monitor

In computer work in general, and in computer graphics work in particular, CRT display units are called upon to perform two logically distinct functions. One of these functions is that of a monitor, providing the operator with information about system level operations. The other is that of an output device in system applications, providing the user with solutions to problems (often in the form high resolution graphics displays). These two logical functions are often assigned to separate physical devices, because it is often desirable to monitor system operation while applications programs are running.

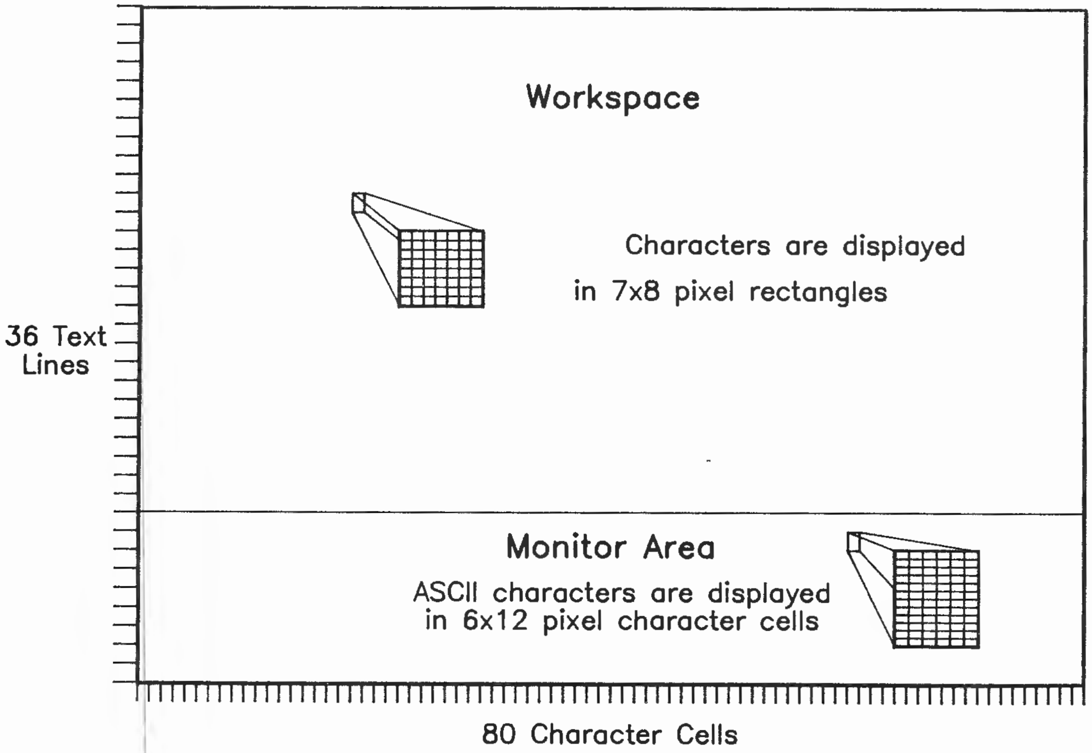
When operating in the 4Ø27 Mode, the 2427 terminal can perform both these logical functions simultaneously. In this mode of operation, the display area of the terminal's screen can be divided into two regions: workspace and monitor area. The monitor area is normally used to display messages to the operator from the host or the terminal's processor, and 4Ø27 Mode commands issued from the keyboard. The workspace is normally used to display text and graphs which are generated in system applications.

Upon entry into 4Ø27 Mode, the the terminal's entire display area is automatically assigned to the monitor and all text is directed to the monitor. Subsequently, workspace may be allocated for applications, and text can be redirected by means of commands issued from the keyboard or the host. (See the Workspace and Monitor Commands.)

Unlike the Tektronix 4Ø27 terminal, the 2427 does not provide for scrolling either within the workspace or within the monitor area. This implies that at any given moment there is no difference between the capacity of the monitor's memory and the capacity of the monitor area and no difference between the capacity of the application memory and the capacity of the workspace.

When an operator uses the terminal in 4Ø27 Mode, he must think of the display area as representing a page of able to hold 36 lines of text, each line having 80 character spaces. Workspace and monitor area are allocated line by line with reference to this scheme.

Display Organization in 4027 Mode



The workspace always begins at the top of the display area and its width is initially defined as the entire width of the display area. Displays within the workspace are generated with high resolution techniques. Text displayed in the workspace is composed of "dot matrix" characters represented within rectangles measuring 7 x 8 pixels. System firmware provides for the generation of two dot matrix character fonts. Other character sets may be created by the user and stored in memory.

Note: The 4027 Mode display is based on composite video signals which are produced by combining the outputs of the standard resolution and high resolution display generators according to an exclusive OR function. Accordingly, if some color other than black is used as a background for standard resolution displays the high resolution display will not be visible. A non-black background cannot be produced once the terminal is in 4027 Mode. However, if a non-black background is defined in Transparent Mode and a switch is made to 4027 Mode without first resetting the background to black, the colored background will be retained and problems will occur when workspace is defined.

The monitor area is always 80 columns wide and at least one line high. Text displayed in this area is composed of characters belonging to the standard resolution display generator's character set. (See Part Five, on the Transparent Mode for details.) While the monitor area is defined in lines relative to a 36 line screen, the actual number of text lines available in this area is given by

$$L_M = (288 - 8L_W) / 12$$

where: L_M is the actual number of lines available when it is rounded to next lowest integer form

L_W is the number of workspace lines (36 minus the nominal number of monitor lines)

The reason for the discrepancy between the nominal number of monitor lines and the real number is that text within the monitor is displayed on 6 x 12 pixel character cells using standard resolution techniques.

Note: If the terminal's special character set is selected while the terminal is in Transparent Mode, and the terminal is then switched to 4027 Mode, the special characters will be displayed in the monitor instead of lower case ASCII characters.

Switching standard resolution character sets is not possible in 4027 Mode. Accordingly, the characters displayable in the 4027 Mode monitor always depend on selections made in Transparent Mode. Characters displayed in the workspace, however, are in no way dependent

Graphics Region

When it is desired to draw graphs or "vector" characters using high resolution techniques, a graphics region is defined relative to the workspace by means of a command issued from the keyboard or host. (See the Graphic Command.) A workspace and a graphics region must be defined before 4027 Mode graphics commands can be executed by the terminal.

The graphics region may coincide with the workspace, may fall wholly within it, or may fall partly within and partly without it. In the last case, a graph drawn in the defined region will be visible only where the region overlies the workspace. (See the Graphic Command for further details.)

The pixels within the active graphics region represent points within an (x,y) coordinate system in which x and y are always integers equal to or greater than zero. Accordingly, the origin of the system is always the point represented by the pixel at the lower left corner of the graphics region. The limits of the system depend on the parameters evaluated in the Graphic Command.

Coordinate Scaling

Once a graphics region has been defined, the terminal can be commanded to draw lines, figures, etc. by supplying absolute (2427 screen) coordinates or coordinates based on the resolution of either the Tektronics 4Ø27 screen or the Tektronics 4Ø1Ø screen. When 4Ø1Ø or 4Ø27 screen coordinates are supplied, they must be transformed to 2427 coordinates before plotting is done. Transforms are accomplished by the graphics processor in response to the Shrink commands.

Graphics Beam Position

Some graphics commands do not specify the position in the graphics region at which a line is to begin, at which a cross-hair cursor is to be placed, etc. Instead, such commands refer the graphics processor to the current "graphics beam position". This position is that at which the last command executed was completed. Immediately following the definition of a graphics region, it is the postion corresponding to the origin of the region's coordinate system.

Screen Saver

The operator should recall that the 2427 has a "screen-saver" feature. Two items on the Setup Menu, described in Part Three, permit independent selection of standard and high resolution screen savers. If a screen saver is selected, a monitor or workspace display will be turned off automatically after about ten minutes of inactivity. The display returns as soon as there is any activity from the host or keyboard.

4Ø27 MODE COMMUNICATIONS PROTOCOL

When operated in the 4Ø27 Mode, the terminal responds to inputs formed in accordance with the Tektronix 4Ø27 protocol.

Information is supplied to the terminal as sequences of characters and/or control symbols, encoded according to ASCII conventions. As these characters and control symbols are received, they are processed either as elements of a text or as elements of a command.

Inputs are accepted from two sources, the host and the keyboard. Text received these sources can be routed either to the monitor or to the workspace. (See the Workspace and Monitor Commands.) Commands are processed according to their contents. Generally, the processing of a command involves a change in one or more of the terminal's operating conditions or the creation of a high resolution graphics display.

Valid commands from the host are not displayed on the screen; commands from the keyboard are displayed in the monitor before execution, which begins with the termination of the command by a carriage return. When commands are being entered from the keyboard, characters typed in error can be deleted with the BACK SPACE key. The entire line can be deleted with a (Control X).

4Ø27 Mode Command Syntax

A 4Ø27 Mode command is distinguished from text by means ASCII codes which mark the beginning and end of the command.

The rules for forming a 4Ø27 command are:

- A command must begin with a command introducer, usually the exclamation point "!", which is the default command introducer.

  A different command introducer can be selected with the **COM**mand Command.

  The character used as the command introducer may not be used as a delimiter or as an element in straight text.

  The character used as the command introducer may be included in a delimited string, in which case it will not function as an introducer.

- The introducer must be followed by a command keyword. Each keyword is an ordinary language name for a function performed in response to a command. The keyword may be spelled out completely in the command or may be abbreviated to its first three letters. Only the first three letters are significant to the machine. The letters of the keyword may be upper or lower case.

  The keyword must follow the command introducer immediately; that is, there must be no spaces or characters between.

- Depending on the keyword in the command, one or more parameters may be evaluated by means of a string of numbers and/or letters following the keyword. Numbers are decimal; letters are upper or lower case.

  When the first parameter value following the keyword is a letter, it must be separated from the keyword by a delimiter.

  When the first parameter value following the keyword is a number, it need not be separated from the keyword by a delimiter.

  When more than one parameter value appears in the command, these values must be separated by delimiters.

  The comma "," or the space " " may be used as a delimiter.

- Following the last parameter value, or (if no values are specified) the keyword, must be a command terminator. The terminator may be a carriage return <CR>, a semicolon ";" or the command introducer for a new command.

  A series of commands may be sent to the terminal as a continuous string, in which each command is terminated by the introducer of the next command and the last command is terminated by a carriage return.

Examples of Valid 4027 Mode Commands

```
                              !BEL<CR>
                              _ _   _
                              | |   |
      Command Introducer _____| |   |
                                |   |
             Keyword ___________|   |
                                    |
               Command Terminator __|
                              !MONitor1Ø H K<CR>
                              _ ___   _ ____  ___
                              | |     | ||||  |
     Command Introducer ______| |     | ||||  |
                                |     | ||||  |
       Keyword _________________|     | ||||  |
                                      | ||||  |
         Numeric Parameter Value ____|  ||||  |
                                        ||||  |
           Delimiter ___________________||||  |
                                         |||  |
             Letter Parameter Value _____|||  |
                                          ||  |
               Delimiter _________________||  |
                                           |  |
                 Letter Parameter Value ___|  |
                                              |
                   Command Terminator ________|
```

```
                              !WORkspace3Ø!GRAphic1,15,1,4Ø<CR>
                              _ ___     ___ ___   _________ __
                              | |        ||  |        |     |
     Command Introducer ______| |        ||  |        |     |
                                |        ||  |        |     |
       Keyword _________________|        ||  |        |     |
                                         ||  |        |     |
         Numeric Parameter Value ________||  |        |     |
                                          |  |        |     |
           Command Introducer ____________|  |        |     |
                                             |        |     |
             Keyword ________________________|        |     |
                                                      |     |
               Numeric String with Delimiters ________|     |
                                                            |
                 Command Terminator ________________________|
```

A Note on Invalid Commands

When the terminal does not recognize a command entered at the keyboard, it displays the WHAT? message; when it does not recognize a command from the host, it displays the invalid command in the monitor. Valid Tektronix 4Ø27 commands not executable by the terminal are accepted without response.

COMMAND GROUP A: DISPLAYING, REPORTING, AND CHANGING STATUS

Many of the settings which can be made with these commands can also be made in the terminal's Setup Mode. The Systat Command provides a convenient means of checking settings while in 4027 Mode. The Commands in this group include the following:

- BEL
- COMmand
- GTEst
- MARgins
- REPort
- SNOopy
- STOps
- SYStat
- TESt

BEL Command - Sounding the Audible Indicator

The host can cause the terminal's audible indicator to sound by sending the ASCII <BEL> control code (hex Ø7) any number of times. The terminal operator can do the same with the (Control G) key combination if local echo is selected. While in 4Ø27 Mode, the terminal will respond to the command described here as well as to the other bell commands.

Syntax: **!BEL<CR>**

Example:

!BEL<CR>	The terminal is commanded to sound its audible indicator once.

COMmand Command - Setting the Command Introducer Character

The terminal is normally set up to treat the ASCII exclamation point, "!", as the 4Ø27 Mode command introducer character. When this setting of the command introducer character is saved in the terminal's permanent (nonvolatile) Setup memory, it becomes the default setting and goes into effect automatically at power-up. Either Setup Mode or the Command Command is used to substitue some other character for the introducer. When the Command Command is used, the setting made is not stored in the permanent Setup memory and is valid only for the current operating session.

The character selected as the introducer may not be used as a delimiter or be included in straight text. It may be used as a text character when sent in a delimited string.

Syntax: **!COM**mand **chr**<CR>

Allowable parameter values:

chr:	any ASCII character =	the new command introducer
	ADE (Ø-127) =	the decimal equivalent for the new command character

Examples:

!COM #<CR>	The terminal is commanded to recognize the character "**#**" as the 4Ø27 command introducer. To change back to "**!**", the command would be **#COM !**<CR>.
!COM34<CR>	The terminal is commanded to recognize the character '**"**' as the 4Ø27 command character. To change back to "**!**", the command would be **"COM !**<CR>.

GTEst Command - Testing the Graphics Character Memories

When this command is issued the terminal's high resolution screen memory and graphics character memories (ROM and RAM) are tested.

The terminal responds to the command by first displaying a full screen test pattern for a brief period. The presence of this pattern indicates that the screen memory test is under way. The character memories are then tested. No display appears while this part of the test is underway. After a few seconds test results are displayed in the monitor.

Syntax: **!GTE**st<**CR**>

Terminal's normal response:

```
(Brief full screen test pattern)

RAM OK

ROM OK
```

When the test has been completed, the terminal is ready to accept 4Ø27 Mode commands.

MARgins Command - Setting the Workspace Left and Right Margins

The Montior Margins are permenantly set at column 1 (left) and column 8Ø (right). Upon entry into 4Ø27 Mode and immediately after the definition of a workspace, the width of the workspace is the width of the display area (8Ø columns). The Margins Command is used to change the width of the workspace on a column by column basis.

Syntax: **!MAR**gins {**left**{**,right**}}<**CR**>

Allowable parameter values:

left:	1-79 =	column at which left margin is set
	none =	left margin at column 1
right:	2-8Ø =	column at which right margin is set
	none =	right margin at column 8Ø

The right margin may not be specified unless the left margin is specified. The right margin column number must always be greater than the left margin column number.

Examples:

!MARgins**11**<**CR**>	The width of the workspace is set to 7Ø columns, with 1Ø columns in the margin to the left. The right marging is at column 8Ø.
!MARgins**3,78**<**CR**>	The width of the workspace is 76 columns, with 2 columns on each margin.

REPort Command

The Report command is normally issued by the host and causes the terminal to send a report of its status to the host in the form of an ANSwer sequence.

Syntax: **!REPort status<CR>**

```
Allowable values for status:  -1 = Firmware version
                              ØØ = Dummy status
                              Ø1 = Text cursor status
                              Ø2 = Graphics beam status
                              Ø3 = Crosshair status
                            Ø4-14 = Dummy status
```

ANSwer Sequences

The terminal responds to a Report command by issuing an Answer sequence to the host. The possible answer sequences are given in the table below. Notice that each sequence is terminated by a semicolon and not by a <CR>.

Command	Response	Meaning of Response
!REP -1<CR>	**!ANS-1,V1.3,INTECOLOR 2427;**	V1.3 is the graphics software version.
!REPØØ<CR>	**!ANSØØ,ØØØØ,ØØØ;**	Dummy Response
!REPØ1<CR>	**!ANSØ1,rrr,ccc, ;**	Text cursor is at row rrr and column ccc, where rrr and ccc are three digit decimal numbers. A space is given where a Tektronix terminal would give the character on display at this position.
!REPØ2<CR>	**!ANSØ2,xxx,yyy,zzzz,ØØ1;**	Graphics beam is at (xxx,yyy), where xxx and yyy are three digit decimal numbers. "zzzz" is the currently selected drawing color register (CØØØ-CØØ7) or pattern number (PØØØ-P119).

!REPØ3<CR>	**!ANSØ3,sss,xxx,yyy;**	The crosshair cursor is at (xxx,yyy), where xxx and yyy are three digit decimal numbers. "sss" is a three digit decimal number representing the status of the crosshair, either ØØØ for off, ØØ1 for on in 4Ø27 Mode, ØØ2 for on in 4Ø1Ø Mode.
!REPØ4<CR>	**!ANSØ4,Ø,ØØ,ØØØØ;**	Dummy Response
|	|	|
|	|	|
!REPØ9<CR>	**!ANSØ9,Ø,ØØ,ØØØØ;**	Dummy Response
!REP1Ø<CR>	**!ANS1Ø,Ø,ØØØØØØ;**	Dummy Response
|	|	|
|	|	|
!REP13<CR>	**!ANS13,Ø,ØØØØØØ;**	Dummy Response
!REP14<CR>	**!ANS14,Ø,F,ØØØ;**	Dummy Response

Note that a cursor or beam position is always reported as an (x,y) coordinate pair in which x and y are three digit decimal integers. The coordinates must be interpreted in accordance with the scale facter in effect when they are sent (Shrink Command).

Coordinate Scaling in Effect at Terminal	Coordinates Sent to Host in ANSwers
4Ø27 to Screen (!SHR N<CR>)	4Ø27
4Ø1Ø to Screen (!SHR Y<CR>)	4Ø27
Screen to Screen (!SHR D<CR>)	Screen

The color and pattern numbering system used in some ANSwer sequences is explained in the descriptions of the COLor and MAP commands.

SNOopy Command - Setting and Disabling Control Code Representation Mode

This command is used to enable and disable the display of control characters and control sequences received from the keyboard and host. When Snoopy Mode is in effect, control codes and control sequences are displayed on the screen as text and are not acted upon in the usual way.

Snoopy Mode displays of control codes are always placed in the 4027 Mode workspace. Therefore, a workspace must be defined and the source of workspace data must be specified before Snoopy Mode is entered. See the Workspace Command in Group B.

Syntax: **!SNOopy {setting}<CR>**

Allowable parameter values:

setting: Y =Enable display of control characters

N = Disable display of control characters

none = Enable display of control characters

Example:

!SNO Y<CR>	The terminal is commanded to display control codes and control sequences. No action is taken in response to receipt of such codes or sequences except that of displaying them.

STOps Command - Setting Horizontal Tabulation Stops in the Workspace

This command is used to specify the columns in the current workspace line to which the cursor is to be moved when the terminal receives the horizontal tabulation code <HT> from a source able to furnish workspace text.

The maximum number of stops which may be set is 8Ø, since there are 8Ø columns per line. A single command may be used to set multiple stops. More than one tab stop is set by including in the command a sequence of column numbers in ascending order (corresponding to the order of stops from left to right).

Syntax: **!STO**ps {**stop**$_1$ ... ,**stop**$_n$}**<CR>**

Allowable parameter values:

stop: 1-8Ø = The column number at which a stop is to be placed

Example:

!STOps5,15,25,35,45,55,65,75**<CR>**Tab stops are set at intervals of ten columns, beginning at column five.

SYStat Command - Requesting a Display of Status Information

This command is issued by the host or the keyboard operator in order to cause a terminal status message to be displayed on the monitor. At least six lines must be allocated to the monitor if the entire status message is to be viewable. (See the MONitor and WORkspace Commands.)

Syntax: **!SYS**tat<**CR**>

The following is a typical status message:

```
TB=4800  RB=4800  LM=1  RM=80  WL=0  V1.1
TS= 10 20 30 40 50 60 70
CC=!  TM=24  SN=N  KB=M  CM=M  PA=E  RS=Y  DU=F
C0 0,100,100    C1 120,50,100   C2 240,50,100   C3 0,50,100
C4 180,50,100   C5 300,50,100   C6 60,50,100    C7 0,0,100
```

The message is interpreted according to the following legend:

TB = Transmit Baud (decimal)
RB = Receive Baud (decimal)
LM = Left Margin (the column number, in decimal)
RM = Right Margin (the column number, in decimal)
WL = Workspace Lines (decimal)
V = Graphics command processor firmware version
TS = Tab Stops (column numbers, in decimal)
CC = 4027 Mode Command Character
TM = Transparent Mode (the code to enter this mode, in ADE)
SN = Snoopy mode (Yes/No)
KB = Keyboard data destination (workspace or monitor)
CM = Computer data destination (workspace or monitor)
PA = Parity setting (Even/Odd/No)
RS = Remote start/stop, same as XON/XOFF (Yes/No)
DU = Duplex/modem control (Full, Half)
C0 - C7 = Color patterns (hue angle, in degrees; percent lightness; percent saturation)

TESt Command - Testing the High Resolution Graphics Display Memory

Syntax: **!TESt<CR>**

When this command is issued, a high resolution test display is created on the screen. The test display includes all the standard graphics characters and all the default drawing colors. The test routine automatically creates a workspace and a graphics region and automatically scales coordinates properly. Accordingly, no supplementary commands need be given to make the test display visible, and the test may be run immediately upon entry into 4Ø27 Mode.

When the test display has been completed, the terminal screen may be cleared using the Setup Mode Erase function, the (Command - Erase Page) key combination, or the 4Ø27 Mode Erase Command. Normal 4Ø27 Mode operation may then begin.

COMMAND GROUP B - PREPARING FOR GRAPHICS AND TEXT

Several parameters must be assigned values before a graphics display can be created on the screen in 4027 Mode. First the workspace and monitor must be defined. Then a graphics region must be created within the workspace. Other definitions and selections that affect the display can also be made. The commands used for these purposes are described on the pages immediately following, and are given roughly in the order in which they are usually used.

The following commands are included in this group:

- WORkspace
- MONitor
- GRAphic
- SHRink
- COLor
- LINe
- DFLine
- ERAse

WORkspace Command

When the terminal is first placed in the 4Ø27 Mode, the entire display area (36 lines) is automatically assigned to the monitor, which displays straight text originating at the host, messages from the terminal processor to the operator, and commands issued to the terminal from the keyboard.

No workspace is available to the graphics generator for executing graphics procedures until it is assigned either with the WORkspace command or the MONitor command. Up to 33 display lines can normally be assigned to the high resolution graphics generator, in one line increments. At least one line is always left to the monitor.

Text can be displayed in the workspace immediately following its definition. Graphs cannot be drawn in the workspace until a graphics region has been defined. The WORkspace Command can be used to specify the source or sources of any straight text which is to go into the workspace.

Syntax: **!WOR**kspace {**lines**} {**source**{**,source**}}<**CR**>

Allowable parameter values:

lines: Ø-33 = number of lines allocated to workspace in normal 4Ø27 Mode operation (With 33 workspace lines there are 2 actual monitor lines.)

34 = 36 lines allocated to workspace (one line of monitor, in workspace area). Allowed when 4Ø1Ø Mode graphics commands (coordinates in ASCII equivalents of binary) are used in 4Ø27 Mode.

source: H = text from host directed into workspace

K = text from keyboard directed into workspace

When only **lines** is evaluated, the command has no effect on the way text is directed into the display, except that if **lines** = Ø no workspace exists and all text is directed into the monitor. Whenever the workspace allocation is changed, anything then on display in the workspace is erased.

When only **source** is evaluated, the command has no effect on the number of display lines allocated to the workspace and no effect on the current display.

If no parameters are evaluated, the effect of the command depends on its point of origin. If the command comes from the host, the terminal responds as if a **!WOR**kspace **H**<**CR**> was sent. If the command comes from the keyboard, the terminal responds as if a **!WOR**kspace **K**<**CR**> was sent.

Examples:

!WORkspace<**CR**> (from host) — Text from the host is directed into the workspace. No change is made in workspace size.

!WORkspace**33**<**CR**> — 33 display lines (the maximum normally possible) are designated workspace. No change is made in the way text is directed.

!WORkspace**3Ø,H**<**CR**> — 3Ø lines are assigned as workspace and text from the host is directed to the workspace.

MONitor Command

The use of the Monitor Command is complementary to that of the Workspace Command. When a number of lines less than 34 is assigned to the Monitor with this command, the remaining lines are assigned as workspace.

When the terminal enters 4Ø27 Mode, text from the host and from the keyboard is directed into the Monitor. If text from one or both of these sources is subsequently directed into the workspace by means of the Workspace Command, it can be redirected to the Monitor by means of the Monitor Command.

There is a discrepancy between the number used in this command to define the size of the monitor and the number of actual text lines allocated with this definition. See the section on the 4Ø27 Display at the beginning of Part Four for an explanation.

Syntax: **!MON**itor {**lines**} {**source**} {**source**}<**CR**>

Allowable parameter values:

lines:	1-34 =	2-24 actual lines assigned to monitor (33-Ø lines assigned to workspace)
	Ø =	1 actual line assigned to monitor (36 lines assigned to workspace), generally used only for 4Ø1Ø commands in 4Ø27 Mode
source:	H =	Host is source of text displayed in monitor
	K =	Keyboard is source of text displayed in monitor

Examples:

!MON6<CR>	Six nominal lines (5 actual lines) are assigned to the monitor and 3Ø lines are assigned for workspace. No change is made in the way text is handled.
!MON K<CR>	No change is made in monitor and workspace line assignments. Text from the keyboard is directed into the monitor.
!MON<CR>	If this command comes from the host, text from the host is directed to the monitor. If it comes from the keyboard, text from the keyboard is directed to monitor.

GRAphic Command - Defining the Graphics Region

The Graphic Command is used to define a currently active graphics region relative to the workspace. The command allows the upper and lower boundaries of the region to be defined in terms of workspace lines and the left and right boundaries to be defined in terms of columns in the display's character cell matrix. Accordingly, the region can always be represented as some rectangular block of character cells. However, the display area within the region resolves beyond character cells to pixels, each character cell encompassing a block of 7 x 8 pixels.

The resolution within the graphics region permits it to represent a coordinate system to which the graphics processor makes reference when it processes commands which include coordinate pairs. The limits of the coordinate system vary with the size of the region defined in the Graphic Command. The lower left corner of the region always represents the point (Ø,Ø) or the origin of the system. The coordinates of the point at the upper right corner of the region can be given in general terms as follows:

$$(x,y) = (7W-1,8H-1)$$

Where: W is the width of the region in character cells
H is the height of the region in character cells (lines)

The coordinates of points in the system represented by the graphics region are absolute coordinates or 2427 screen coordinates. Coordinates sent to the terminal within graphics commands may be absolute coordinates or they may be related to absolute coordinates by a shrink factor. Coordinates based on the resolutions of the Techtronix 401Ø and 4Ø27 terminals are acceptable to the 2427 but are relative and must be transformed into absolute coordinates by the 2427 before they can be properly mapped on the screen. (See the Shrink Command.)

A graphics region must be defined after a workspace has been defined and before the terminal will execute commands to draw graphs. More than one graphics region can be defined within the workspace, but only one region can be active at a given time.

It is possible to define a graphics region which overlaps a previously defined region. In that case, the part of the previously defined region which falls within the new region is erased.

It is also possible to define a region which extends beyond the upper limit of the workspace. In that case, only that part of the region which falls within the workspace will be visible.

Syntax: **!GRAphic row**$_b$**,row**$_e${**,col**$_b${**,col**$_e$}}<CR>

Allowable parameter values:

row$_b$: positive integer < **row**$_e$ = cell row or line with which region begins

1 = cell row or line with which region begins, for 4Ø1Ø type graphics commands in 4Ø27 Mode

row$_e$: positive integer < **row**$_b$+53 = cell row or line with which the region ends

35 = cell row or line with which region ends, for 4Ø1Ø type graphics commands in 4Ø27 Mode

col$_b$: integers 1-79 < **col**$_e$ = cell column with which region begins

1 = default value

col$_e$: integers 2-8Ø = cell row with which region ends

8Ø = default value

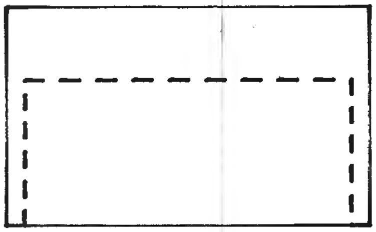
The beginning and ending rows must always be specified. If columns are not specified, their default values are assumed. Beginning column may be specified without specifying end column, in which case end column has the default value.

The point at the upper right corner of the coordinate system defined with the Graphic Command can be determined from the row and column numbers used in the command by the formula:

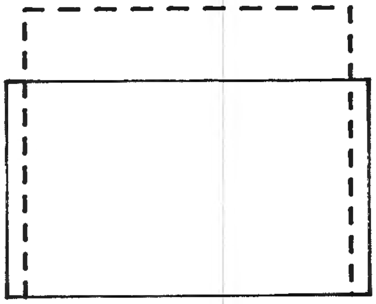
$$[x,y] = [7(col_e-col_b+1)-1,8(row_e-row_b+1)-1]$$

The position of the graphics region as a whole, relative to the workspace is fixed by the following rules (L_W = workspace lines):

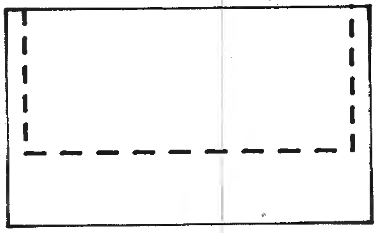
- If $(row_e-row_b)<L_W$ and $row_e<=L_W$ then the bottom of the region is at the bottom of the workspace and the entire region is visible.



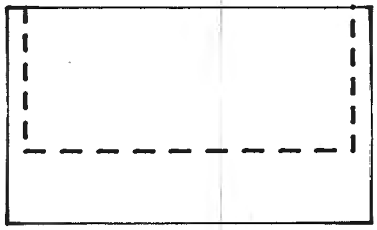
- If $(row_e-row_b)>L_W$ then the bottom of the the region is at the bottom of the workspace and the top of the region is not visible.



- If $(row_e-row_b)<L_W$ and $row_b>L_W$ then the top of the region is at the top of the workspace and the entire region is visible.



- If $(row_e-row_b)<L_W$ and $row_b<L_W$ but $row_e>L_W$ then the top of the region is at the top of the workspace and the entire region is visible.



Examples:

!GRAphic1,3Ø<CR>

Vertically, the graphics region begins with line 1 and ends with line 3Ø of the workspace. It covers the entire width of the workspace (8Ø columns). The limits of the coordinate system, given as absolute or pixel coordinate pairs are (Ø,Ø) and (559,239).

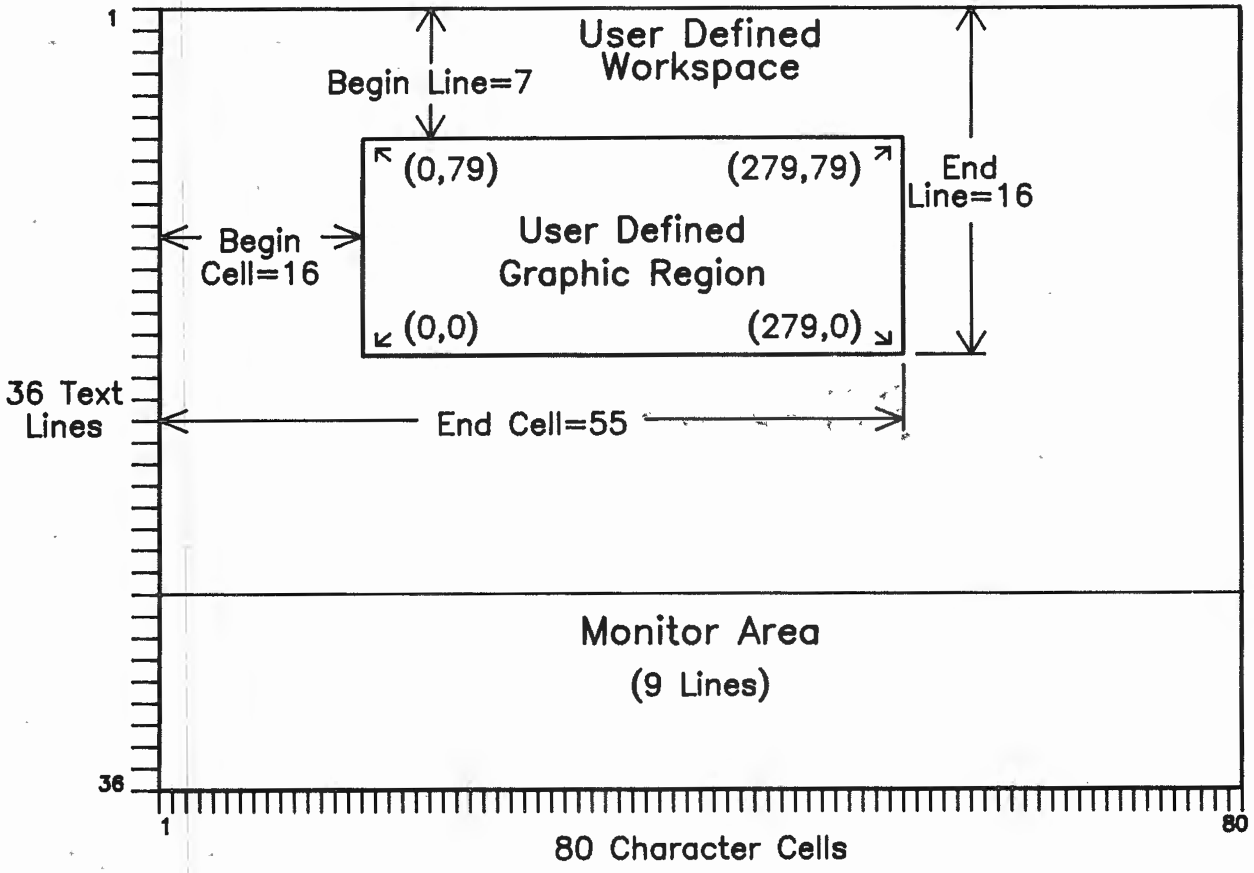
!GRAphic5,2Ø,1Ø<CR>

Vertically, the graphics region begins with line 5 and ends with line 2Ø of the workspace. Horizontally, it begins with the 1Øth character cell and ends with the 8Øth cell. The limits of the coordinate system, given as absolute or pixel coordinate pairs, are (Ø,Ø) and (496,127).

!GRAphic7,16,16,55<CR>

Vertically, the graphics region begins with line 7 and ends with line 16 of the workspace. Horizontally, it begins with cell 16 and ends with cell 55. The absolute coordinate pairs defining the limits of the coordinate system are (Ø,Ø) and (279,79).

See the illustration of the region defined by this command.



SHRink Command - Coordinate Scaling

In 4Ø27 Mode the terminal can be commanded to draw graphs using coordinates based on the resolution of any one of three screens:

- The Tektronix 4Ø27 screen, 64Ø pixels x 476 pixels
- The Tektronix 4Ø1Ø screen, 1Ø24 pixels x 784 pixels
- The 2427 screen itself, 56Ø pixels x 288 pixels

When the coordinates sent to the terminal are based on the resolution of either the Tektronix 4Ø27 or the Tektronix 4Ø1Ø screen, these coordinates must be transformed into 2427 screen coordinates by the terminal's processor before they are used in plotting points, lines or figures. The Shrink command is used to instruct the processor to perform the necessary transformations. When the coordinates sent to the terminal are absolute (2427 screen) coordinates, the Shrink Command is used to set the scaling factor to 1.

The scaling factors used to transform coordinates are given in the table below. These factors themselves are not used to evaluate the factor parameter in the Shrink Command and are provided for reference only.

Conversion of Coordinates*

From	To	Multiply x by	Multiply y by
4Ø27	2427	Ø.875	Ø.571
4Ø27	4Ø27	1.ØØØ	1.ØØØ
4Ø27	4Ø1Ø	1.75Ø	1.571
4Ø1Ø	2427	Ø.5ØØ	Ø.364
4Ø1Ø	4Ø27	Ø.571	Ø.636
4Ø1Ø	4Ø1Ø	1.ØØØ	1.ØØØ
2427	2427	1.ØØØ	1.ØØØ
2427	4Ø27	1.143	1.75Ø
2427	4Ø1Ø	2.ØØØ	2.75Ø

*Coordinates are assumed given as decimal integers; results of transformations must be rounded to integer form.

Note that the shrink factors are not simply the ratios of the total horizontal and vertical resolutions of the various screens. The use of the Shrink Command supposes the definition of a graphics region in terms of character cells. Accordingly, the shrink factors are ratios of the dimensions, in pixels, of character cells. The 2427 character cell is 7 x 8 pixels (in 4027 Mode); the 4027 character cell is 8 x 14 pixels; the 4010 character cell equivalent is 14 x 22 pixels.

When the terminal is first placed in 4027 Mode it is automatically set up to scale 4027 screen coordinates to 2427 screen coordinates. Accordingly, the Shrink Command need not be given unless the user desires to change the shrink factor. When the command is given without a parameter value, the terminal assumes the value Y.

Syntax: **!SHR**ink {**factor**}<**CR**>

Allowable parameter values:

factor: N =	4027 coord. to 2427 coord. (default)
Y =	4010 coord. to 2427 coord.
H =	not used
B =	not used
D =	2427 coord. to 2427 coord.
none =	4010 coord. to 2427 coord.

Example: **!SHR**ink **N<CR>**

Suppose that the 2427 is operating in the 4027 Mode and the command **!GRA**phic**7,17,16,56<CR>** has been given to define a graphics region. This region represents a coordinate system defined by the points (0,0) and (279,79). See the examples of the Graphic Command for an illustration.

If the same command were sent to a Tektronics 4027 terminal, the coordinate system represented would be defined by the points (0,0) and (319,139). Accordingly, the point (300,100) would be representable on the 4027 screen.

Suppose the coordinate pair (300,100) is sent to the 2427. Unless the coordinates are scaled for the 2427 screen, the point is not representable, for it falls beyond the limit of the 2427's system, defined by (279,79).

If a **!SHR**ink **N<CR>** has already been received by the 2427 when the coordinates (300,100) are received, these coordinates will be scaled to (263,57) by the terminal and will be represented on the screen.

LINe Command - Selecting the Line Type

When the terminal enters 4Ø27 Mode the type of line used for vectors is the "normal" solid line. Other line types can be selected with the Line Command.

Syntax: **!LINe** {**type**}<CR>

Allowable parameter values:

type: 1-8 = one of eight standard line types

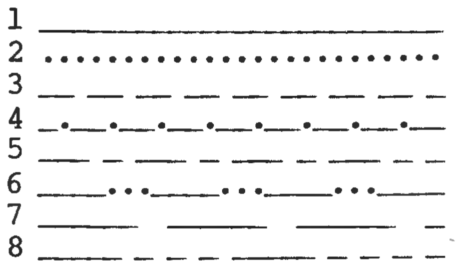
9 = line type defined with DFLine

P = endpoint of line only is plotted

E = erasing line

none = line type 1 (default)

Standard Line Types



Examples:

!LIN2<CR> The dotted line is selected for vectors drawn in response to subsequent commands.

!LIN P<CR> When a vector command is given subsequently, only the end point of the vector is plotted.

!LIN E<CR> When a command has been given to create a graphic element (line, circle, etc.) and the element created is to be erased, without disturbing other elements on display, this command is issued, followed by a repetition of the original command. After erasure the line type should be reset.

DFLine Command - Defining Line Type 9

If a line type other than one of the standard types (1-8) is needed, it can be defined using this command and then selected with the command **!LIN9**<**CR**>.

Line 9 is defined as a pattern according to which pixels lying on a line are to be turned "on" and "off". The DFLine Command provides for a pattern having up to eight segments. In the successive segments, pixels are alternately "on" and "off". Each segment may contain from 1 to 63 pixels.

Syntax: **!DFL**ine Pon$_1$**,**Poff$_2${**,**Pon$_3$**,**Poff$_4$ ... **,**Pon$_7$**,**Poff$_8$}<**CR**>

Allowable parameter values:

Pon: 1-63 = number of pixels "on"

Poff: 1-63 = number of pixels "off"

Examples:

!DFL2Ø,1Ø<**CR**>

This command defines a broken line pattern in which there are 2Ø pixels "on" followed by 1Ø pixels "off". In long lines, this pattern is repeated.

!DFL3Ø,1Ø,2Ø,5<**CR**>

This command defines a more complex line pattern, in which there are 3Ø pixels "on", then 1Ø "off, then 2Ø "on, then 5 "off".

COLor Command

Upon entry into 4Ø27 Mode, the color registers in the high resolution display generator (with addresses CØ-C7) are loaded with default color codes, i. e., the codes for the eight basic displayable colors. At the same time, the three-bit code selected for use in the high resolution screen memory when graphics commands are executed is the code which points to color register CØ. Since CØ is loaded with the color code for white, white is the color used to draw graphic elements.

Any of the other seven default colors can be selected for graphics by instructing the graphics processor to load the appropriate pointers (color register addresses) into the screen memory. The Color Command is used for this purpose.

When the contents of the color registers have been changed and no longer hold default color codes, the selection of registers CØ-C7 with the Color Command results in the display of the other colors. (See the description of the MAP command in Command Group D.)

The Color Command permits the selection one color for vectors and polygon fills and a second color for polygon outlines or boundaries. When a boundary color is not specified, all graphics elements drawn subsequent to the command are represented with the one color specified.

While the Color Command can be used simply to select color registers, it can also be used to select color patterns which have already been defined by the user with the Pattern Command. A color pattern is built up using some combination of color registers. Up to 12Ø patterns may be defined and stored in memory at any one time. (The Pattern Command is described fully, along with other related commands, under Command Group D.)

Syntax: **!COLor** $\mathbf{Cn_v}$\{**,** $\mathbf{Cn_b}$\}<**CR**>

Allowable parameter values:

$\mathbf{Cn_v}$: CØ-C7 = one of the eight color registers. The content of the register selected represents the color to be used for drawing vectors and for filling polygons. This color is also used to outline filled polygons, unless a boundary color register is specified.

PØ-P119 = one of 12Ø possible user-defined fill patterns. When a pattern is specified instead of a color register that pattern is used for fills and the default color register is used for vectors and polygon boundaries.

Cn$_b$: CØ-C7 = one of the eight color registers. The content of the register selected represents the color to be used to outline a filled polygon (the boundary color).

Default Assignments to Color Registers

CØ - white
C1 - red
C2 - green
C3 - blue

C4 - yellow
C5 - cyan
C6 - magenta
C7 - black

Examples:

!COLor C6,C1<CR> This command causes the color whose code is stored in color register C6 (default color is magenta) to be used for lines and polygon fills. C1 (default red) is used for polygon outlines.

!COLor P18,C1<CR> This command specifies that pattern P18 (which must have been defined by the user) is to be used for polygon fills and that the color defined by the content of register C1 (default red) is to be used to outline polygons.

!COLor C2<CR> This command specifies that register C2 (default content green) is to be used for lines, polygon fills and polygon outlines.

!COLor P18<CR> This command specifies that pattern P18 is to be used for polygon fills. Lines and polygon outlines are to be done with CØ (default white).

ERAse Command - Erasing Text and Graphics

The Erase Command is used to clear the monitor, the workspace or the graphics region of the workspace, depending on the evaluation of the area parameter in the command.

> Note: Complement Mode should be disabled before an Erase Command is issued.
>
> The entire screen may be cleared by the operator in either of two ways. He may strike the (Command - Erase Page) key combination, or he may cause the terminal to go into Setup Mode, select Erase on the Graphics Submenu, and then cause a return to 4Ø27 Mode. (See Part Three for details about Setup Mode.) In either case, the sizes of the workspace and monitor remain unchanged. If the terminal has been in Graphics Input (GIN) Mode, this mode is cancelled and trailer codes are sent to the host.

Syntax: **!ERA**se {**area**{**,background**$_{gr}$}}<**CR**>

Allowable parameter values:

area:

W = Workspace (The size of the workspace remains unchanged, but any previous definition of a graphics region is cancelled.)

M = Monitor (The size of the monitor remains unchanged.)

G = Graphics region (The size of the region remains unchanged. The entire region goes to black unless a fill pattern or background color is specified.)

background$_{gr}$**:**

CØ-C7 = One of the eight color registers. The color whose code is held in the specified register is used to fill the graphics region when this region is erased.

PØ-P119 = One of up to 12Ø fill patterns previously defined by the user. The pattern specified is used to fill the graphics region when this region is erased. (See the PATtern Command.)

A background color or fill pattern may not be specified for the graphics region unless **G** is the argument for **area.**

If no parameter value is included in the command, the source of the command and the routing of text, as specified with a previous Monitor or Workspace Command, determine the area of the display erased. If the command comes from the host, that area to which host text is routed is erased. If the command comes from the keyboard, that area to which keyboard text is routed is erased.

Examples:

!ERAse G<CR> The content of the entire currently active graphics region is erased to black. The region remains defined, and anything on display outside the region is not affected.

!ERAse G P6<CR> The content of the currently active graphics region is erased and the region is filled with pattern P6, which must have been defined previously.

!ERAse<CR> If the command comes from the host, and if text from the host is routed to the workspace, the workspace is erased and any definition of a graphics region there is canceled.

If the command comes from the keyboard, and if text from the keyboard is routed to the monitor, the monitor is erased.

Note on Means of Erasing Individual Display Elements

The Erase Command is used to erase the content of an entire region of the screen. Some of the graphics and color commands, which are discussed fully in the following sections of the manual, permit the erasure of individual elements of the high resolution display.

Individual graphics elements can be erased by redrawing them in the background color or with line type "E". (See the DFLine Command.) It should be noted that this method of erasure can produce breaks where the line being erased crosses another line.

Text within the workspace can be erased by re-entering the characters in the background color. For example, if text is displayed in white characters following the issuance of the command **!ATTribute CØ<CR>**, the text can be erased by issuing the command **!ATTribute C7<CR>** and then re-entering the characters in whatever manner they were originally entered (directly or with the !STRing command).

Text can also be erased without changing colors, by writing over it with spaces. However, if the background color of the text is not the same as the color of the area in which text is written the spaces will show up against the color of the surrounding area. For example, if text is displayed in white characters on a black background within a polygon which is filled with white, writing over the characters with spaces will leave a black "window" within the polygon.

In addition, the use of spaces to erase text prevents the drawing of graphics elements in the area occupied by these spaces, even when the color of the spaces and the color of the surrounding area are the same and the spaces are not distinguishable.

COMMAND GROUP C: GRAPHICS COMMANDS

Following the definition of a workspace and a graphics region and the setting of a coordinate scaling factor (if other than 4Ø27 to 2427), the graphics commands belonging to this group can be sent to the terminal. These commands include:

The Basic Graphics Commands

- VECtor
- RVEctor
- CIRcle
- PIE
- POLygon
- RPOlygon

The GIN (Graphics Input) Mode Commands

- ENAble
- INPut
- DISable

Using 4Ø1Ø Type Graphics Commands in 4Ø27 Mode

Each command in this group is described in detail in the pages immediately following. Examples used assume a workspace of 33 lines, a graphics region of 33 lines the full width of the workspace, and a coordinate shrink factor of 1. These conditions would be established with the following commands: **!WOR33!GRA1,33,!SHR D<CR>**

The Basic Graphics Commands

VECtor Command - Moving the Graphics Beam Position and Drawing Lines

The Vector command causes a line or series of connected lines to be drawn in the graphics region. The currently selected color and line type are used. (If the current line type is "P", only end points will be plotted.)

A beginning point and any number of end points are sepecified in the command as (x,y) coordinates relative to the origin of a previously defined system. If an end point specified in a vector command falls outside the visible graphics region, the vector is "clipped" at the edge of the region.

If a single pair of coordinates is given in the command, the graphics beam position is changed to the point indicated and nothing is drawn.

Syntax: **!VEC**tor **xb**,**yb**{,**xe**$_1$,**ye**$_1${,**xe**$_2$,**ye**$_2$ ... ,**xe**$_n$,**ye**$_n$}}<**CR**>

Where: **xb** = the x coord. of the beginning point of a vector or of the point to which the graphics beam is to be moved (positive decimal integer)

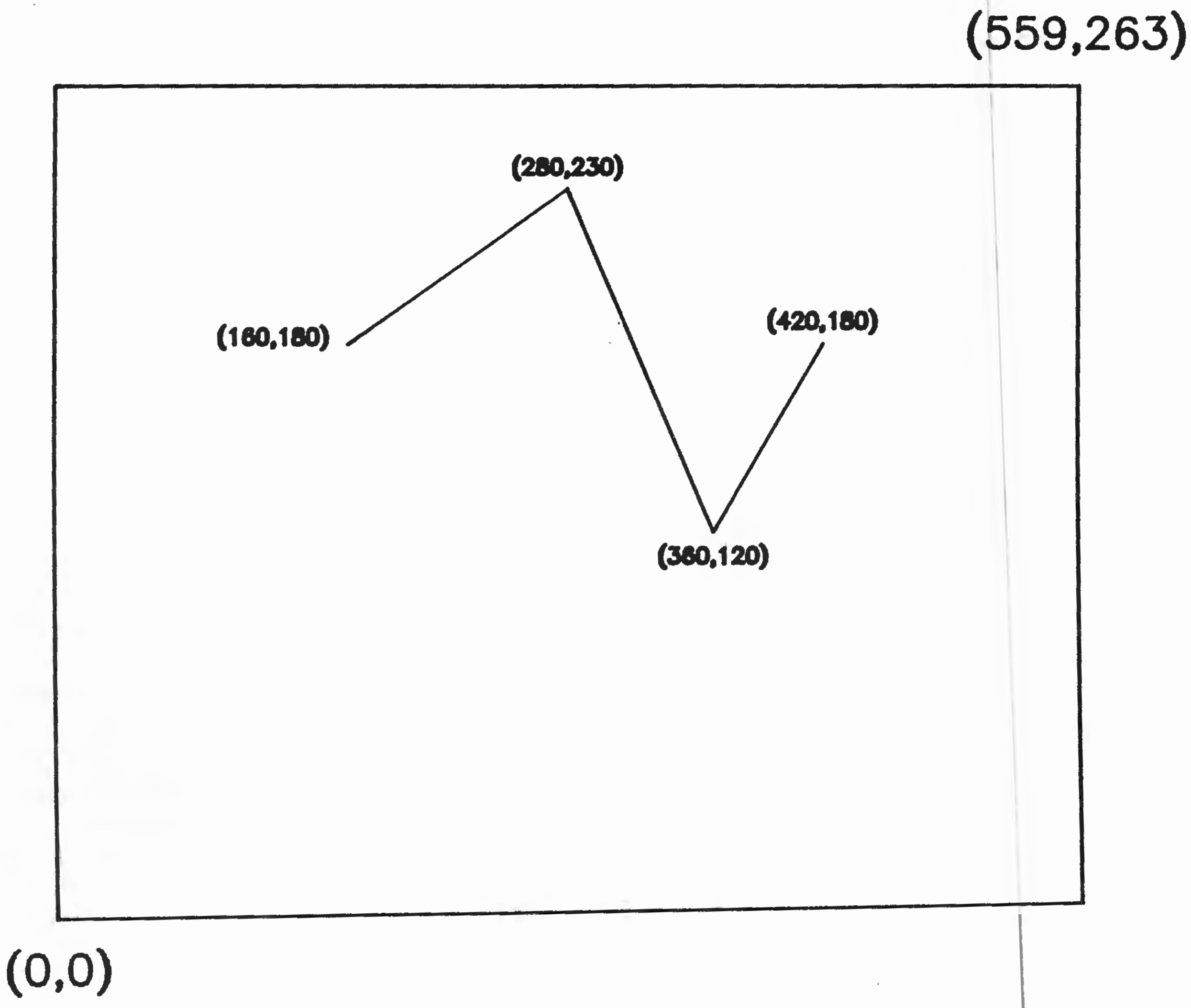
yb = the y coord. of the beginning point of a vector or of the point to which the graphics beam is to be moved (positive decimal integer)

xe = the x coord. of an end point (positive decimal integer)

ye = the y coord. of an end point (positive decimal integer)

Example:

A graphics region is defined using the commands **!WOR33!GRA1,33!SHRD<CR>**. A series of connected vectors are then drawn in the region.



```
!VEC 160,180,280,230,360,120,420,180<CR>
```

RVEctor Command - Relative Moves and Relative Line Drawing

The Relative Vector command causes the same actions as does the Vector command, but each pair of coordinates is relative to the current graphics beam position. Relative coordinates can be positive or negative.

When only one pair of coordinates is specified, the command causes the graphics beam to be moved without drawing.

Syntax: **!RVE**ctor **dx**b,**dy**b{,**dx**e$_1$,**dy**e$_1${,**dx**e$_2$,**dy**e$_2$ ...,**dx**e$_n$,**dy**e$_n$}}**<CR>**

Where: **dx**b = delta x from current beam position to beginning point (positive or negative decimal integer)

dyb = delta y from current beam position to beginning point (positive or negative decimal integer)

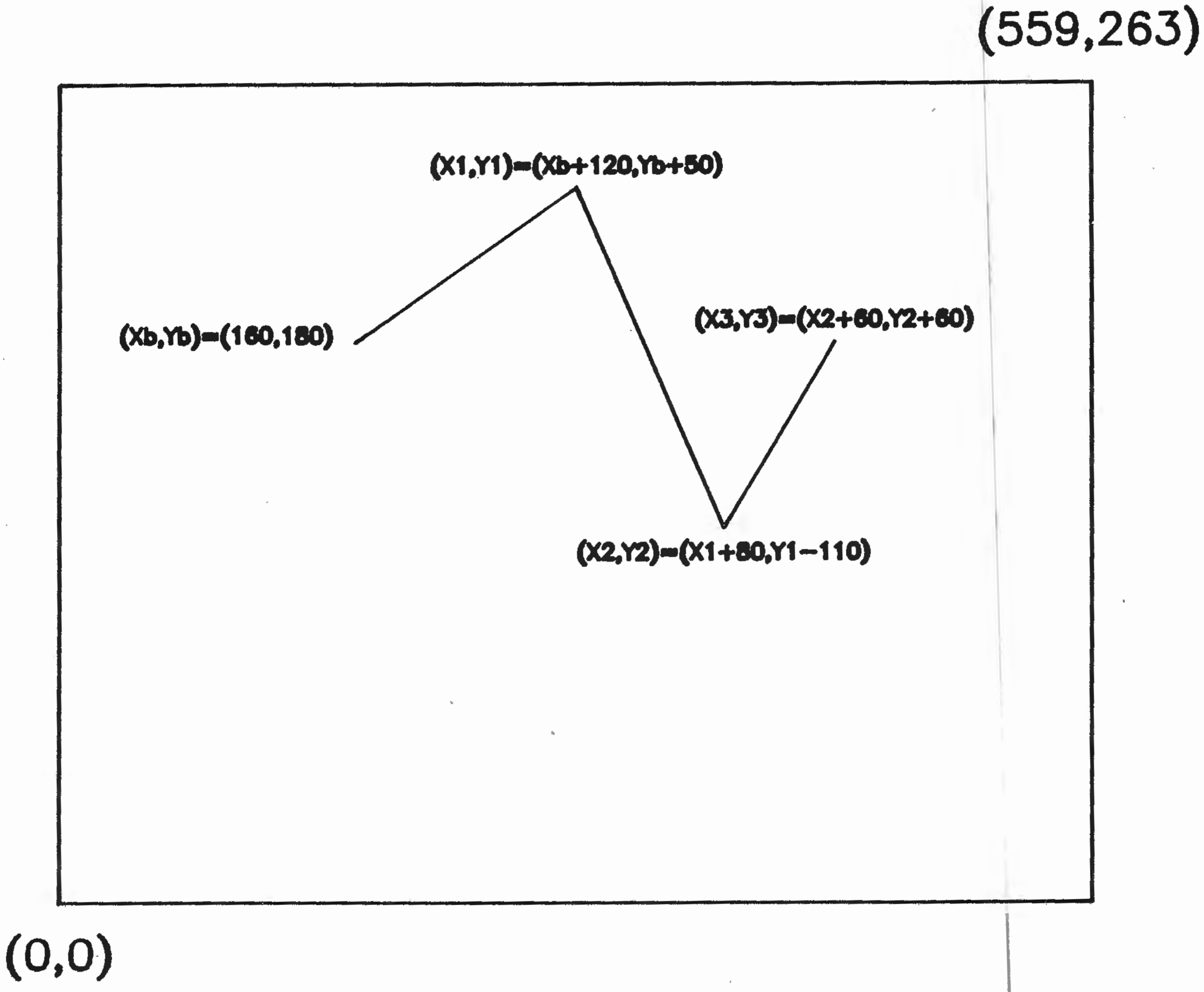
dxe = delta x from beginning point or from last end point to end point (positive or negative decimal integer)

dye = delta y from beginning point or from last end point to end point (positive or negative decimal integer)

Note: With early versions of terminal firmware, drawing a series of relative vectors which involve coordinate scaling can result in cumulative scaling errors, especially if many short vectors are drawn.

Example:

A graphics region is defined according to the commands **!WOR33!GRA1,33!SHR D<CR>.** The graphics beam is moved to (Ø,Ø) with a Vector command to establish a convenient initial reference point. A series of connected vectors is then drawn, using relative coordinates.



```
!VEC 0,0<CR>
!RVE 160,180,120,50,80,-110,60,60<CR>
```

CIRcle Command - Drawing Unfilled Circles, Arcs and Regular Polygons

The Circle Command causes the drawing of an unfilled arc, circle or regular polygon around the current graphics beam position. The currently selected color and line type are used.

To draw a circle, only the radius need be specified in the command. An arc is drawn by specifying the radius, a start angle and a stop angle. The angles are measured in degrees moving counter-clockwise from an imaginary line which extends from the beam position (center of the arc) horizontally to the right.

Arcs and circles are not perfectly curvilinear but are composed of very short straight line segments. The length of these segments can be changed by specifing the angle across which the segments are drawn. When the segment length is increased in this way, the figure drawn is changed from an arc or circle to a regular polygon or regular polygon segment. The number of sides in the polygon varies with the segment angle.

Syntax: **!CIRcle rad**{**,start**{**,end**{**,seg**}}}<CR>

Allowable parameter values:

rad: positive integer = radius

start: Ø-36Ø = angle in degrees at which an arc is to begin (counter-clockwise from a horizontal line extending to the right from the beam position (center)

Ø = default (no start angle specified)

end: Ø-36Ø = angle in degrees at which an arc is to end (counter-clockwise from a horizontal line extending to the right from the beam position (center)

36Ø = default (no end angle specified)

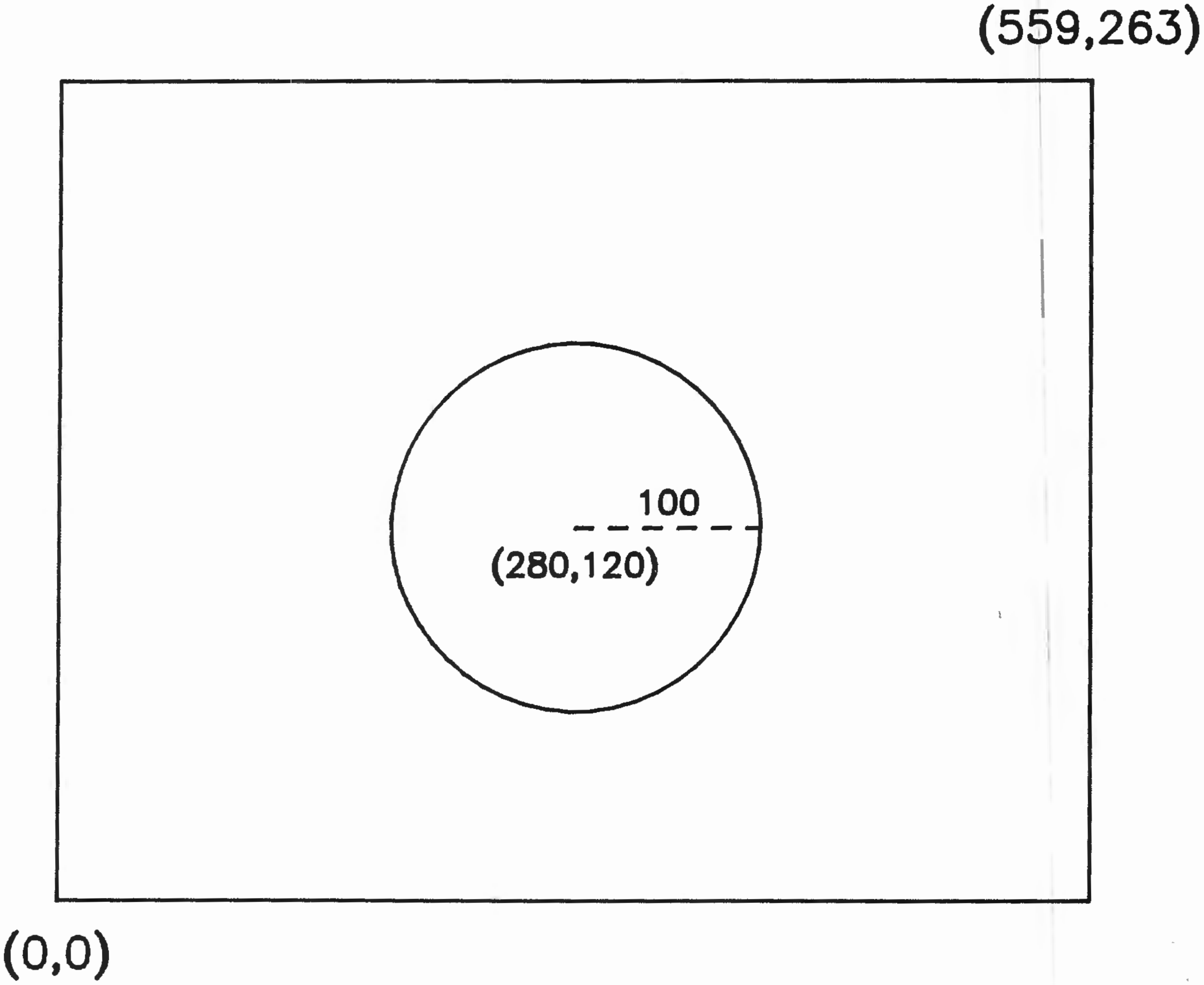
seg: Ø-18Ø = angle in degrees across which segments are drawn

4 = default (minimum angle, used for circles and arcs)

An end angle cannot be specified unless a start angle is specified. A segment angle cannot be specified unless both a start angle and an end angle are specified. A complete regular polygon is drawn when the start and end angles are Ø° and 36Ø° respectively.

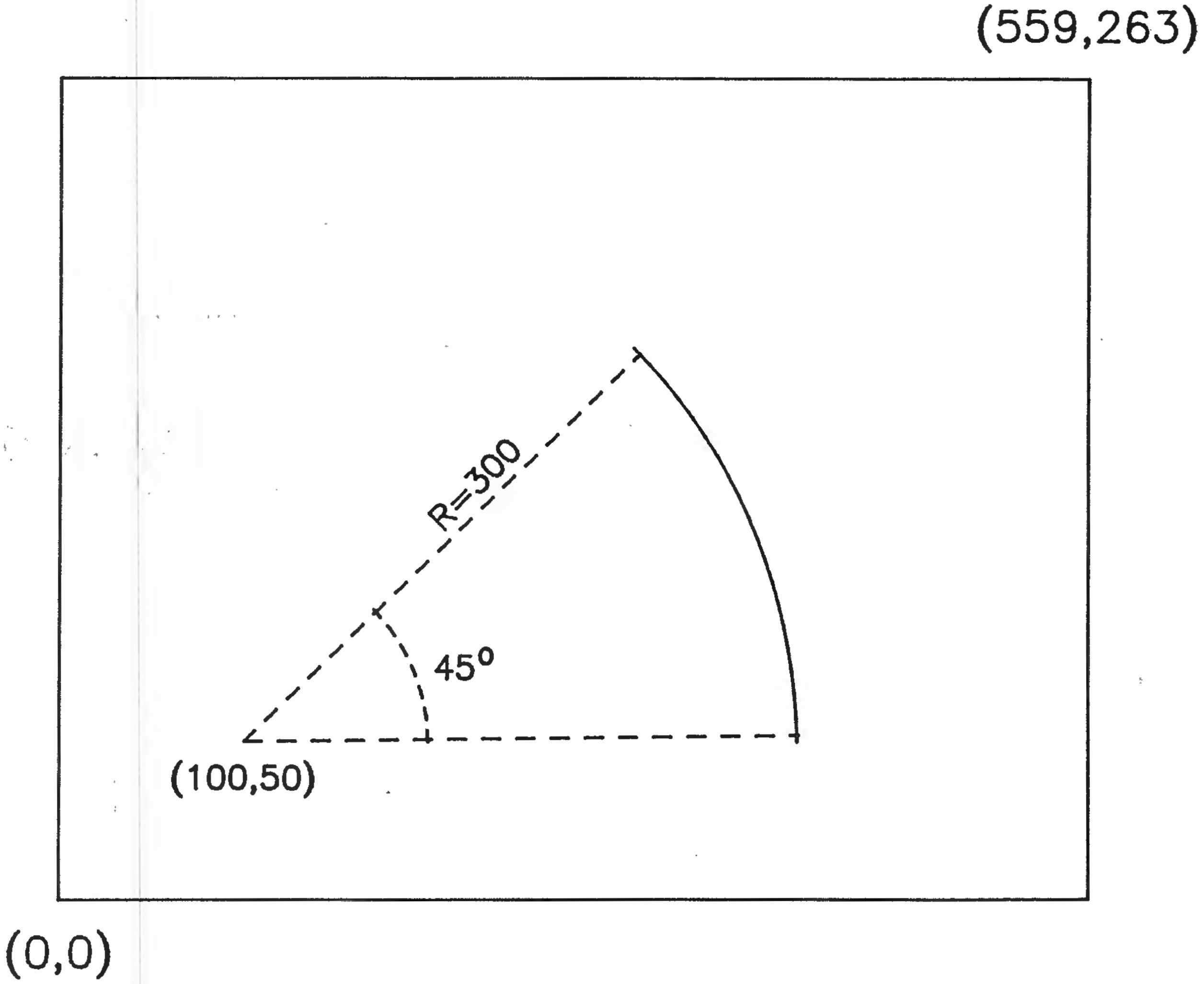
Examples:

The graphics beam is first moved to the point (28Ø,12Ø) and then a circle with radius 1ØØ is drawn around this point.



```
!VEC 280 120<CR>
!CIR 100<CR>
```

The graphics beam is first moved to (1ØØ,5Ø) and then an arc with radius 3ØØ is drawn about this point from Ø° to 45°.



```
!VEC 100,50<CR>
!CIR 300,0,45<CR>
```

PIE Command - Drawing a Filled Arc, Circle or Regular Polygon

The Pie Command has the same effect as the Circle Command, except that the figure drawn is filled with the currently selected fill color or fill pattern.

The center of the "pie" or "pie slice" which is drawn is the current graphics beam position.

Syntax: **!PIE rad**{**,start**{**,end**{**,seg**}}}<**CR**>

Allowable parameter values:

rad: positive decimal integer = the radius of the figure

start: Ø-36Ø = angle in degrees at which "pie slice" starts (counter-clockwise from horizontal line extending to right of beam position)

Ø = default (no start angle specified)

end: Ø-36Ø = angle in degrees at which "pie slice" ends

36Ø = default (no end angle specified)

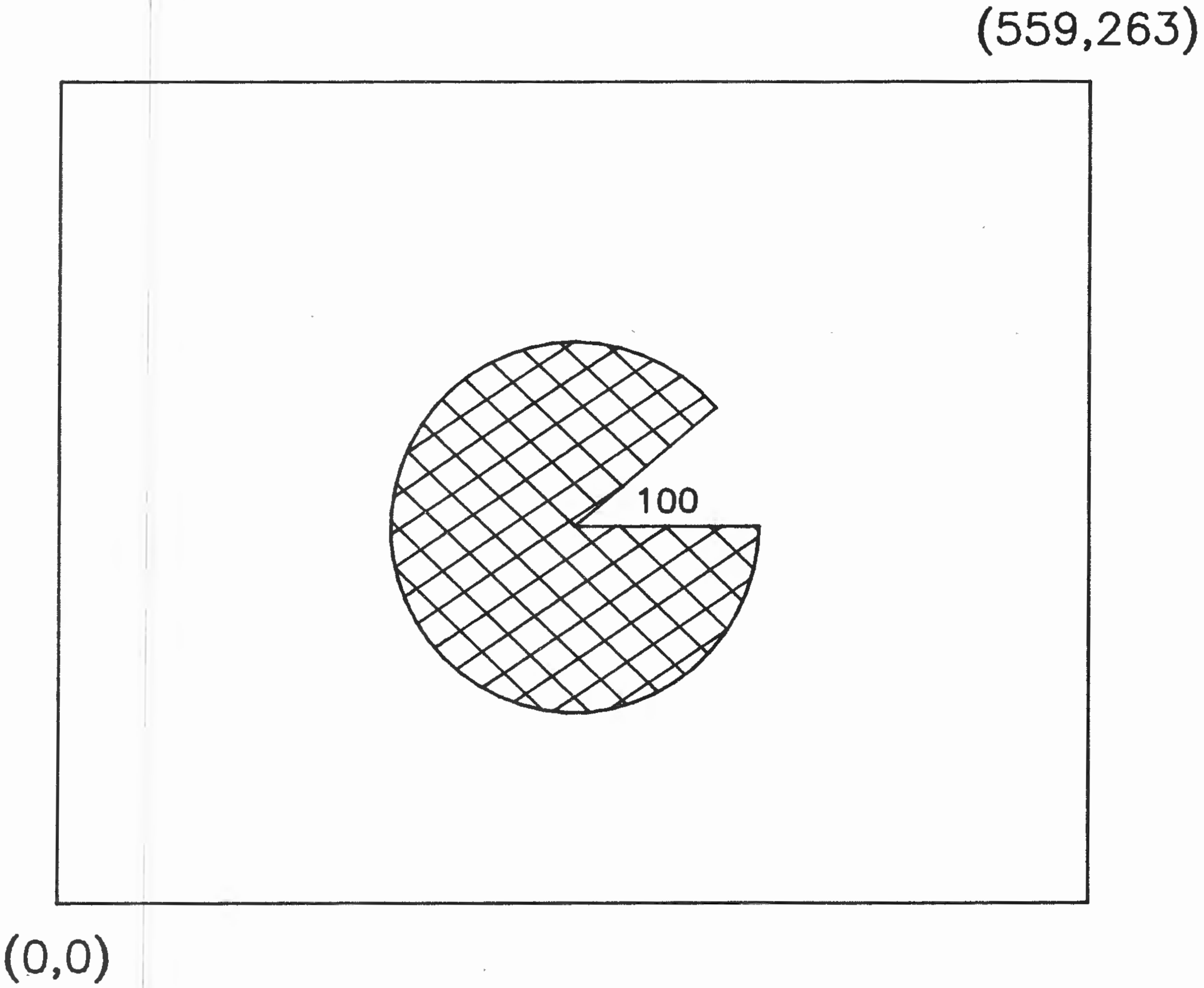
seg: Ø-18Ø = angle in degrees across which a boundary line segment is drawn

4 = default (minimum angle, for arcs)

A start angle must be specified if an end angle is specified. Both start and end angles must be specified if a segment angle is specified.

Example:

The graphics beam is first moved to the point (28Ø,12Ø) and a pie with radius 1ØØ is then drawn about this point from 4Ø° to 36Ø°.



```
!VEC 280 120<CR>
!PIE 100,40 0<CR>
```

POLygon Command - Drawing Filled Polygons

The Polygon Command causes the drawing of a polygon filled with the currently selected fill color or fill pattern. The boundary or outline is drawn in the selected boundary color and line type.

The closing side of the polygon is drawn automatically; accordingly, the beginning coordinates need not be repeated in the command.

When only one pair of coordinates is specified a move is executed and nothing is drawn. When only two pairs of coordinates are give, a line is drawn. Up to 1ØØ vertices may be specified per polygon.

Syntax: **!POL**ygon **xb,**yb{**,**xv_1**,**yv_1{**,**xv_2**,**yv_2 ...**,**xv_{n-1}**,**yv_{n-1}}}<**CR**>

Where: **xb** = the x coord. of the beginning point, which is also the final vertex (positive decimal integer)

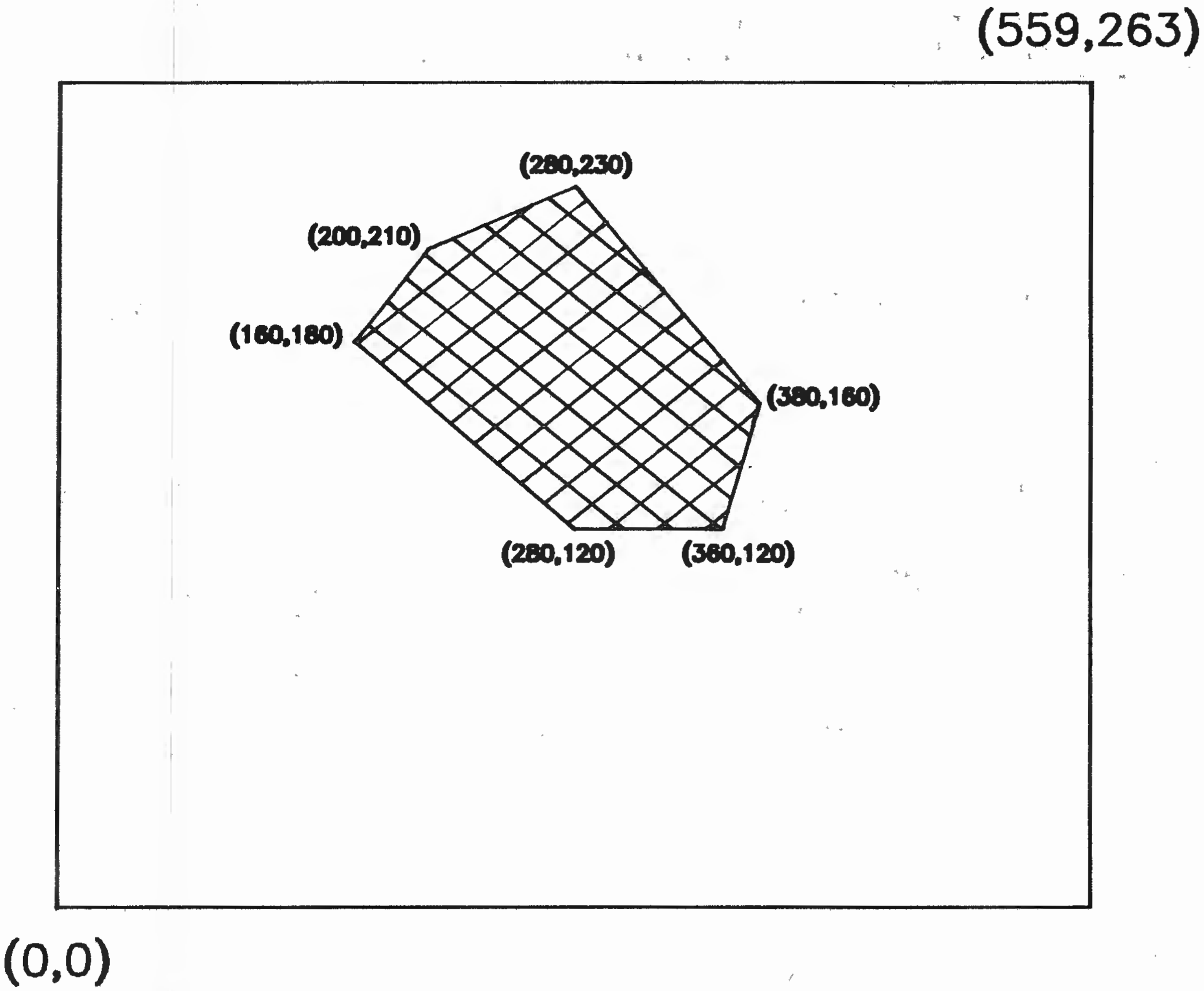
yb = the y coord. of the beginning point, which is also the final vertex (positive decimal integer)

xv = the x coord. of a vertex (positive decimal integer)

yv = the y coord. of a vertex (positive decimal integer)

Example:

With a single command, the graphics beam is moved to (16Ø,12Ø) and filled polygon is drawn beginning there by specifying a sequence of points in absolute (2427 screen) coordinates. These points are the vertices of the polygon.



```
!POL 160,180,200,210,280,230,380,160,360,120,280,120<CR>
```

RPOlygon Command - Drawing Filled Polygons Using Relative Coordinates

The Relative Polygon Command has the same effect as does the Polygon Command. However, coordinates are specified relative to the current beam position, and can be positive or negative.

Up to 1ØØ vertices may be specified in the command. If only beginning point coordinates are specified, a move is executed. If only one vertex is specified, a line is drawn.

Syntax: **!RPO**lygon **dx**b**,dy**b{**,dx**v_1**,dy**v_1 ... **,dx**v_{n-1}**,dy**v_{n-1}}<**CR**>

Where: **dx**b = delta x from beam position to beginning point, which becomes the last vertex (positive or negative decimal integer)

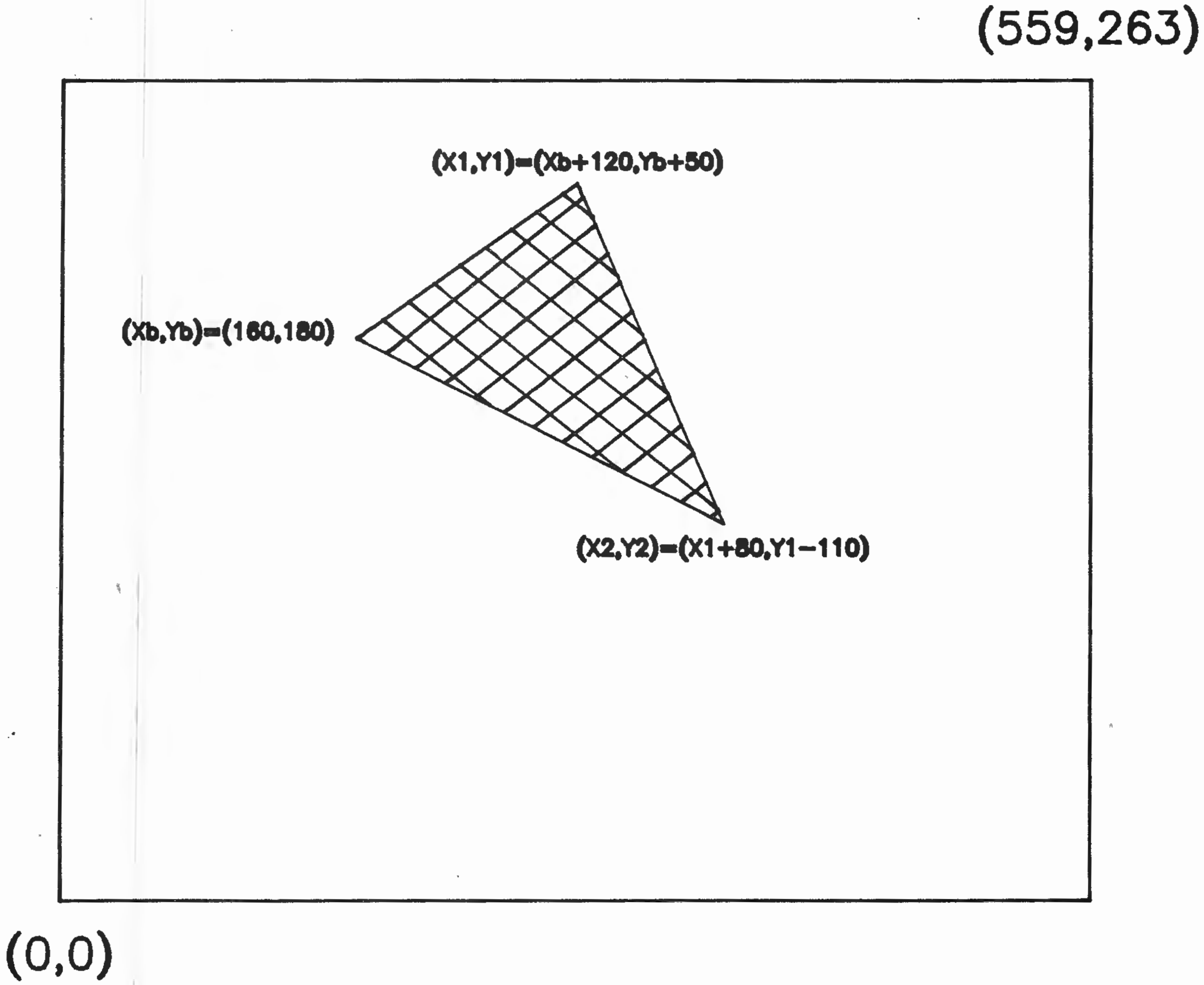
dyb = delta y from beam position to beginning point, which becomes the last vertex (positive or negative decimal integer)

dxv = delta x from the beginning point, or from the last specified vertex, to the vertex (positive or negative decimal integer)

dyv = delta y from the beginning point, or from the last specified vertex, to the vertex (positive or negative decimal integer)

Example:

The graphics beam is first moved to point (Ø,Ø) with a Vector Command and a filled triangle is then drawn using relative coordinates.



```
!VEC 0,0
!RPO 160,180,120,50,80,-110<CR>
```

Note: When drawing involves coordinate scaling, scaling errors may occur.

Graphics Input (GIN) Mode Commands

ENAble Command - Entering Graphics Input (GIN) Mode

The Enable Command is sent to place the terminal in GIN mode. When the terminal is in GIN mode, the operator is permitted to interact with a host program in the creation of graphic displays.

When the terminal is in this mode of operation, the operator may change the graphics beam position either with a light pen or by positioning a crosshair cursor with the Arrow Keys and then striking another key. A selection of one of these input devices (light pen or crosshair cursor) must be made by the operator with the terminal in Setup Mode, before GIN Mode operation is begun. The crosshair cursor is normally selected. The light pen is an optional device not installed on all 2427s.

The crosshair cursor and Arrow keys are used in the following manner: Upon entry into GIN Mode the cursor is displayed at (Ø,Ø). It may be moved in one pixel increments by tapping the appropriate Arrow keys. By holding the keys down, the cursor may be moved quickly over the entire graphics region.

Once the cursor has been positioned, the operator may cause the graphics beam to be moved to the cursor position and cursor position coordinates to be sent to the host by striking one of the ASCII character keys on the keyboard. If the host is properly programmed, it can use these coordinates to generate new terminal command sequences. The terminal transmits the coordinates according to the coordinate scaling factor selected with the Shrink Command. The coordinates are transmitted as part of a Data sequence:

!DAT Ø3,key,x-coord,y-coord,CØØ;

Where: **key** = the ASCII Decimal Equivalent of the code generated by the key pressed (in three digits)

x-coord= the beam position's x coordinate in decimal (three digits)

y-coord= the beam position's y coordinate in decimal (three digits)

CØØ = a dummy color code. (Unlike the Tektronix 4Ø27, the 2427 does not send color information to the host.)

A definite number of Data sequences may be specified within the Enable Command. In case some number of sequences is specified, the terminal will automatically leave GIN mode after that number sequences have been transmitted to the host. If no number is specified in the Enable Command, the terminal will remain in GIN mode until a Disable command, described below, is received.

When the terminal leaves GIN Mode, either after sending a specified number of Data sequences or after receiving a Disable Command, it transmits Trailer Codes specified in Setup Mode (Graphics Submenu).

While the terminal is in GIN mode it can be commanded to "ink" vectors as the graphics beam is moved from one position to another. For example, when the crosshair cursor is moved from the current beam position with the Arrow keys and an ASCII key is then struck to adjust the beam position to the cursor, a line can be drawn as the beam is moved. The INK command, described below is used to enable and disable this proceedure.

> Note: While the terminal is in GIN Mode it will accept and execute all 4Ø27 type commands received without leaving GIN Mode and without transmitting Data sequences to the host after movements of the graphics beam which result from these commands.
>
> The terminal will also accept straight text from the host.

Syntax: **!ENA**ble {**seq**}<**CR**>

Allowable parameter values:

seq: 1-n = the number of beam position reports (Data sequences) to be transmitted to the host before GIN Mode is cancelled.

none = an infinite number of beam position reports to be transmitted before GIN Mode is cancelled.

Examples:

!ENAble<**CR**>	GIN Mode is enabled indefinitely.
!ENAble1Ø<**CR**>	GIN Mode is enabled for 1Ø beam reports.

INK Command - Drawing Vectors in GIN Mode

The Ink Command is issued to enable the drawing of vectors in GIN Mode. When "inking" is enabled, the terminal automatically draws a vector from the current graphics beam position to the crosshair or light pen position when an ASCII character key is struck or the light pen touch switch is actuated. The new graphics beam position (vector end point) is reported to the host in the normal manner for GIN Mode.

Syntax: **!INK** {**setting**}<CR>

Allowable parameter values:

setting:	Y = Yes, enable inking
	N = No, disable inking
	no value = Yes, enable inking

Examples:

!INK<CR>	Enable inking.
!INK N<CR>	Disable inking.

DISable Command - Leaving GIN Mode

If no number of beam position reports is specified in the Enable Command which is sent to place the terminal in GIN Mode, the terminal remains in this mode until it receives a Disable Command.

No parameter is evaluated in this command.

Syntax: **!DIS**able<**CR**>

Using 4Ø1Ø Type Graphics Commands in 4Ø27 Mode Operation

The terminal's ability to respond to both 4Ø1Ø type graphics commands and 4Ø27 type commands can be utilized to advantage in certain applications in which use of host processing and I/O time must be kept to a minimum. The most efficient use of host time is achieved by programming the host to send the terminal graphics commands in 4Ø1Ø form whenever simple graphics elements like points or vectors are to be drawn and in 4Ø27 form whenever more complex graphics elements are to be drawn.

When a problem solution is to be represented as a graphic point or vector, the host can express the solution just as quickly in the form of 4Ø1Ø commands as in the form of 4Ø27 commands. Little extra host processing time is involved in such cases. On the other hand, the 4Ø1Ø commands can be transmitted much more quickly, since they involve fewer characters. Thus, when simple graphic elements are to be drawn, 4Ø1Ø commands and 4Ø27 commands can be formed with about the same amount of host processing, but host I/O time is significantly reduced when 4Ø1Ø type commands are used.

When a problem solution is to be represented as a complex graphic element, one involving an arc or a filled region, for example, the host can express the solution much more quickly in the form of 4Ø27 commands than in the form of 4Ø1Ø commands. Here, the fact that 4Ø27 commands generally have more characters and take longer to transmit is negligible. Host processing time is the crucial factor. 4Ø27 commands include high level commands for drawing circles, filling regions, etc. which are not available in 4Ø1Ø Mode. An arc could be created in 4Ø1Ø Mode terminal operation only with a long series of commands, each of which would contain coordinates which the host would have to calculate. In 4Ø27 Mode operation, an arc can be created in response to a single command from the host, containing a very few numeric parameters.

The host can be programmed to command the terminal to switch back and forth between 4Ø27 Mode and 4Ø1Ø Mode as the case demands. It can also be programmed to form most commands in 4Ø27 Mode and to switch into 4Ø1Ø Mode for vector drawing only.

When the terminal is in 4Ø27 Mode, the receipt of a <**GS**> character puts it in 4Ø1Ø vector mode. 4Ø27 Mode settings made with commands like GRAphic, SHRink and CMOde remain in effect. The terminal is then ready to receive a string of ASCII characters representing vector end point coordinates, according to 4Ø1Ø conventions. These conventions are fully described in the following part of the manual. Following the transmission of the 4Ø1Ø Mode coordinates, the host sends a <**US**> to return the terminal to 4Ø27 Mode operation. (Here, 4Ø1Ø vector drawing is not ended by a carriage return as it is in 4Ø1Ø Mode.)

Note: If the terminal is in "Snoopy Mode" (!SNOopy Y<CR>), control characters such as <GS> and <US> do not have their normal effects; instead, their mnemonics are displayed as text. Therefore, 4010 vector drawing cannot be done while the terminal is in 4027 Snoopy Mode. See Part Four, Command Group A, for more information about Snoopy Mode.

Example:

To save host I/O time, the following 4010 command

```
<GS>#D!r#D2x=V2x=V!r#D!r<US>
```

may be inserted into a series of 4027 Mode commands to cause the drawing of a series of vectors. Note that the terminal is returned to 4027 Mode following the 4010 type command. Assuming a !SHR Y<CR> command in effect, the equivalent 4027 Mode command is

```
!VEC100,50,100,600,950,600,950,50,100,50<CR>
```

COMMAND GROUP D: COLOR COMMANDS

The Color Command, in Command Group B, is used to select one of the eight color registers or one of up to 12Ø fill patterns for use when drawing graphic elements. Several other commands have the color register addresses (CØ-C7) and fill pattern addresses (PØ-P119) for arguments as well.

Upon entry into 4Ø27 Mode the color registers are loaded with codes defining the eight basic displayable colors. No fill patterns are defined.

The commands in this group are used to modify the codes in the color registers (giving the user access to the 2427's full range of 64 displayable colors), to define fill patterns, and to manipulate color for the purpose of distinguishing intersecting display elements (Complement Mode Command).

The commands include:

- CMOde
- MAP
- RMAp
- MIX
- PATtern

CMOde Command - Using Color to Distinguish Overlapping Graphics Elements

Upon entry into 4Ø27 Mode the terminal is set up to "draw" graphic elements and text in a given area of the display without regard to what might be displayed there already. For example, a solid line is drawn by "turning on" each pixel along the line in the selected color, regardless of the pixel's previous state. Accordingly, when a new graphic element is drawn over all or part of another element, the original element is masked from view in the area of intersection. This effect is generally not a problem where the intersection of lines is concerned. However, it is sometimes undesirable, for example when the complete outlines of overlapping filled figures should be viewable.

Complement Mode can be used to avoid this problem. When Complement Mode is enabled, graphics commands cause the activation of pixels not on the basis of the currently selected drawing color alone, but also on the basis of their states before the commands are issued. The color register whose content will determine a pixel's state is always selected by writing a three-bit pointer into the screen memory register assigned to that pixel. In Complement Mode this three-bit pointer is calculated by combining the pointer already stored at that memory register with the pointer corresponding to the currently selected drawing color, according to the Exclusive OR function.

For example, if a pixel was activated in response to a graphics command when C6 was selected for drawing, and the pixel is to be reactivated in response to a new command with C4 selected for drawing, then, if Complement Mode has been enabled, color register C2 will be assigned to the pixel when the new command is executed. C2 is selected as shown in the following chart.

```
Pointer Originally Assigned             Pointer for Current Color

B2      B1      BØ                      B2      B1      BØ

1       1       Ø                       1       Ø       Ø

|       |       |                       |       |       |
|       |        ---------------EOR---------------------
|       |                       |       |       |
|        ---------------EOR---------------------
|                       |       |       |
 ---------------EOR---------------------
                |       |       |

                Ø       1       Ø

                B2      B1      BØ
                Pointer Used for Drawing
```

The Complement Mode can also be used in other ways. When the color registers are loaded with the default color codes (and only then, unless the user has reprogrammed the registers appropriately), the command may be used to generate the complement of the color last used in drawing. The color black is selected (C7), then Complement Mode is selected, and then the figure is redrawn. In this case, if C6 (magenta) were originally selected, the new color would be red (C1).

Note that if a filled figure is drawn over another filled figure without changing the drawing color, but with Complement Mode, the intersecting area will be black.

Syntax: **!CMO**de {**setting**}<**CR**>

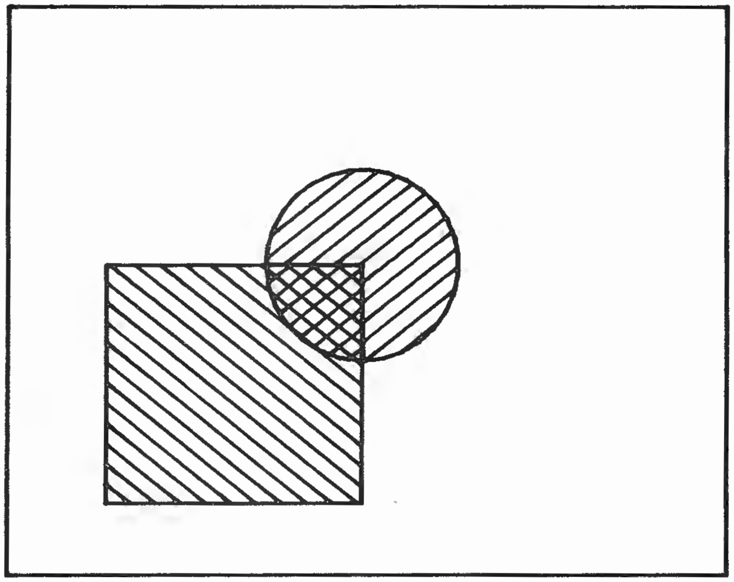
Allowable parameter values:

setting:	Y = Yes, enable Complement Mode
	N = No, disable Complement Mode
	no value = Enable Complement Mode

Example:

Suppose that a workspace, a graphics region, and a shrink factor have been specified to give a coordinate system defined by (Ø,Ø) and (559,239). A pie with radius 75 has been drawn about the point (275,13Ø) with the color in C6 (default magenta). A filled rectangle is now to be drawn with C4 (default yellow) in such a way that it overlaps the pie. Unless Complement Mode is selected, that part of the pie which will intersect with the rectangle will be masked off by the rectangle's yellow filling. With Complement Mode selected, the area of intersection will be green (C2), and the outlines of both figures will be completely visible.

An illustration of the situation is given on the next page, along with the commands needed to create the display.



```
!WOR3Ø!GRA1,3Ø!SHR D<CR>
!COL C6<CR>
!VEC275,13Ø<CR>
!PIE75<CR>
!COL C4<CR>
!CMO<CR>
!POL75,3Ø,275,3Ø,275,13Ø,75,13Ø<CR>
```

MAP Command - Changing the Contents of the Color Registers

Upon entry into 4Ø27 Mode, the eight color registers CØ-C7 are loaded with the codes for the eight basic colors (CØ-white, Cl-red, C2-green, C3-blue, C4-yellow, C5-cyan, C6-magenta, C7-black). The codes in these registers may be changed to make any of the other 56 possible colors available for display. Changes are made with the Map Command or with one of the other color control commands described below.

The Map Command permits colors to be defined by a hue angle, a saturation level, and a lightness or intensity level, expressed in absolute terms.

- A color's hue is defined with reference to a color wheel or "map" as the angular distance, in degrees, counter-clockwise from a reference line cutting across the wheel through blue to that color.

- Saturation is expressed as a percent with reference to a pure color, that is, a color in which no amount of its complement is mixed. A color's complement is the color opposing it (18Ø° away) on the wheel. When a color is mixed with its complement in equal parts, the result is a grey. Accordingly, the percentage of saturation may also be considered as a measure of a color's difference from a grey at a given level of intensity.

- Lightness or intensity is expressed as a percent with reference to white, which is represented at the center of the color wheel. (Black is at the outer edge of the wheel.) Colors on the same concentric band of the wheel have the same lightness level, and in general the bands closer to the wheel's center have higher lightness levels.

Syntax: **!MAP Cn,hue{,light{,sat}}<CR>**

Allowable parameter values:

Cn: CØ-C7 = one of the eight color registers, whose color code is to be modified

hue: Ø-36Ø = hue angle in degrees

light: Ø-1ØØ = percent lightness
none = 1ØØ%

sat: Ø-1ØØ = percent saturation
none = 1ØØ%

The definitions of the color register contents in terms of Map Command parameters are always available throught the SYStat Command. The eight basic colors are defined as follows:

Color Register	Default Color	Hue	Light	Sat
CØ	White	Ø	1ØØ	1ØØ
C3	Blue	Ø	5Ø	1ØØ
C6	Magenta	6Ø	5Ø	1ØØ
Cl	Red	12Ø	5Ø	1ØØ
C4	Yellow	18Ø	5Ø	1ØØ
C2	Green	24Ø	5Ø	1ØØ
C5	Cyan	3ØØ	5Ø	1ØØ
C7	Black	Ø	Ø	1ØØ

The color map on the following page contains all of the 64 colors which can be displayed by the 2427. The map can be a useful tool for obtaining the correct Map Command parameter values to define a particular color. The following are step by step instructions for using the color map to find the parameter values for a given color.

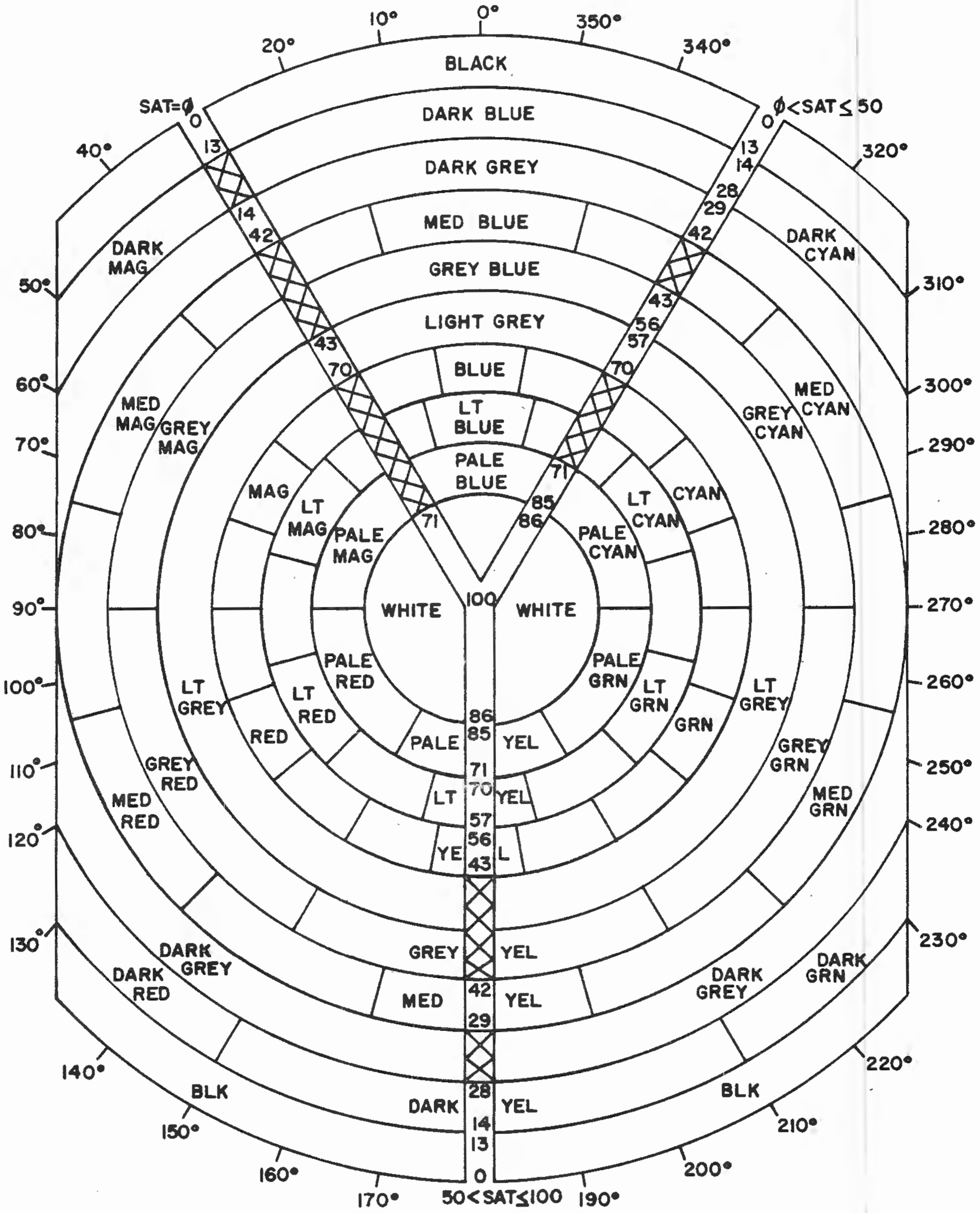
1) Look to the wheel to find the color to be "mapped" into a color register with the command. Lay a straightedge on the wheel in such a way that it represents a radial line cutting through the chosen color. Look along the straightedge to the margin of the wheel to find the hue angle for the color.

2) Look along the circumference of the band in which the chosen color is situated until a lightness scale is found which contains a set of numbers for that band. The numbers on the scale at the inner and outer edges of the band define a range of values for lightness. Any number within this range can be used to evaluate the lightness parameter in the Map Command.

   Note that more than one range of lightness values is given for some color bands, while only one range is given for others. In case two ranges are given, either may be used, but the one used determines the saturation range. (See the next step.)

3) Once a range of lightness values has been found for the band, note in which of the three scales radiating from wheel center that range is represented. Each scale is associated with a particular range of saturations. When the scale in the upper left quadrant of the wheel is used, the saturation level must be Ø. When the scale in the upper right quadrant is used, the saturation level must be greater than Ø and less than or equal to 5Ø%. The scale at bottom center is associated with saturation levels over 5Ø%.

Color Map


0°
10°
350°
20°
340°
BLACK
SAT=Ø
Ø<SAT≤50
DARK BLUE
40°
320°
DARK GREY
MED BLUE
GREY BLUE
LIGHT GREY
BLUE
LT BLUE
PALE BLUE
DARK MAG
DARK CYAN
50°
310°
MED MAG
GREY MAG
GREY CYAN
MED CYAN
60°
300°
70°
290°
MAG
LT MAG
PALE MAG
LT CYAN
CYAN
PALE CYAN
80°
280°
90°
270°
WHITE
100
WHITE
100°
260°
PALE RED
LT RED
RED
LT GREY
PALE GRN
LT GRN
GRN
LT GREY
PALE
YEL
LT
YEL
YE
L
110°
250°
GREY RED
MED RED
GREY GRN
MED GRN
120°
240°
GREY
YEL
MED
YEL
130°
230°
DARK GREY
DARK RED
DARK GREY
DARK GRN
140°
220°
BLK
DARK
YEL
BLK
150°
210°
160°
200°
170°
190°
50<SAT≤100
0
13
14
28
29
42
43
56
57
70
71
85
86

Example:

Suppose one wanted to assign the color orange to register C7. Referring to the color map, he would find orange on the second band out from the center (white region) at an angle of 15Ø$_{o}$. Looking along the band containing orange, he would find that the lightness scale at the bottom center of the wheel shows lightness values for that band in the range 57-7Ø. He would pick a lightness value in this range, say 6Ø. He would then note that this lightness scale is associated with saturation levels over 5Ø%. He would pick a saturation value in this range, say 9Ø%. Accordingly, he would give the terminal the following command:

```
!MAP C7,15Ø,6Ø,9Ø<CR>
```

Note that once orange (or some other color) is assigned to C7, black (default) is no longer available unless it is assigned to some register other than C7 with another Map command.

RMAp Command -

The Relative Map Command has the same use as does the Map Command. However, the parameters in this command are evaluated relative to the current values rather than in absolute terms.

The current color parameter values assigned to a given color register (CØ-C7) are easily determined by the operator, by means of the SYStat Command. The values assigned to the register whose content is to be changed are the reference values used with this command.

Syntax: **!RMAp Cn,d-hue**{**,d-light**{**,d-sat**}}<**CR**>

Allowable parameter values:

Cn:	CØ-C7	=	one of the eight color registers, whose content is to be altered
d-hue:	pos or neg integer |Ø-36Ø|	=	the amount the current hue angle is to be changed, in degrees
d-light:	pos or neg integer |Ø-1ØØ|	=	the amount the current lightness is to be changed, in percent
	no value	=	no change
d-sat:	pos or neg integer |Ø-1ØØ|	=	the amount the current saturation is to be changed, in percent
	no value	=	no change

Example:

Suppose that the color orange is to be displayed with C4, instead of the color yellow (default color for the register).

1) The current values assigned to C4 are:

hue = 180° lightness = 50% saturation = 100%

2) The values needed for orange, as determined from the Color Map, are:

hue = 150° lightness = 60%* saturation = 100%**

3) The changes required for each parameter are:

hue = -30° lightness = 10% saturation = 0%

The command to be given to effect the change in C4's assignment from yellow to orange is therefore:

```
!RMAp C4,-30,10<CR>
```

*This value may vary within the range 57% - 70%.

**This value may vary within the range 51% - 100%.

MIX Command - Alternate Means of Changing Colors

The Mix Command, like the Map and Relative Map Commands, is used to change the contents of the eight color registers CØ-C7 in order to display any of 56 colors available in addition to the eight basic colors. However, the Mix Command does not define new color register assignments in terms of hue, lightness, and saturation. Instead, it uses percentages of the three prime colors (red, green, and blue), whose mixture according to some ratio constitutes a new color.

Within the range of Ø - 1ØØ%, four color levels are defined for each prime color, as follows:

Level	Percentage
Ø (off)	Ø
1	1 - 25
2	26 - 5Ø
3 (full on)	51 - 1ØØ

Since each prime color has four possible levels, the total number of combinations possible is 64. The colors defined by this command are in fact the same colors which can be defined with the Map and RMap Commands.

When the color in a given register has been defined with this command, the definition appears in the SYStat message in terms of percentages of red, green and blue, and not in terms of hue, lightness and saturation.

Syntax: **!MIX Cn**{**,red**{**,grn**{**,blu**}}}<**CR**>

Allowable parameter values:

Cn: CØ-C7 = one of the eight color registers, whose content is to be modified by this command

red: Ø-1ØØ = percent red
none = 1ØØ percent red

grn: Ø-1ØØ = percent green
none = 1ØØ percent green

blu: Ø-1ØØ = percent blue
none = 1ØØ percent blue

A value for blue may not be specified unless a value for green is specified, and green may not be evaluated in turn unless red is evaluated.

Examples:

!MIX C3<CR>

Color register C3 is loaded with the code for white, since the command specifies 100% red, 100% green and 100% blue.

!MIX C3,80,80,0<CR>

Color register C3 is loaded with the code for yellow. The command specifies 80% red (state 3 - full on), 80% green (state 3 - full on) and no blue. Since red and green are mixed in equal proportions (both full intensity) the resulting color is yellow. To see this, refer to the color map given with the description of the Map Command. Yellow appears half way between red and green on the map.

!MIX C3,30,30,30<CR>

Color register C3 is loaded with the code for light grey. The command specifies 30% of each prime color. Since all the prime colors are used, and since they are mixed in equal proportions, the result is a neutral (grey) tone. Thirty percent is in the range for state 2; therefore light grey. (On the grey scale, state 0 is black, state 1 is dark grey, and state 3 is white.)

PATtern Command - Defining Fill Patterns

Space is available in the terminal's memory to store up to 12Ø user-defined fill patterns. With regard to memory, the 12Ø possible patterns are treated as a font of user-defined dot matrix characters (font 31), and some of the commands in Group E are relevant to their use. The patterns are generally used to fill polygons. (See the Polygon and Pie Commands in Group C.) They are defined with the Pattern Command, which is related to the Symbol Command described in Command Group E.

Patterns are defined with reference to an 8 x 14 pixel matrix, in accordance with Tektronix protocol. However, when they are actually displayed, they are scaled down to fit into a 7 x 8 pixel matrix. When an area is filled, the pattern in use is placed repeatedly in character cells along the workspace rows within the area. Scaling errors may occur when more complicated patterns are reduced for display, resulting in occasional gaps in the filled area.

Syntax: **!PAT**tern **Pn**{,**Cn**$_{bg}$},**Cn**$_{fg1}$,**row**$_1$ ... ,**row**$_{14}${,**Cn**$_{fg2}${,**row**$_1$ ... ,**row**$_{14}$}<CR>

Allowable parameter values:

Pn: PØ-P119 = The identifier for the pattern being defined.

Cn$_{bg}$: CØ-C7 = One of the eight color registers, whose content defines the color of the background dots in the 8 x 14 matrix.

none = Background color determined by the content of C7 (default black).

Cn$_{fg}$: CØ-C7 = One of the eight color registers, whose content defines the color of foreground dots in the 8 x 14 matrix.

row: Ø-255 = The decimal equivalent of an eight-digit binary number, in which the "1"s correspond to those dots in a row of the matrix which are to be foreground.

A pattern is defined by a string of 14 decimal numbers, one number for each row in the pattern (top to bottom). Each decimal number is in the range Ø-255 and is the equivalent of an eight-bit binary number whose digits correspond to the eight pixels in a row of the pattern matrix. The

1s in the binary equivalents of the decimal numbers correspond to the pixels which will represent foreground color. The Øs correspond to the pixels which will represent background color.

The Pattern Command permits the specification of one background color (optional) and up to eight foreground colors. A full string of row numbers must be given for each foreground color. The pattern defined by each successive string will be "overlaid" on the pattern defined by a previous string or strings.

Example:

```
!PAT P5,C2,C3,Ø,Ø,Ø,6Ø,6Ø,6Ø,6Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,C6,Ø,Ø,Ø,Ø,Ø,Ø,Ø,
6Ø,6Ø,6Ø,6Ø,Ø,Ø,Ø<CR>
```

This command defines pattern P5 with C2 (default green) for background, C3 (default blue) for the first foreground, and C6 (default magenta) for a second foreground. The row strings are specified in such a way that C3 will be used only in the upper half of the pattern and C6 in the lower half. The chart below shows how the pattern is constituted.

First Overlay ($C3_{fg}$)		Second Overlay ($C6_{fg}$)		Resulting Pattern
Decimal	Binary	Decimal	Binary	(B=Blue, G=Green M= Magenta)
Ø	Ø Ø Ø Ø Ø Ø Ø Ø	Ø	Ø Ø Ø Ø Ø Ø Ø Ø	B B B B B B B B
Ø	Ø Ø Ø Ø Ø Ø Ø Ø	Ø	Ø Ø Ø Ø Ø Ø Ø Ø	B B B B B B B B
Ø	Ø Ø Ø Ø Ø Ø Ø Ø	Ø	Ø Ø Ø Ø Ø Ø Ø Ø	B B B B B B B B
6Ø	Ø Ø 1 1 1 1 Ø Ø	Ø	Ø Ø Ø Ø Ø Ø Ø Ø	B B G G G G B B
6Ø	Ø Ø 1 1 1 1 Ø Ø	Ø	Ø Ø Ø Ø Ø Ø Ø Ø	B B G G G G B B
6Ø	Ø Ø 1 1 1 1 Ø Ø	Ø	Ø Ø Ø Ø Ø Ø Ø Ø	B B G G G G B B
6Ø	Ø Ø 1 1 1 1 Ø Ø	Ø	Ø Ø Ø Ø Ø Ø Ø Ø	B B G G G G B B
Ø	Ø Ø Ø Ø Ø Ø Ø Ø	6Ø	Ø Ø 1 1 1 1 Ø Ø	B B M M M M B B
Ø	Ø Ø Ø Ø Ø Ø Ø Ø	6Ø	Ø Ø 1 1 1 1 Ø Ø	B B M M M M B B
Ø	Ø Ø Ø Ø Ø Ø Ø Ø	6Ø	Ø Ø 1 1 1 1 Ø Ø	B B M M M M B B
Ø	Ø Ø Ø Ø Ø Ø Ø Ø	6Ø	Ø Ø 1 1 1 1 Ø Ø	B B M M M M B B
Ø	Ø Ø Ø Ø Ø Ø Ø Ø	Ø	Ø Ø Ø Ø Ø Ø Ø Ø	B B B B B B B B
Ø	Ø Ø Ø Ø Ø Ø Ø Ø	Ø	Ø Ø Ø Ø Ø Ø Ø Ø	B B B B B B B B
Ø	Ø Ø Ø Ø Ø Ø Ø Ø	Ø	Ø Ø Ø Ø Ø Ø Ø Ø	B B B B B B B B

COMMAND GROUP E: TEXT IN THE WORKSPACE AND GRAPHICS REGION

The commands in this group include, in the order presented:

- ATTribute
- CITalics
- CROtate
- CSCale
- CSPace
- CVEctor
- CVSymbol
- DFOnt
- FONt
- SFOnt
- STRing
- SYMbol

When the terminal receives ASCII character codes (32 - 127 decimal) which are not part of a command sequence, it displays characters assigned to these codes either in the workspace or in the monitor, as determined with the Workspace and/or Monitor Commands. A single set of characters is assigned to the ASCII codes for display in the monitor, but various character sets may be assigned to the codes for display in the workspace. (See the Attribute Command.) Text is displayed in the Workspace beginning at the current workspace cursor position.

Note: The workspace cursor is not visible in the display. When the workspace is initially defined, the cursor is located in the upper left corner of the workspace. When a graphics region is defined, the cursor is moved to the workspace line below the region, if there is a line available; otherwise, it is moved to the last line of the graphics region.

The cursor marks the position at which the next character is to be displayed. As text is entered into the workspace, the cursor is moved to the right one character space at a time until the end of the current line is reached, at which point it is moved to the first space on the next line.

The cursor may be moved by means of commands described in Command Group F.

There is one important exception to the rule stated on the preceding page. When a String Command (described in this section) is received, the terminal displays characters corresponding to arguments within the command. The handling of text transmitted with this command is independent of that specified in the Workspace and Monitor Commands. The display always begins at the current graphics beam position within the graphics region (distinct from the workspace cursor position). The characters displayed are those belonging to the preselected font (Attribute Command).

The character sets or fonts which can be used in the workspace and graphics region include two fonts of standard dot matrix characters (ASCII and APL) and one font of standard vector characters (ASCII). The definitions of the characters in these fonts are permanently stored in a read-only memory within the terminal.

One dot matrix font and one vector font may be defined by the user. The definitions of the characters belonging to user-defined fonts are stored in a portion of the terminal's read-write memory (RAM). These definitions are retained only while power is applied to the terminal.

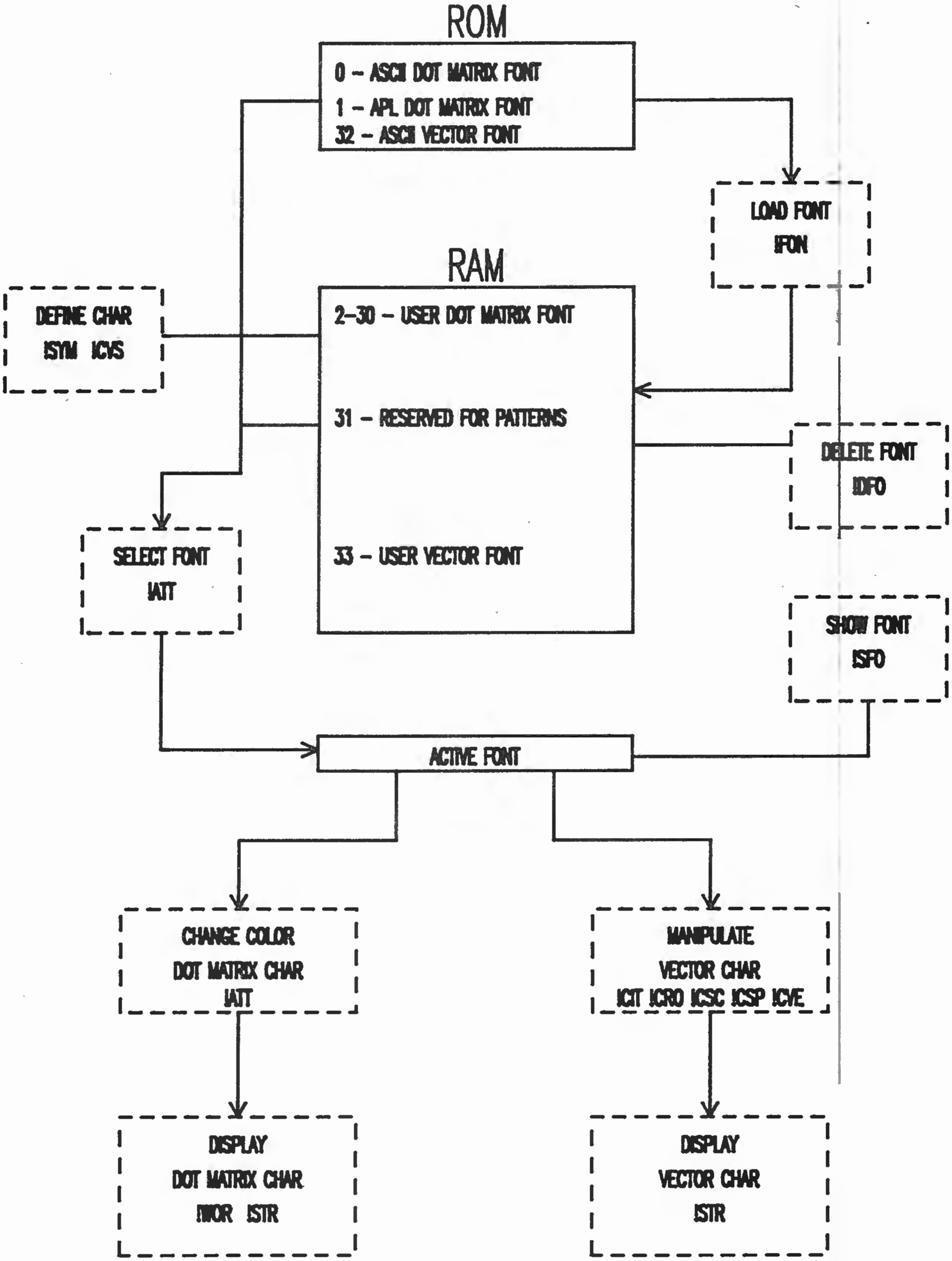
The user-defined fonts may include standard characters which have been "downloaded" from the standard fonts. (See the Font Command.) Individual characters within a font may be defined using the Symbol and CVSymbol Commands.

A dot matrix character is defined with reference to an 8 x 14 pixel matrix but is displayed on a 7 x 8 pixel matrix. (See the Symbol Command.) A vector character is defined with reference to an (x,y) coordinate system, in which x and y are integers representing numbers of pixels. The limits of a vector character coordinate system are represented by the points (-63,-63) and (63,63).

Vector characters may be displayed with various sizes, spacings and orientations as determined with the CITalics, CROtate, CSCale, CSPace and CVector Commands.

The colors with which characters are to be displayed in the workspace can be specified independently of the colors used to draw graphs. Colors may be selected both with the Attribute Command and with the commands used to define characters and fonts.

The chart on the following page shows how the various commands in this group are involved in the manipulation of characters and character fonts.


ROM
0 – ASCII DOT MATRIX FONT
1 – APL DOT MATRIX FONT
32 – ASCII VECTOR FONT
LOAD FONT
!FON
RAM
DEFINE CHAR
!SYM !CVS
2–30 – USER DOT MATRIX FONT
31 – RESERVED FOR PATTERNS
DELETE FONT
!DFO
SELECT FONT
!ATT
33 – USER VECTOR FONT
SHOW FONT
!SFO
ACTIVE FONT
CHANGE COLOR
DOT MATRIX CHAR
!ATT
MANIPULATE
VECTOR CHAR
!CIT !CRO !CSC !CSP !CVE
DISPLAY
DOT MATRIX CHAR
!WOR !STR
DISPLAY
VECTOR CHAR
!STR

ATTribute Command - Selecting the Character Font to be Displayed

Upon entry into 4Ø27 Mode, Font Ø, the ASCII dot matrix font, is selected for the display of text in the workspace. The Font Ø characters are displayed in white on a black background, because color register CØ (default content white) is selected to define the foreground color of each dot matrix and register C7 (default content black) is selected to define the background color.

The Attribute Command is used to switch to a different font or to change the color registers used to display the characters of the selected font. Besides Font Ø, one other standard dot matrix font (Font 1, APL) and one standard vector character font (Font 32, ASCII) are always available. An additional dot matrix font (Font 2-3Ø) and an additional vector font (Font 33) are available when they have been assigned characters by the user. (See the Font, Symbol and CVSymbol commands.)

The font to be activated is specified by number, as described below. Note that any number between 2 and 3Ø may be used to specify the user defined dot matrix font, regardless of the number used (2-3Ø) to signify it when assigning characters to it.

Unlike dot matrix characters, vector characters are not displayed by using contrasting foreground and background colors. Or, if one wishes to think of these characters in terms of background and foreground, the background is always black. Accordingly, the use of the Attribute command to change the colors applies only to the foreground when vector characters are involved.

Syntax: **!ATT**ribute {**font**}{**,logical**}{**,visual**{**,-visual**}}**<CR>**

Allowable parameter values:

font:

Ø =	ASCII dot matrix font
1 =	APL dot matrix font
2-3Ø =	User defined dot matrix font (must be loaded with characters before selection)
32 =	ASCII vector character font
33 =	User defined vector character font (must be loaded with characters before selection)

logical: This 4Ø27 parameter is not used in the 2427. It may be evaluated in accordance with 4Ø27 protocol but will not be acted upon.

visual: CØ-C7 = One of eight foreground-background color combinations. The color code in the specified color register gives the foreground color. The background color is determined by the code in C7 (default black).

ICØ-IC7 = One of eight inverted foreground-background color combinations. The code in the specified color register (CØ-C7) gives the background color. The foreground color is given by the code in C7 (default black).

none = No change in colors. Default condition is foreground determined by CØ, background by C7 (default white on black)

-visual: This 4Ø27 parameter is evaluated like **visual.** With the 4Ø27 it is used to specify a color combination to which the displayed characters blink. When this parameter is evaluated in a command sent to the 2427, no blink is produced. Instead, the value of **visual** is overridden by the value of **-visual.**

Examples:

!ATTribute**1<CR>** Font 1, the APL dot matrix font, is selected and no change is made in the colors used for text display.

!ATTribute **C3<CR>** Register C3 is selected to provide the foreground color (default blue). C7 provides the background color (default black). No change of Font is made.

!ATTribute**Ø,IC2<CR>** Font Ø, the ASCII dot matrix font, is selected. C2 (default green) will provide the background color and C7 (default black) the foreground color.

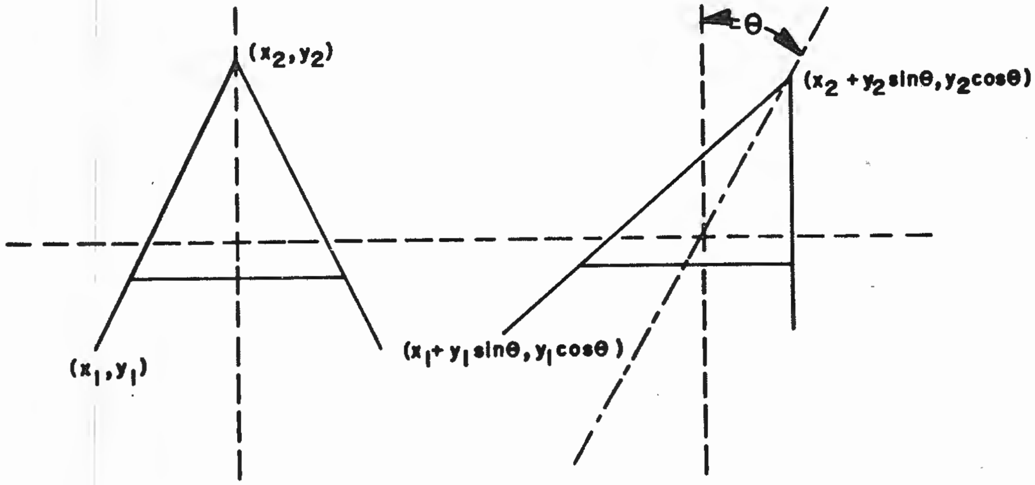
!ATTribute**32,C6<CR>** Font 32, the ASCII vector font, is selected. The vector characters will be drawn with C6 (default magenta). Note that if IC6 had been specified, the vector characters would drawn with C7 and would be invisible unless the content of C7 (default black) had been changed.

CITalics Command - Setting the Slant of Vector Characters

This command is used to cause vector characters to be displayed with a slant or in the Italic style. The characters may be slanted to the right or to the left of vertical by specifying a negative or positive slant angle in the command.

Each vector character is defined with reference to an (x,y) coordinate system whose origin is relative to the position of the graphics beam at the time the character is to be drawn. The x-axis of this system is normally horizontal (parallel to the x-axis of the graphics region) and the y-axis normally vertical (parallel to the y-axis of the graphics region). Accordingly, the coordinates which define the vector character are normally related to graphics region coordinates by constants.

When a character slant angle is specified with the CITalics Command trigonometric functions are introducted into the relation of vector character coordinates to graphics region coordinates. The y-axis of the vector character coordinate system is effectively rotated about the system origin. The graphs below illustrate how an "A" is slanted to the right. The angle is -3Ø°.


(x_2,y_2)
(x_1,y_1)
θ
$(x_2 + y_2 \sin\theta, y_2 \cos\theta)$
$(x_1 + y_1 \sin\theta, y_1 \cos\theta)$

Syntax: **!CITalics {angle}<CR>**

Allowable parameter values:

angle: pos or neg integer |Ø-9Ø| = Slant angle of the y-axis in degrees. When the slant is to the left of vertical, the angle is signed positive. When the slant is to the right, the angle is signed negative.

none = Slant angle Ø° (default)

Examples:

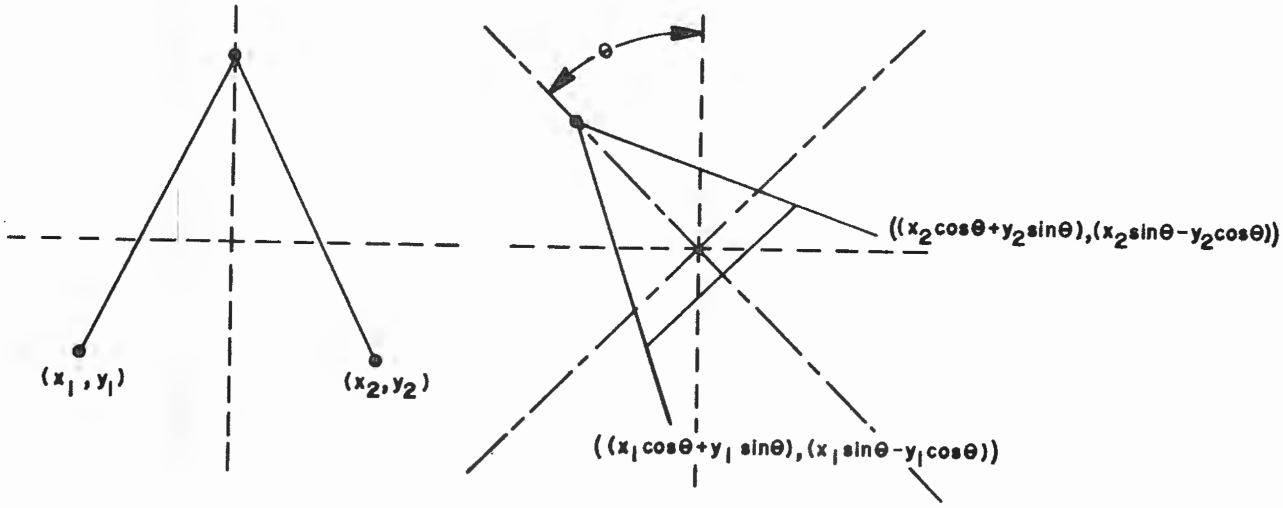
!CITalics3Ø<CR> Vector characters displayed subsequent to the issuance of this command are slanted 3Ø° to the left.

!CITalics -4Ø<CR> Vector characters displayed subsequent to the issuance of this command are slanted 4Ø° to the right.

CROtate Command - Setting the Orientation of Vector Characters

Vector characters are defined with reference to a coordinate system whose x-axis is normally parallel to the x-axis of the graphics region and whose y-axis is normally parallel to the y-axis of the graphics region. The CROtate Command is used to change the orientation of the vector character coordinate system with respect to the graphics region coordinate system. The command causes vector characters to be rotated about their system origins when they are subsequently displayed. The shapes of the characters are not distorted, but their orientations in relation to the viewer are changed.

The graphs below show how a vector character "A" can be rotated with the command. The angle of rotation is 45°.


(x_1, y_1)
(x_2, y_2)
θ
$((x_2\cos\theta+y_2\sin\theta),(x_2\sin\theta-y_2\cos\theta))$
$((x_1\cos\theta+y_1\sin\theta),(x_1\sin\theta-y_1\cos\theta))$

Syntax: **!CROtate {angle}<CR>**

Allowable parameter values:

angle: positive integer = the angle of rotation in degrees, measuring counterclockwise from the vertical

Example:

!CROtate9Ø<CR> Vector characters displayed subsequently to the issuance of this command are rotated 9Ø$^{\circ}$ to the left.

CSCale Command - Setting the Height and Width of Vector Characters

Vector characters are defined with reference to an (x,y) coordinate system with limits given by the points (-63,-63) and (63,63). An x or y value represents a number of pixels. Accordingly, the largest possible vector character would be 127 pixels wide and 127 pixels high, or about 2.15" wide and 3.18" high.

Vector characters are normally displayed with a size determined by the coordinate values used to define them. For example, if the letter "1" is defined by a vector with end points (Ø,-1Ø) and (Ø,1Ø) it will normally be 21 pixels or about 1/2" high when displayed. This character may not be made larger except by redefining it using the CVSymbol Command. However, it may be reduced in height, in width, or in overall size by specifying the size for display as a percentage of the normal or definitional size.

Syntax: **!CSC**ale {**width**{**,height**}<CR>

Allowable parameter values:

width: Ø-1ØØ = percent of maximum width

height: Ø-1ØØ = percent of maximum height

Examples:

!CSCale**5Ø**<CR>	Vector characters displayed subsequently to the issuance of this command are reduced in width by 5Ø% and their heights are not changed.
!CSCale**1ØØ,5Ø**<CR>	Vector characters displayed subsequently to the issuance of this command are reduced 5Ø% in height, with no changes in width.

Note : Values may be larger than 100.
No arbitrary limit has been found.

CSPace Command - Selecting Type of Spacing between Vector Characters

Upon entry into 4Ø27 Mode operation, the horizontal and vertical spacing between successive vector characters is that determined with the CVector Command described below.

The CSPace Command is used to switch between this type of spacing and a predefined standard spacing. With the selection of standard spacing, a succession of vector characters is displayed in such a way that the x-axes of the characters' coordinate systems fall on a single horizontal line in the graphics region. A standard horizontal spacing of 26 pixels (a little less than 1/2") center-to-center is used between the characters.

Syntax: **!CSP**ace {**type**}<**CR**>

Allowable parameter values:

type: N = Non-standard spacing, as defined with the CVEctor Command.

Y = Standard ASCII vector font spacing (26 pixels center-to-center, horizontally; Ø pixels center-to center, vertically)

none = Standard ASCII vector font spacing

Examples:

!CSPace<**CR**>	Standard spacing is selected.
!CSPace **N**<**CR**>	Non-standard spacing, defined by the CVector Command, is selected.

CVEctor Command - Setting the Spacing between Vector Characters

Upon entry into 4Ø27 Mode and following the issuance of the Command **!CSP**ace **N<CR>**, the spacing between vector characters is dependent on parameter values specified with this command.

The x value specified in the command defines the horizontal distance between successive characters in pixels. The distance is measured from the origin of one character's coordinate system to the origin of the next character's system. Horizontally, there are about 59 pixels per inch.

The y value specified in the command defines a vertical displacement between successive characters in pixels. The vertical displacement is always downward and is measured from the origin of one character's coordinate system to the origin of the next. Vertically, there are about 4Ø pixels per inch.

Syntax: **!CVE**ctor **x-dist**{**,y-dist**}<**CR**>

Allowable parameter values:

x-dist: pos integer = number of pixels horizontally between two vector characters

26 = horizontal spacing in pixels upon entry into 4Ø27 Mode (center to center)

y-dist: pos integer = number of pixels vertically between two vector characters

Ø = vertical spacing in pixels upon entry into 4Ø27 Mode

none = vertical spacing Ø

Examples:

!CVEctor**2Ø**<**CR**> The horizontal spacing, center-to-center, between vector characters is set at 2Ø pixels. The vertical displacement of successive characters is Ø; accordingly, characters are displayed along a horizontal line.

!CVEctor2Ø,5<CR>

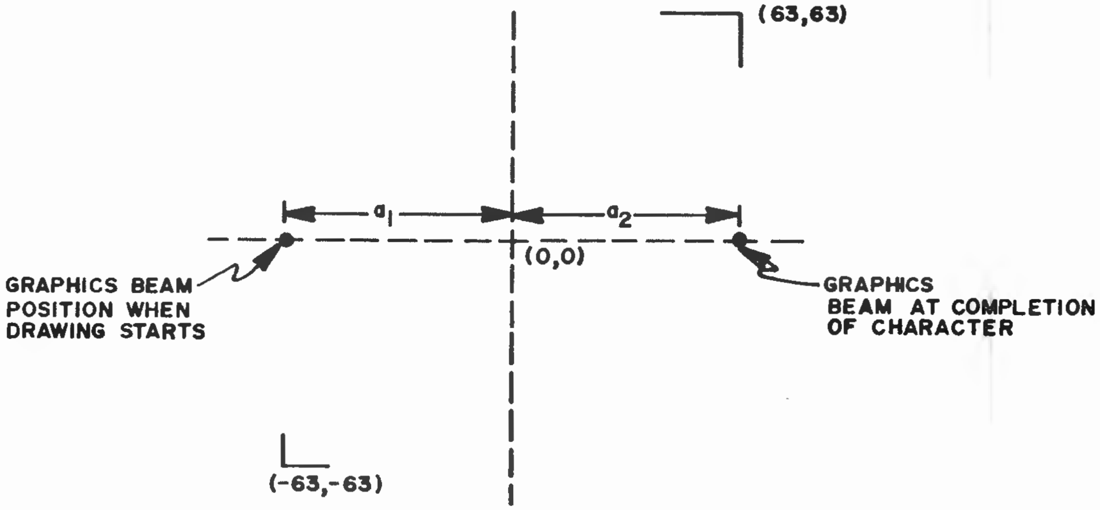
The horizontal spacing, center-to-center, between characters is set at 2Ø pixels. The vertical displacement of successive characters is 5 pixels downward. A string of characters will be displayed on a line which descends from left to right at an angle of about 14° from horizontal.

CVSymbol - Defining Vector Characters

The 2427 provides one font of standard ASCII vector characters (Font 32) and one empty vector character font (Font 33). Font 33 can be filled by "downloading" characters from the standard font (FONt Command), by using the command described here to create characters, or by a combination of the two methods. Once Font 33 has been loaded with characters it may be selected for use with the Attribute Command.

A vector character is defined with the CVSymbol Command in relation to an (x,y) coordinate system which has limits represented by the points (-63,-63) and (63,63). Coordinates, consisting of decimal integer pairs representing points in this system, are specified in the command to define the end points of vectors which are to consitute the character. The command allows for moves of the graphics beam between vectors. When the drawing of a character requires a move between two vectors, the coordinates for the new beam position are given following the pair (64,64).

Besides the coordinate pairs, the command must include numbers (**a**$_1$ and **a**$_2$) indicating the distance, along the x-axis, from the left margin of the system to its origin and from the origin to the right margin. These numbers are used to place the orgin of the system in relation to the graphics beam position at the time drawing starts, and to reposition the graphics beam after drawing has been completed. (Note that these values have an effect on the horizontal spacing between characters and should be taken into account when the CVEctor Command is used.)


(63,63)
a1
a2
(0,0)
GRAPHICS BEAM POSITION WHEN DRAWING STARTS
GRAPHICS BEAM AT COMPLETION OF CHARACTER
(-63,-63)

Normally, the points within the vector character coordinate system correspond to pixels. Accordingly, the largest possible vector character would be 127 pixels high and 127 pixels wide, or about 3.18" high and about 2.15" wide. Once characters have been defined they may be scaled down when displayed by the use of the CSCale Command.

Note that a font parameter is not included in the command. All characters defined with this command are automatically assigned to Font 33.

Syntax: **!CVSymbol chr,**a_1**,**a_2**,**x_1**,**$y_1$$\{$**,**$x_2$**,**$y_2$ ... **,**x_n**,**$y_n$$\}$**<CR>**

Allowable parameter values:

chr: 32-127 = The decimal equivalent of the ASCII character code to be used to produce the character defined.

a_1**:** pos integer < 64 = The distance from graphics beam position when drawing starts to the origin of the system on which the character is defined (along the x-axis).

a_2 pos integer < 64 = The distance from the origin of the system on which the character is defined to the graphics beam position at the end of drawing (along the x-axis).

x: integer -63 to 63 = The x-coordinate of a point defining the beginning or end of a vector (relative to the origin of the vector character system).

64 = Used with y=64 when the following pair defines the position to which a move is made

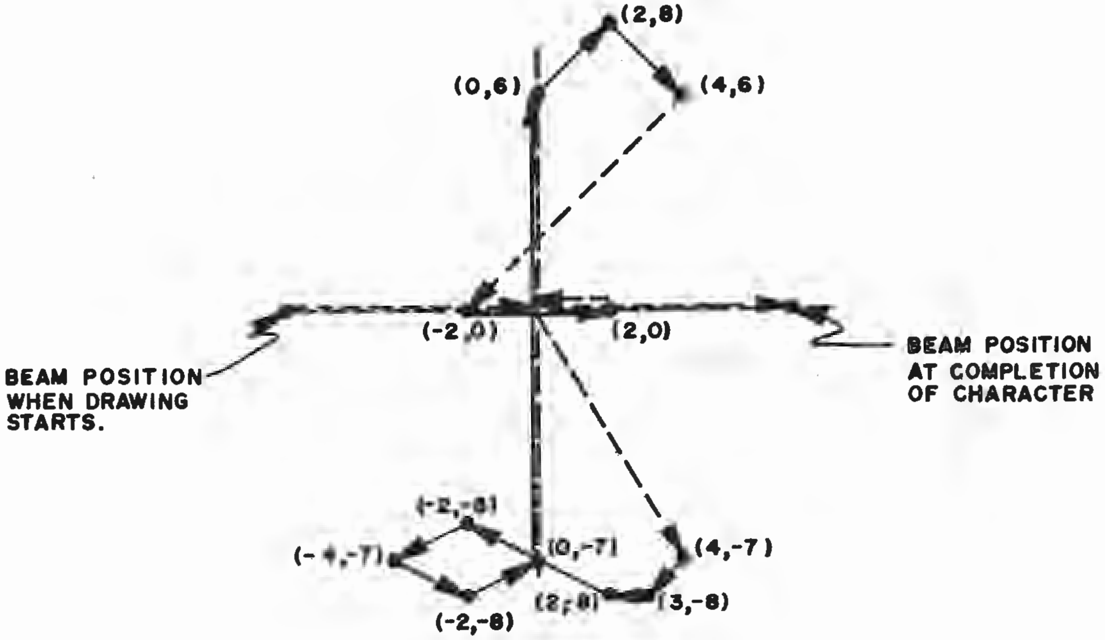
y: integer -63 to 63 = The y-coordinate of a point defining the beginning or end of a vector (relative to the origin of the vector character system).

64 = Used with x=64 when the following pair defines the position to which a move is made.

When defining vector characters of non-standard size, the CSpace and CVector Commands must be used to set character spacing before displays are created. (Uppercase characters in the standard ASCII vector font are 21 pixels high and 15 pixels wide. They are drawn on matrices 28 pixels high and 26 pixels wide.)

Example:

```
!CVSymbol35,7,7,4,-7,3,-8,2,-8,-2,-6,-4,-7,-2,-8,Ø,-7,Ø,6,2,8,
4,6,64,64,-2,Ø,2,Ø<CR>
```


(2,8)
(0,6)
(4,6)
(-2,0)
(2,0)
BEAM POSITION AT COMPLETION OF CHARACTER
BEAM POSITION WHEN DRAWING STARTS.
(-2,-8)
(-4,-7)
(0,-7)
(4,-7)
(2,-8)
(3,-8)
(-2,-8)

This command defines character 35 (ASCII "#") in Font 33 as a British pound symbol which will normally be 17 pixels high when displayed. The command specifies that when the symbol is to be drawn the graphics beam will be moved 7 pixels to the right to establish the origin about which the character will be drawn. It also specifies that after the character has been drawn the beam will be moved 7 pixels to the right from the orgin.

DFOnt Command - Deleting a User Defined Font or Fill Patterns

This command is used to delete or erase user-defined character fonts and user defined fill patterns, which are stored in memory as characters in Font 31.

Syntax: **!DFOnt font<CR>**

Allowable parameter values:

font: 2-3Ø = User defined dot matrix font

31 = Fill patterns PØ-P119 inclusive

33 = User defined vector character font

Examples:

!DFOnt24<CR>	This command causes the user-defined dot matrix font to be deleted. The font number used in the command does not have to be the same as the numbers used in other commands to indicate this font. The only requirement is that the number be in the range 2-3Ø.
!DFOnt31<CR>	This command causes all user defined fill patterns to be deleted.

FONt Command - Assigning Character Sets to Fonts

The Font Command is used to "download" characters from the standard fonts (which are permanently stored in read-only memory or ROM) to the user defined fonts (which are stored in read-write memory or RAM).

With this command the character set either in the ASCII dot matrix font (Fonts Ø) or in the APL dot matrix font (Font 1) can be loaded into the user-defined dot matrix font (Font 2-3Ø).

In addition, the ASCII vector character set in Font 32 can be loaded into the user defined vector character font (Font 33).

This command is useful when a user defined font is to include characters in a standard font, with perhaps just a few different characters. In such a case, the whole standard character set is "downloaded" and then specific characters in the newly created font are modified as necessary with the Symbol or CVSymbol Command.

Syntax: **!FONt** {$\textbf{font}_{or}$,}$\textbf{font}_{dest}${,$\textbf{Cn}_{bg}$,$\textbf{Cn}_{fg}$}<CR>

Allowable parameter values:

$\textbf{font}_{or}$:

Ø = ROM-based ASCII dot matrix font, whose characters are to be moved to font 2-3Ø

1 = ROM-based APL dot matrix font, whose characters are to be moved to font 2-3Ø

32 = ROM-based ASCII vector character font, whose characters are to be moved to font 33

none = ROM-based ASCII dot matrix font (font Ø)

$\textbf{font}_{dest}$:

2-3Ø = The user-definable dot matrix font, to which the characters in font Ø or 1 are to be loaded

33 = The user-definable vector character font, to which the characters in font 32 are to be loaded

$\textbf{Cn}_{bg}$:

This parameter (background color) can be evaluated according to Tektronix protocol but is not used in the 2427

$\textbf{Cn}_{fg}$:

This parameter (foreground color) can be evaluated according to Tektronix protocol but is not used in the 2427

Examples:

!FONtØ,5<CR>	The command causes the ASCII dot matrix font to be "downloaded" to the user defined dot matrix font.
!FONt32,33<CR>	The command causes the ASCII vector character font to be "downloaded" to the user defined vector character font.

SFOnt Command - Displaying the Active Font

A character font is selected for use with the Attribute Command. Once a font has been activated, it may be displayed in response to this command. The command is useful especially for editing user defined fonts. It permits either the entire font or some part of it to be displayed. Characters in the font are displayed in numerical order.

Syntax: **!SFOnt** {**chr**$_b${**,chr**$_e$}}<**CR**>

Allowable parameter values:

chr$_b$: 32-127 = The decimal code for the first character in the font to be displayed

chr$_e$: 32-127 = The decimal code for the last character in the font to be displayed

When no parameter values are specified all characters in the font are displayed.

Examples:

!SFOnt<**CR**>	The entire currently active font is displayed.
!SFOnt32,41<**CR**>	The first ten characters in the active font are displayed.

SYMbol Command - Defining a Dot Matrix Character

Characters for the user defined dot matrix font (the font indicated by any number between 2 and 3Ø) are created with this command.

Dot matrix characters are defined with reference to a matrix with dimensions 8 x 14, in accordance with Tektronix protocol. However, when dot matrix characters are displayed, their patterns are scaled down to fix on a 7 x 8 pixel matrix.

A character is defined by specifying a background color register, one or more foreground color registers, and a string of 14 decimal numbers for each foreground color register specified. Each of the 14 numbers defines one row of dots in the matrix. Each number is the equivalent of an eight-bit binary number whose digits represent the dots in a row. When a digit is a 1, the corresponding dot will have the foreground color; otherwise, it will have the background color.

When more than one foreground color is specified, the 14 row numbers associated with each color constitute a "bit map overlay"; the complete dot pattern is represented when all the overlays are superimposed.

Syntax: **!SYMbol chr,font{,Cn**$_{bg}$**},Cn**$_{fg1}$**,row**$_{1}$ **... ,row**$_{14}$
{,Cn$_{fg2}$**,row**$_{1}$ **... ,row**$_{14}$**}**<CR>

Allowable parameter values:

chr: 32-127 = The ASCII decimal code to be used to display the character being defined (corresponding to one of the alphanumeric keys)

font: 2-3Ø = The user-definable dot matrix font, to which the character is to be assigned (Any number in the given range selects this one font.)

Cn$_{bg}$**:** CØ-C7 = One of the eight color registers, supplying the code for the background color

none = Background color defined by code in C7 (default black)

Cn$_{fg}$**:** CØ-C7 = One of the eight color registers, supplying the code for the foreground (character) color

row: Ø-255 = The decimal equivalent of an eight-digit binary number whose 1s constitute a pattern determining which dots in a row of the 8 x 14 matrix are foreground

none = Row is all background

Example:

```
!SYMbol66,2,CØ,Ø,28,34,34,36,56,36,34,34,34,6Ø,32,64,Ø<CR>
```

This command defines character number 66 (ASCII "B") in Font 2 as a Greek beta. The background will be in accordance with the contents of C7 (default black) since it is not specified in the command. The foreground will be in accordance with the contents of CØ (default white).

Decimal	Binary	Corresponding Pattern
Ø	Ø Ø Ø Ø Ø Ø Ø Ø	. . . .
28	Ø Ø Ø 1 1 1 Ø Ø	. . . * * * . .
34	Ø Ø 1 Ø Ø Ø 1 Ø	. . * . . . * .
34	Ø Ø 1 Ø Ø Ø 1 Ø	. . * . . . * .
36	Ø Ø 1 Ø Ø 1 Ø Ø	. . * . . * . .
56	Ø Ø 1 1 1 Ø Ø Ø	. . * * * . . .
36	Ø Ø 1 Ø Ø 1 Ø Ø	. . * . . * . .
34	Ø Ø 1 Ø Ø Ø 1 Ø	. . * . . . * .
34	Ø Ø 1 Ø Ø Ø 1 Ø	. . * . . . * .
34	Ø Ø 1 Ø Ø Ø 1 Ø	. . * . . . * .
6Ø	Ø Ø 1 1 1 1 Ø Ø	. . * * * * . .
32	Ø Ø 1 Ø Ø Ø Ø Ø	. . * . . . . .
64	Ø 1 Ø Ø Ø Ø Ø Ø	. * . . . . . .
Ø	Ø Ø Ø Ø Ø Ø Ø Ø	. . . .

STRing Command - A Means of Entering Text into the Workspace

This command permits text to be transmitted to the terminal for display in the workspace in the form of a string of alphanumeric characters. Text received by the terminal in this form is displayed beginning at the current graphics beam position and not at the workspace cursor position. This is the case regardless of the type of font selected. Text routing specified with the Workspace or Monitor Commands is not relevant when the String Command is used.

If a dot matrix font is used, text received is displayed normalized to workspace text rows and columns, beginning at the character space which contains the graphics beam. Text may extend beyond the limits of the graphics region to the workspace margin.

When a vector character font is selected, the following points must be taken into account. Because vector characters are drawn as "graphics" they cannot be displayed outside the graphics region. Strings that extend beyond the graphics region are clipped at the boundary of the region. Vector characters are not drawn in a "cell" so there is no background color; they are drawn against the background of the graphics region. In addition, the spacing, slant, etc. of vector characters may depend on specifications given in other commands.

Text entered with the String Command may be specified as one or more delimited strings of ASCII characters, as one or more delimited ASCII decimal equivalent values (ADEs), or as a series of delimited strings and delimited ADE values in combination.

Syntax: **!STRing,string**$_1$**,**{**,string**$_2$**, ... ,string**$_n$**,**}**<CR>**

Where a **string** may consist

1) of the ASCII characters which constitute a text directly, or

2) of the ADE of one character in the text

In place of the comma, the following other characters may be used as string delimiters within the command:

@ # $ % & * ( ) _ + = [ ] : " . ? /

Of these characters, those which are not used as the delimiter in a given command may be included within a string.

The control character, normally the exclamation point "!", may not be used as a delimiter but may be included within a delimited string.

Examples:

Each of the following commands causes the word "Test" to be displayed beginning at the graphics beam position current when the command is received.

```
!STRing"Test"<CR>

!STRing,84,,1Ø1,,115,,116,<CR>

!STRing/T//1Ø1//st/<CR>
```

COMMAND GROUP F: CURSOR MOVEMENT

The commands in this group include:

- BACktab
- TAB
- DOWn
- LEFt
- RIGht
- UP
- JUMp

When operating in the 4Ø27 Mode, the terminal has two text cursors, one in the monitor area and one in the workspace. (The GIN Mode crosshair cursor is not effected by the commands in this section.) Only the monitor cursor is actually visible. However, the invisible workspace cursor is effectively present whenever text is directed into the workspace, and it can be moved with commands in this group.

Each area's cursor marks the position at which the next element of text routed to that area will be displayed. Routing of text can be determined with the Monitor and Workspace Commands described in Group B. Initially, the cursor is placed at the first character position on the first line of the area (upper left corner of the area). However, when a graphics region is defined, the workspace cursor is moved to the first position on the first workspace line below the region, if such a line exists. Otherwise, the workspace cursor is moved to the first position of the last line within the graphics region.

The workspace cursor is automatically repositioned one space to the right as a character is displayed in its area. From the last position on a line the cursor is moved to the first postion of the next line (end of line wrap), except that once the cursor is at the last position of the last line in its area, it is moved back to the beginning of the last line (no end of page wrap). The monitor cursor is moved in a similar manner when wrap is selected in the terminal's Setup Mode. See the description of the selectable wrap feature in Part Three.

The cursors may be moved in other ways with the commands belonging to this group. With the exception of the JUMp Command, which applies only to the workspace cursor, the commands can be used to control either cursor. The source of the command (keyboard or host) determines which cursor is moved according to specifications for text routing made with the Workspace and Monitor Commands. For example, if text entered at the keyboard is routed to the monitor, then a cursor control command issued from the keyboard will affect the monitor cursor; if text transmitted by the host is routed to the workspace, then a cursor command issued by the host afftects the workspace cursor.

Note that both automatic and specially commanded cursor movements are subject to the limitations imposed by the setting of margins and tab stops and the definition of regions with other commands.

A carriage return (other than a carriage return that terminates a command) brings the cursor to the left margin of the current line.

BACktab Command

This command is used to move either the workspace cursor or the monitor cursor a specified number of tab stops to the left. Stops are set with the Stops Command in Group A.

Syntax: **!BACktab**{**count**}<**CR**>

Allowable paramete values:

count: positive integer = number of tab stops

no value = one stop

Examples:

!BACktab3<CR> The cursor is moved to the left three tab stops or to the left margin of the current line if there are not three tab stops to the left on the line.

!BACktab<CR> The cursor is moved one tab stop to the left or to the left margin of the current line.

TAB Command

This command is used to move either the workspace cursor or the monitor cursor a specified number of tab stops to the right. Stops are set with the Stops Command in Group A.

Syntax: **!TAB**{**count**}<**CR**>

Allowable parameter values:

count: positive integer = number of tab stops

no value = one stop

Example:

!TAB6<CR> The cursor is moved six tab stops to the right. If the end of the current line is reached before the sixth stop, the cursor is moved to the next line and tabing continues until the sixth stop is reached. A margin stop is not counted as a tab stop.

DOWn Command

This command is used to move either the workspace cursor or the monitor cursor a specified number of lines down.

Syntax: **!DOWn{count}<CR>**

Allowable parameter values:

count: positive integer = number of character rows

no value = one row

Example:

!DOWn1Ø<CR>	The cursor is moved down 1Ø lines.

LEFt Command

This command is used to move either the workspace cursor or the monitor cursor a specified number of character spaces to the left.

Syntax: **!LEFt{count}<CR>**

Allowable parameter values:

count: positive integer = number of columns

no value = one column

Example:

!LEFt3<CR>	The cursor is moved 3 columns or character spaces to the left.

RIGht Command

This command is used to move either the workspace cursor or the monitor cursor a specified number of columns or character spaces to the right.

Syntax: **!RIGht**{**count**}<**CR**>

Allowable parameter values:

count: positive integer = number of columns

no value = one column

Example:

!RIGht2Ø<**CR**>	The cursor is moved 2Ø columns to the right.

UP Command

This command is used to move either the workspace cursor or the monitor cursor a specified number of lines upward.

Syntax: **!UP**{**count**}<**CR**>

Allowable parameter values:

count: positive integer = number of lines

no value = one line

Example:

!UP1Ø<**CR**>	The cursor is moved up 1Ø lines.

JUMp Command

This command is used to move the workspace cursor only. It permits this cursor to be positioned at any character space within the workspace. When the command parameters (row and column) are not evaluated the cursor is moved to the upper left corner of the workspace (the "home" position). When a graphics region has been defined, the "home" position is at the left margin of the first workspace line below the region, if there is one, or the left margin of the last line in the region.

Syntax: **!JUM**p{**row**{**,col**}}<**CR**>

Allowable parameter values:

row: positive integer = number of the workspace row to which cursor is to be moved

no value = row one

col: 1-8Ø = number of the workspace column to which the cursor is to be moved

no value = column one

Examples:

!JUMp<**CR**> The workspace cursor is moved to row one, column one (the "Home" position) of the workspace.

!JUMp**1Ø**<**CR**> The workspace cursor is moved to row 1Ø, column one of the workspace.

!JUMp**1Ø,39**<**CR**> The workspace cursor is moved to row 1Ø, column 39 of the workspace.

COMMAND GROUP G: TRANSFERRING BLOCKS OF GRAPHICS DATA

The commands in this group include:

- BITs
- MOVe
- RASter
- READback

Three "block transfer" functions are available in 4Ø27 MODE: copying one rectangular part of the display to another location in the graphics region, reading part or all of the display back to the host, and loading display data from the host.

BITs Command - Setting Format of Graphics Data Transfers

The Raster and Read Commands permit the transfer of pixel data between the high resolution display memory and the host computer. A pixel is defined by three data bits; however, when pixel information is transferred with either of these commands, data are handled in eight-bit groups (bytes). Accordingly, it is necessary to specify the manner in which pixel data are to be fitted into bytes before transfers can proceed. The Bits Command is used to make this specification.

To understand the manner in which high resolution memory data are fitted into bytes it must be recalled that there are three memory data bits associated with each pixel. These data bits constitute a three-digit binary number (V_2 V_1 $V_Ø$) which "points" to a color register, whose content determines how that pixel is to be illuminated. The values of the binary pointers do not correspond directly to the decimal numbers used in the names of the color registers (CØ - C7); instead, the pointers refer to the registers according to the following chart:

V_2	V_1	$V_Ø$	Register Name	Default Color
Ø	Ø	Ø	C7	Black
Ø	Ø	1	C1	Red
Ø	1	Ø	C2	Green
Ø	1	1	C4	Yellow
1	Ø	Ø	C3	Blue
1	Ø	1	C6	Magenta
1	1	Ø	C5	Cyan
1	1	1	CØ	White

It may be seen from the chart that V_2 is set to select a subset of registers which normally hold the codes for colors including a blue component, V_1 is set to select a subset of registers which normally hold the codes for colors including a green component, and $V_Ø$ is set to select a subset of registers which normally the codes for colors including a red component. The bits can therefore be interpreted by the host computer as significative of color.

In every data byte transmitted to or from the host which contains pixel information, the two highest order bits, D_7 and D_6, are set to Ø and 1 respectively. Accordingly, the bytes will always have values in the range 4Ø to 7F hexadecimal (ASCII codes for characters from "@" to "DEL"). There is one exception to this rule: 3F hex (ASCII "?") may be substituted for 7F hex. The six low order bits, D_5 through $D_Ø$, in each byte are available to carry the pixel information.

The argument of the Bits Command determines an assignment of video memory data bits to the available data bits within bytes transmitted to or from the host.

Syntax: **!BITs{number}<CR>**

Allowable parameter values:

number: 1 = Bits D_0 - D_5 in each transmitted byte represent the V_0 bits for six successive pixels.* The V_1 and V_2 bits for each pixel are not transmitted. Accordingly, only (default) red color information is transmitted; green and blue color information is not transmitted. If successive pixels are defined by the pointers a_2 a_1 a_0, b_2 b_1 b_0, c_2 c_1 c_0, etc., information about them is transmitted as follows:

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
0	1	f_0	e_0	d_0	c_0	b_0	a_0

2 = Bits D_0 - D_5 in each transmitted byte represent the V_0 and V_1 bits of three successive pixels.* The V_2 bit for each pixel is not transmitted. Accordingly, only (default) red and green (and not blue) color information is transmitted. If successive pixels are defined by the pointers a_2 a_1 a_0, b_2 b_1 b_0, c_2 c_1 c_0, etc., information about them is transmitted as follows:

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
0	1	c_1	c_0	b_1	b_0	a_1	a_0

3, or no value = Bits D_0 - D_5 in each transmitted byte represent the V_0, V_1 and V_2 bits of two successive pixels.* Full color information is transmitted. If successive pixels are defined by the pointers a_2 a_1 a_0, b_2 b_1 b_0, c_2 c_1 c_0, etc., information about them is transmitted as follows:

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
0	1	b_2	b_1	b_0	a_2	a_1	a_0

4, 5 or 6 = Bits D_0 - D_2 and bits D_3 - D_5 of each transmitted byte represent the V_0, V_1 and V_2 bits of a single pixel. Full color information is transmitted twice for each pixel. If successsive pixels are defined by the pointers a_2 a_1 a_0, b_2 b_1 b_0, etc., information about them is transmitted as follows:

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
0	1	a_2	a_1	a_0	a_2	a_1	a_0

*Unless the byte is at the end of a string.

MOVe Command - Moving Data within the High Resolution Memory

This command can be used to reproduce displays within the graphics region at other locations without redrawing the displays on a command by command basis. Reproduction is accomplished by copying a block of data at one group of high resolution memory registers into another group of registers. The original display is nor erased or altered.

The block of data to be moved must correspond to a rectangular area within the graphics region. Within the command, two coordinate pairs are given to specify the rectangle whose content is to be moved, and one more coordinate pair is given to specify the lower left corner of a new rectangle into which this content is to go. The new rectangle will have the same dimensions as the original one. Undesirable results may be produced if the rectangles overlap.

If there is already something on display in the rectangle to which a display content is moved, what is already on display there will be "overwritten". The use of Complement Mode would allow both the rectangle's old content and its new content to be seen.

Syntax: **!MOVe** x_1,y_1,x_2,y_2,x_3,y_3<CR>

Where: (x_1,y_1) is the coordinate pair representing the point at the lower left corner of the rectangle whose content is to be moved

(x_2,y_2) is the coordinate pair representing the point at the upper right corner of the rectangle whose content is to be moved

(x_3,y_3) is the coordinate pair representing the point at the lower left corner of the rectangle into which the content is to be moved

NOTE: Moves to the left or down from the orignal behave as expected when overlap occurs. Right and up do not.

The coordinate scaling factor selected with the Shrink Command should be taken into account when evaluating the coordinates in this command.

Example:

!MOV50,50,150,150,200,50<CR> The terminal is commanded to move the display within the rectangle defined by (50,50) and (150,150) to the rectangle defined by (200,50) and (300,150).

RASter Command - Transferring Graphics Data from the Host to the Terminal

This command is used when pixel information is to be transferred from the host directly to the high resolution display memory. Following the command word, the host supplies a string of bytes in ASCII code representing graphics memory data. The terminal extracts the three-bit pointers (or parts of pointers) which represent pixels from the incoming stream of bytes according to the specifications made in the Bits Command.

The graphics memory is loaded by placing received data at successive memory registers, beginning at the current beam position. The succession of registers corresponds to the movement of the graphics beam as its x and y coordinates are incremented. The beam moves horizontally from the beginning position to the right edge of the graphics region. When it reaches the right edge of the region, it is repositioned at the left edge of the region on the next higher horizontal line of pixels, etc. The movement of the graphics beam continues in this manner while pixel data are supplied. When it reaches the top right corner of the region it is repositioned at the bottom left corner of the region, etc.

The number of bytes is required to fill an entire graphics region varies with the size of the region defined with the Graphic Command and with the setting made with the Bits Command. The maximum size of the viewable region in 4027 Mode operation is 80 text columns by 33 text lines or 560 pixels in the x-direction by 264 pixels in the y-direction. This is equivalent to 147,840 pixels. (A larger region can be defined, in which case memory is allocated and "swept" for the entire region, but only part of the region is visible.)

The amount of pixel data which may be transmitted in a single Raster Command is limited by various factors, including baud rate and the specifications made with the Bits command. At 4800 baud and two pixels per byte (!Bit3) the recommended maximum amount of data is that for 32,767 pixels.

Syntax: **!RAS**ter **data<CR>**

Where: **data** is a string of ASCII characters representing pixel data encoded in accordance with specification made with the Bits Command.

Repeated bytes can be compacted with a positive integer; for example, AAAAAAAAAAAA is sent as 12(A). Only one level of parentheses can be used when compacting data.

The command character must not be one that will occur as video data.

Example:

Assume that the command **!BIT4<CR>** has been issued, indicating that each byte will contain data for one pixel. Assume also that the graphics beam has been positioned at absolute coordinates (0,100) with the command **!VEC0,100**<CR>. Finally, assume that graphics data are to be supplied which will result in the display of a yellow line from this point to the point (200,100).

Since the drawing color is yellow, color register C4 (default content yellow) will be used. The three-bit memory pointer for this register is 011 (from the chart on page 4.102). Since the !BIT4<CR> Command has been issued this pointer will be supplied in a byte of the form:

01 011 011

The hexadecimal equivalent for this byte is 5B, and the corresponding ASCII character is [, as shown in Appendix A. Thus the character [will represent one pixel in the line to be drawn.

The line to be drawn must contain 200 yellow pixels, because absolute or 2427 screen coordinates, in which x and y values directly represent pixels, are being used. (If 4027 or 4010 coordinates were in use, the number of pixels in a horizontal line from (0,100) to (200,100) would have to be calculated by multiplying the delta-x value by the appropriate scaling factor. See the chart on page 4.27.)

Accordingly, the command

!RAS200([)<CR>

would be issued by the host to cause the drawing of the line.

REAdback Command - Transferring Graphics Data to the Host

This command is used when contents of the high resolution display memory are to be transferred to the host. In response to the command, the terminal takes data from successive video memory registers and packs them into bytes for transmission according to specifications made with the Bits Command.

Data transfer starts at the memory register corresponding to the current graphics beam position. The transfer proceeds as the beam's x-coordinate is incremented. When the end of a line is reached, the y-coordinate is incremented and the x-coordinate is set to zero. The x-coordinate is then incremented again, etc. The data transfer continues until the number of pixels specified in the command has been sent or the top right corner of the display has been reached.

> Note: Because the execution of the command involves the incrementing of graphics region coordinates, it is relative to the specified coordinate shrink factor.

Upon receipt of a Readback Command, the terminal goes into a bypass condition; further transmission from the host is ignored until any ASCII control code is received. If Trailer Codes have been specified they will be appended to the string of bytes representing the graphics data.

The number of pixels to be "read back" to the host is specified in the command. At 4800 baud and two pixels per byte, the maximum number of pixels which may be specified is 32,767. If the host requires information on more than this number it must issue the Readback Command more than once.

Syntax: **!REA**dback **count<CR>**

Allowable parameter values:

count: positive integer = The number of pixels to be sent to the host, beginning at the current beam position.

Example:

Assume that <CR><LF> has been specified as a trailer sequence, that a graphics region has been defined with the command **!GRA1,30<CR>**, that a coordinate scaling factor of 1 has been set with **!SHR D<CR>**, and that some display has been created in the lower part of the region defined by the points (0,0) and (560,50).

Let the display be a horizontal yellow line with end points (5Ø,25) and (1ØØ,25).

The host could request a readback of pixel data in this area of the region with the command sequence:

```
!VECØ,Ø!BIT3!REA2856Ø<CR>
```

These commands request data for 2856Ø (51x56Ø) pixels at the rate of two pixels per byte, beginning at (Ø,Ø). The terminal would respond by going into a bypass condition and sending the following sequence:

```
7Ø25(@)25([)C7229(@)<CR><LF>
```

Where: 7Ø25(@) = 7Ø25 successive bytes, each having a binary value of Ø1ØØØØØØ. This is equivalent to 14Ø5Ø pixels (25 lines of 56Ø pixels plus 5Ø pixels) with pointers ØØØ (C7-default black).

25([) = 25 successive bytes, each having a binary value of Ø1Ø11Ø11. This is equivalent to 5Ø pixels with pointers Ø11 (C4-default yellow).

C = one byte having a binary value of Ø1ØØØØ11. This is equivalent to two pixels, one with pointer ØØØ (C7-default black) and one with pointer Ø11 (C4-default yellow). The yellow pixel comes before the black.

7229(@) = 7229 successive bytes, each having a binary value of Ø1ØØØØØØ. This is equivalent to 14458 pixels (458 pixels plus 25 lines of 56Ø pixels) with pointers ØØØ (C7-default black).

COMMAND GROUP H: COMMANDS FOR PROGRAMMING KEYS AND MACROS

All keys programmable on the Tektronix 4Ø27 can be programmed and used with the 2427 terminal when it is operating in 4Ø27 Mode. These keys then give ready access to as many as 185 special functions (or a total of 1Ø24 bytes, the amount of memory allocated for storage of keystroke sequences and macros). The commands in this group are involved in the programming and use of these special keystroke sequences and macros.

The commands in this group include:

- LEArn
- EXPand
- CLEar

LEArn Command

The Learn Command is used both to assign special sequences or strings to keyboard keys and to define macro commands which are executable by means of the Expand Command.

Note: The keyboard function keys retain their standard functions; they cannot be assigned sequences with the Learn Command.

Within the command, the key or macro command to be defined is identified, and one or more strings are then assigned. A string may consist of printable ASCII characters or the ADE for an ASCII character or control code.

To execute a programmed ASCII key, press the appropriate key.

Syntax: **!LEArn identifier**{**,,string**$_1$**, ... ,string**$_n$**,**}**<CR>**

Allowable parameter values:

identifier: M1-M16 =One of 16 macros to be defined.

printable ASCII character = The character whose key is to be programmed

Ø-127 = The ADE for a printable character or control code whose key is to be programmed.

string: A string may consist of a delimited sequence of printable ASCII characters if specified by its ADE value. Control characters must be specified by ADE.

Example:

Each of the following commands program the "#" key to initialize a workspace of 33 lines (text from the keyboard):

```
!LEA #,/!wor33,k/13

!LEA35,33,/wor33,k/13

!LEA35,33,119,111,114,32,51,51,44,107,13
```

Note that in each case the <CR> which terminates the command is given as an ADE.

EXPand Command

The Expand Command is used to initiate execution of a Macro Function defined with the Learn Command.

Syntax: **!EXP**and **macro**<**CR**>

Allowable parameter values:

macro: M1-M16 = One of 16 macros, previously defined with the Learn Command.

CLEar Command

The Clear Command is used to cancel definitions made with the Learn Command. All key sequences and macros defined with Learn are cleared.

Syntax: **!CLE**ar<**CR**>

COMMAND GROUP I: COMMANDS FOR PRODUCING HARDCOPY OUTPUT

The commands in this group include:

- HCOpy
- LPRinter

(At present the 2427 is not equipped with software which would allow these commands to be executed. The commands are included here for reference, with the expectation that future versions of the terminal will be equipped with this software.)

HCOpy Command

This command would be issued to cause a dump of high resolution screen data to a printer or other device connected at the terminal's auxiliary data port. At present the command is not implemented in software.

LPRinter Command -

This command would be issued to set transmission and data format parameters for transfers of high resolution screen data to an auxiliary device through the terminal's auxiliary data port. At present this command is not implemented in software.

Syntax: **!LPR**rinter {**pr**} {**sq**} {**rot**} {**rdc**} {**line**} {**ff**}<CR>

Allowable parameter values:

pr:	1-25 =	Printer number
sq:	D =	Direct
	S =	Square (compensation)
rot:	H =	Horizontal
	V =	Vertical
rdc:	F =	Full size
	R =	Reduced size
line:	C =	Computer (host)
	K =	Keyboard
	E =	Keyboard or host
	N =	Graphics only
ff:	L =	Leading form feed
	T =	Trailing form feed

no value = no change (for all parameters)

PROGRAMMING THE HOST

The following 4027-MODE demonstration program in BASIC code creates a bar graph using the information given in the problem statement. The same sequence of graphics commands can be coded in FORTRAN, PL/1, or any other high-level language supported by the user's host computer system. Figure 4-9 illustrates the completed bar graph as it appears on the terminal screen. Coordinate values refer to the standard Tektronix 4027 display (640 by 480).

PROBLEM: Draw a bar graph comparng the net sales of each quarter in 1977 with the net sales of each corresponding quarter in 1976. Input sales figures for 1977 are, in thousands of dollars: first quarter 210; second quarter, 240; third quarter, 150; and fourth quarter, 260. Figures for 1976 are: 180, 260, 130, 200 respectively.

```
100   REM
110   REM        4027-MODE DEMONSTRATION PROGRAM
120   REM
130   REM        ERASE THE SCREEN
140   PRINT "!ERA"
150   REM        DEFINE WORKSPACE
160   PRINT "!WOR 30"
170   REM        DEFINE GRAPHICS REGION
180   PRINT "!GRA 1,30"
190   REM        SET VECTOR DRAWING COLOR
200   PRINT "!COL C0"
210   REM        DRAW X AND Y AXES
220   PRINT "!VEC 50,350 50,50 600,50"
230   REM        SET FILL, BOUNDARY COLORS FOR 1977 BARS
240   PRINT "!COL C1,C0"
250   REM        DRAW 1977 BARS
260   PRINT "!POL 100,50 100,260 135,260 135,50"
270   PRINT "!POL 200,50 200,290 235,290 235,50"
280   PRINT "!POL 300,50 300,200 335,200 335,50"
290   PRINT "!POL 400,50 400,310 435,310 435,50"
300   REM        SET FILL, BOUNDARY COLORS FOR 1976 BARS
310   PRINT "!COL C5,C0"
320   REM        DRAW 1976 BARS
330   PRINT "!POL 135,50 135,230 170,230 170,50"
340   PRINT "!POL 235,50 235,310 270,310 270,50"
350   PRINT "!POL 335,50 335,180 370,180 370,50"
360   PRINT "!POL 435,50 435,250 470,250 470,50"
370   REM        RESET VECTOR DRAWING COLOR
380   PRINT "!COL C0"
390   REM        DRAW Y AXIS TIC MARKS
400   PRINT "!VEC 40,50 50,50"
410   PRINT "!VEC 40,100 50,100"
420   PRINT "!VEC 40,150 50,150"
430   PRINT "!VEC 40,200 50,200"
440   PRINT "!VEC 40,250 50,250"
450   PRINT "!VEC 40,300 50,300"
460   REM        LABEL Y AXIS
470   PRINT "!VEC 35,50  !STR/0/"
```

```
480   PRINT "!VEC 20,150  !STR/100/"
490   PRINT "!VEC 20,250  !STR/200/"
500   PRINT "!VEC 20,350  !STR/300/"
510   PRINT "!VEC 0,380  !STR/$ THOUSANDS/"
520   REM       LABEL X AXIS
530   PRINT "!VEC 110,40  !STR/1ST QTR/"
540   PRINT "!VEC 200,40  !STR/2ND QTR/"
550   PRINT "!VEC 310,40  !STR/3RD QTR/"
560   PRINT "!VEC 410,40  !STR/4TH QTR/"
570   REM       LABEL OVERALL GRAPH
580   PRINT "!VEC 135,0  !STR/NET SALES COMPARISON:  1976-1977/"
590   REM       SET COLORS, DRAW AND LABEL 1977 KEY IN LEGEND
600   PRINT "!COL C1,C0"
610   PRINT "!POL 520,335 520,350 545,350 545,335"
620   PRINT "!VEC 555,340  !STR/1977/"
630   REM       SET COLORS, DRAW AND LABEL 1976 KEY IN LEGEND
640   PRINT "!COL C5,C0"
650   PRINT "!POL 520,310 520,325 545,325 545,310"
660   PRINT "!VEC 555,320  !STR/1976/"
670   REM       LABEL LEGEND
680   PRINT "!VEC 520,360  !STR/LEGEND/"
690   REM       DRAW BOX AROUND LEGEND
700   PRINT "!COL C0"
710   PRINT "!VEC 495,370 495,300 595,300 595,370 495,370
720   END
```

PART FIVE

PART FIVE: OPERATION IN THE 4Ø1Ø MODE

In 4Ø1Ø Mode operation the 2427 emulates a Tektronix 4Ø1Ø Computer Display Terminal™. Commands formed according to 4Ø1Ø protocol should be correctly performed when recieved.

The use of 4Ø1Ø Mode for graphics display has an advantage deriving from the fact that 4Ø1Ø Mode graphics commands are formed with many fewer characters than are equivalent 4Ø27 or ANSI type commands. 4Ø1Ø Mode commands require very little time to transmit. The encoding required to form 4Ø1Ø type graphics commands is easy for a host but cumbersome for a human operator; accordingly, 4Ø1Ø Mode graphics displays are normally created by host programs rather than from the keyboard.

It is possible to transmit 4Ø1Ø type commands from the keyboard, however, and it might be desirable to do so when an operator or programmer is trying to familiarize himself with this mode of operation. In order to transmit commands from the keyboard it is necessary to select local echo while the terminal is in Setup Mode or 4Ø27 Mode and then enter 4Ø1Ø Mode. It should be kept in mind that the terminal will be operating in an on-line mode and that keystrokes echoed locally will also be transmitted to the host unless it is disconnected. If the host is disconnected, a loopback connector (jumper plug) may be installed at the terminal's host I/O connector as an alternative to the use of local echo.

4Ø1Ø Mode is normally entered from 4Ø27 Mode, after setting workspace, graphics region and coordinate scaling parameters with 4Ø27 Mode commands. It is not possible to create 4Ø1Ø Mode displays until these parameters have been evaluated.

4Ø1Ø MODE DISPLAY CAPABILITIES

In 4Ø1Ø Mode operation the terminal screen can represent both graphs and text. The display area cannot be divided to serve more than a single logical function at a given time. Normally the display area is used as a workspace for applications.

Displays of graphs are created with pixel addressing techniques. In relation to 4Ø1Ø Mode graphics commands the screen represents a coordinate system with limits defined by the points (Ø,Ø) and (1Ø23,779). Characters are displayed in dot matrices. In relation to text commands the screen represents a page having 35 lines and 8Ø columns of character cells. A page of text can be displayed in two columns, as is the case with the Tektronix 4Ø1Ø.

Graphic displays may consist of plotted points and lines. Various line types are available. The drawing colors must be selected in 4Ø27 Mode.

A cursor marks the position where text is entered into the display when the terminal has received a 4010 Mode text command. The cursor is nondestructive; it can be moved over existing graphic and alphanumeric elements without affecting them.

> Note: Setup Mode Main Menu Item H (Cursor Type) should be used to turn off the low resolution cursor when the terminal is in 4010 Mode operation. Otherwise two cursors will be displayed.

Characters are defined relative to a 14 x 22 dot matrix, but are displayed on a 7 x 8 matrix. The character set used to display text is selected in 4027 Mode and may be either Font 0 (ASCII dot matrix font) or Font 1 (APL dot matrix font).

4010 MODE COMMUNICATIONS PROTOCOL

Transitions to and from 4010 Mode are fully described in Part Two of this manual. Note that the <GS> control code is used both to enter 4010 Mode from Transparent Mode and to introduce 4010 Mode vector commands. Accordingly, if the terminal is in Transparent Mode and receives this code it can then receive the argument of a vector command immediately; <GS> does not have to be sent twice. However, if the first command to be executed in 4010 Mode is not a vector command, the <GS> code must be immediatedly followed by the control code which introduces the first command and must be sent again to introduce a subsequent vector command.

At baud rates of 1200 or more, remote start/stop (XON/XOFF) should be used to prevent input buffer overflow and consequent loss of data. Overflows are possible since 4010 Mode commands can contain long argument strings which require a good deal of time to process.

Command Structure

All 4010 Mode functions are initiated with commands introduced by ASCII control codes. Except for the <GS> code, the ASCII control codes which introduce 4010 Mode commands are not recognized by the terminal except when it is in 4010 Mode.

A string of printable ASCII characters immediately following the command introducer constitutes the argument of the command. This argument may represent coordinates to be plotted or a text to be displayed. The size of the string varies depending on the command.

A command is terminated by a control code introducing a subsequent 4010 Mode command or by a terminal control code.

Representation of Coordinates in 4Ø1Ø Mode Graphics Commands

The 4Ø1Ø Mode display can represent an (x,y) coordinate system with limits defined by the points (Ø,Ø) and (1Ø23, 779). This system is based on the resolution of the Tektronix 4Ø1Ø display: 1Ø24 pixels horizontally by 784 pixels vertically. The arguments of 4Ø1Ø Mode plot commands must represent coordinates relative to this system. These coordinates are automatically scaled to 2427 screen coordinates when graphic displays are created on the screen.

In 4Ø1Ø Mode plot commands, x and y coordinates are given in binary form. Ten binary digits are allotted to represent each coordinate. Each ten-digit coordinate is broken down into two groups of five digits. Then, each five-digit group is given a two-bit tag to identify it. The four resulting seven-bit codes can then be represented as four ASCII characters. Within a plot command, these characters are always placed in an order fixed according to the coordinates they represent: high-order y, low-order y, high-order x, low-order x.

The encoding process is given in schematic form below:

```
                          x-coord                  y-coord

Decimal                     563                      133

Binary                  1ØØØ111111               ØØ1ØØØØ1Ø1
                             |                        |
                       -------------            -------------
                       |           |            |           |
                     1ØØØ1       11111        ØØ1ØØ       ØØ1Ø1

Add tags          Ø1 1ØØØ1    1Ø 11111    Ø1 ØØ1ØØ    11 ØØ1Ø1
                  --------    --------    --------    --------
                      |           |           |           |
ASCII chars           1           _           $           e

Reorder chars:     y-high       y-low      x-high       x-low
                      $           e           1           _
```

In accordance with this example, the (x,y) coordinate pair (563,133) would be sent as the string **$el_** following a graphics command introducer.

When a graphics command includes more than one coordinate pair the characters representing the coordinates are included as a continuous string. When successive coordinate pairs are representable by four-character groups in which one or more characters do not change, it is possible to omit certain characters thereby compacting the argument string. Compacting is possible because of the two-bit tags which belong to the binary codes for the coordinates. The following rules must be observed.

1) When the x-low character is the only one that changes, it is the only one which must be sent.

2) When the x-high character changes, the y-low, x-high and x-low characters must be sent.

3) When the y-low character changes, the y-low and x-low characters must be sent.

4) When the y-high character changes, the y-high and x-low characters must be sent.

For example, the string

```
!h!H !w!H !w!W !h!W !h!H
```

may be compacted to the string

```
!h!H  w H    W  h W    H
```

In this example, spaces are used between certain characters to separate groups representing coordinate pairs and to indicate the characters deleted in the compacting process. These spaces would not be inserted into a string actually being transmitted.

Note: If the host system is not able to use the <DEL> character as a y-low value, the sequence <ESC>? can be used instead.

The ASCII coordinate codes are usually generated by a graphics application program. However, when graphics commands are to be entered from the keyboard the encoding process must be done by an operator. In this case it usually most convenient to work with decimal numbers. A procedure, with example, is given below for finding the four ADEs which would correspond to a coordinate pair expressed in decimal form.

1) Represent each coordinate as a decimal number.

   (x,y) = (500,250)

2) Divide each coordinate by 32 to obtain a seven-digit decimal number in the form: nn.mmmmm

   $nn_x.mmmmmx = x/32$ 500/32 = 15.62500 $nn_x = 15$
   $mmmmm_x = .62500$

   $nn_y.mmmmm_y = y/32$ 250/32 = 07.81250 $nn_y = 7$
   $mmmmm_y = .81250$

3) Add 32 to nn_x to get the ADE for x-high. Add 32 to nn_y to get the ADE for y-high.

   x-high (ADE) = nn_x+32 x-high = 15+32 = 47

   y-high (ADE) = nn_y+32 y-high = 7+32 = 39

4) Multiply $mmmmm_x$ by 32 and add 64 to the product to get the ADE for x-low. Multiply $mmmmm_y$ by 32 and add 96 to the product to get the ADE for y-low.

   x-low = $(mmmmm_x*32)+64$ x-low = (.62500*32)+64 = 84

   y-low = $(mmmmm_y*32)+96$ y-low = (.81250*32)+96 = 122

5) Place the four ADE values in the order y-high, y-low, x-high, x-low and substitute ASCII characters for their ADEs.

y-high	y-low	x-high	x-low
39	122	47	84
`	z	/	T

To find the ASCII characters which correspond to the ADE values calculated, it is necessary to refer to an ASCII code table. Such a table is included as Appendix A to this manual. The appendices also include a table constructed from the procedure given above which allows the conversion of decimal coordinates to ASCII characters directly (Appendix E).

PREPARING FOR GRAPHICS AND TEXT

Definitions and settings that affect the 4Ø1Ø Mode display are made both before and after entry into 4Ø1Ø Mode.

Communications settings (baud rate, echo, etc.) must be made in Setup Mode. The GIN source must also be selected in Setup Mode.

A workspace, graphics region and shrink factor must be defined with the following commands in 4Ø27 Mode before entry into 4Ø1Ø Mode:

!WOR34!GRA1,35!SHR Y<CR>

Several additional settings, like drawing color, character font, etc. must be made in 4Ø27 Mode. The following definitions and selections carry over from 4Ø27 to 4Ø1Ø Mode:

- Color settings (Color Command)
- Line type* (Line Command)
- User-defined line (DFLine Command)
- Complement Mode (CMode Command)
- Font** (Attribute Command)

*The last line type (1-9) selected in 4Ø27 Mode carries over to 4Ø1Ø Mode, except that line types P (point) and E (erase) do not carry over.

**Font Ø (ASCII dot matrix) and Font 1 (APL dot matrix) are carried over; if a vector character font was selected in 4Ø27 Mode, the ASCII dot matrix font carries over to 4Ø1Ø Mode.

4Ø1Ø Mode Command for Line Type Selection

Once the terminal is in 4Ø1Ø Mode, the line type used for drawing vectors is either the type previously selected in 4Ø27 Mode or a type selected with this command.

Syntax: **<ESC>type**

Allowable parameter values:

type: ` = ________________ (same as #1 in 4Ø27 Mode)
a = ... (same as #2 in 4Ø27 Mode)
b = . ___ . ___ . ___(same as #4 in 4Ø27 Mode)
c = _ _ _ _ _ _ _ _ _ _(same as #3 in 4Ø27 Mode)
d = ___ ___ ___ ___ (same as #7 in 4Ø27 Mode)
e = _ ___ _ ___ _ ___(same as #5 in 4Ø27 Mode)
f = ___...___...___..(same as #6 in 4Ø27 Mode)
g = ___ _ _ _ ___ _ _(same as #8 in 4Ø27 Mode)
x = Line defined with DFLine Command in 4Ø27 Mode (same as #9 in 4Ø27 Mode)

Example:

<ESC>a This command causes the terminal to draw vectors using dotted lines

DRAWING GRAPHICS

When initial settings have been made, graphics can be drawn by entering either of the following functional submodes:

- Point Plot Mode
- Vector Mode

Upon entry into one of these submodes the graphics beam is positioned at point (Ø,Ø) of the coordinate system represented by the display (lower left corner of the screen). Following plotting or vector drawing, the beam position is the point on the display at which execution of the last command terminated.

The beam position is not normally indicated by a visible mark (cursor). However, a crosshair cursor can be displayed at the beam position in response to a special host command sequence described below.

Use of the Crosshair Cursor to Mark the Graphics Beam Position

After it has issued a graphics command, the host can cause the terminal to display a crosshair cursor at the beam position by issuing the following sequence.

<ESC><SUB><ESC>/f
^B

<ESC><SUB> turns on the crosshair cursor and <ESC>/f moves the cursor to the current graphics beam position. If this sequence is followed by other commands, the cursor dissappears. The sequence must be reissued each time the cursor is to be displayed.

Note that <ESC><SUB> by itself is also used to enter GIN Mode. (See below.)

Point Plot Command

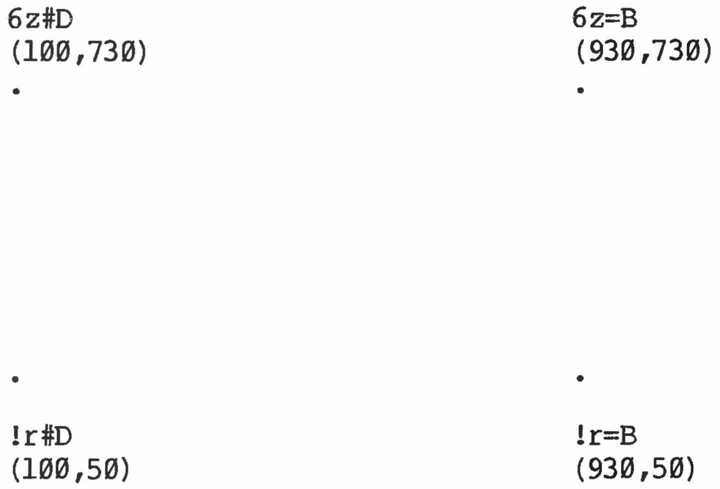
Syntax: **<FS>string$_1${string$_2$ ... string$_n$}**

Where a **string** is a group of ASCII characters representing an (x,y) coordinate pair. Normally, a string contains four characters representing **y-high**, **y-low**, **x-high** and **x-low** values, in that order. However, a string compacted. For the formation of strings, see the section above on the representation of coordinates in 4Ø1Ø Mode commands.

When the terminal is in 4Ø1Ø Mode and receives the <FS> code, it interprets succeeding ASCII character codes as representing (x,y) coordinate pairs. The first pair received is not plotted, but the graphics beam is moved to the point represented. Subsequent pairs are plotted as received. The argument string may represent any number of coordinate pairs. The command is terminated by the introducer of another 4Ø1Ø Mode command or by a terminal control code.

Example:

<FS>!r#D!r#D6z#D6z=B!r=B



Vector Plot Commands

Syntax: <GS>**string**$_1${**string**$_2$ ... **string**$_n$}

Where a **string** is a group of ASCII characters representing an (x,y) coordinate pair. Normally, a string contains four characters representing **y-high, y-low, x-high** and **x-low** values, in that order. However, a string compacted. For the formation of strings, see the section above on the representation of coordinates in 4010 Mode commands.

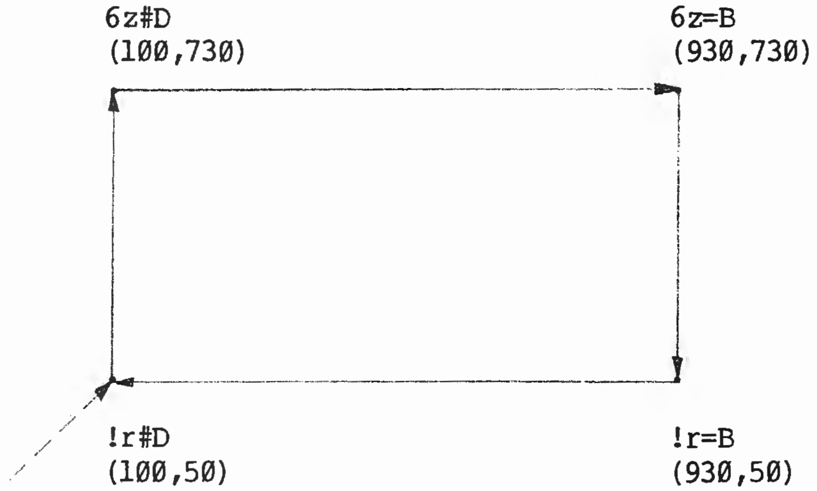
Upon receipt of a <GS> control character, the terminal interprets subsequently received printable ASCII characters as representing (x,y) coordinate pairs.

The first set of coordinates following the <GS> determines the starting point of the first line to be drawn. A line is drawn from there to the point indicated by the next set of coordinates, etc. as coordinates are received. When disconnected lines are to be drawn, separate <GS> sequences must be used. When a single coordinate pair follows <GS> the graphics beam is moved to the point indicated but no drawing is done.

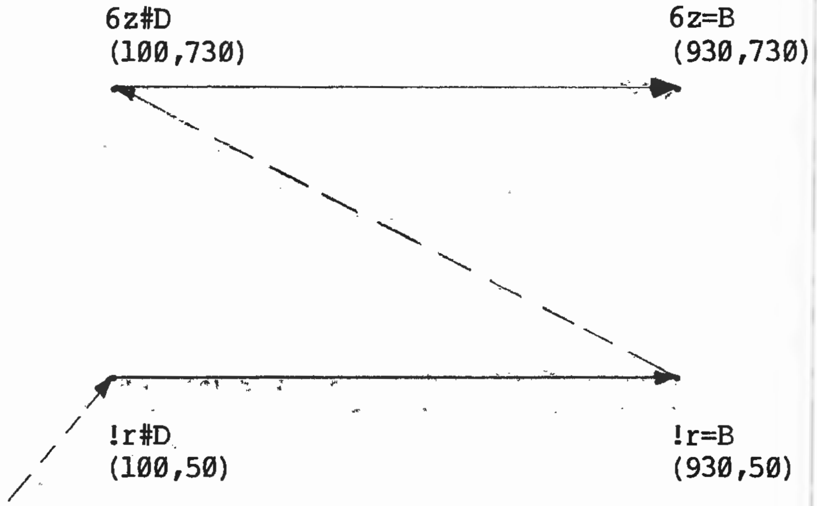
The command is terminated upon receipt of the introducer of another 4010 Mode command or of a terminal control code.

Examples:

<GS>**!r#D6z#D6z=B!r=B!r#D**



<GS>!r#D!r=B<GS>6z#D6z=B



TEXT COMMANDS

Once a text mode has been entered, any printable ASCII characters received by the terminal are added to the graphics display beginning at the cursor location. Text is normalized to a grid of character cells (35 lines, 8Ø cells per line).

When a text line is filled (the right margin of a line is reached), subsequently received characters are automatically displayed from the left margin of the next line (automatic wrap).

ASCII control codes for cursor movement may be inserted into the text stream and will not cause the termination of a text mode command.

For a two column display, text is entered beginning at the cursor "Home" postion in strings of 39 characters or less, followed by <CR><LF> codes and is justified along the left margin (margin Ø). After the 35th line of the first column has been entered, a carriage return <CR> and line feed <LF> positions the cursor at the top of Margin 1 (center top of the page, or line one, column 41) to start the second column. Margin 1 automatically becomes the left margin for the second column of text. Text in the first column that extends past Margin 1 is overwritten by text in the second column. When the cursor is on the 35th line of the second column a <CR><LF> returns it to the "Home" position (top line, margin Ø).

The various text commands permit the cursor to be assinged an initial position in different ways.

Command for Entering Text From the "Home" Position

Syntax: <**ESC**><**FF**>{**string**}

Where a **string** consists of printable ASCII characters which constitute a text.

When the terminal is in 4010 Mode and receives <ESC><FF>, its screen is cleared and a cursor is placed at the "Home" position (upper left corner) on the screen to indicate the point at which a text display will begin. Subsequently received codes for printable ASCII characters are treated as text.

ASCII control codes for cursor movement (<BS>, <CR>, <HT>, <LF> and <VT>) may be inserted into the text string for proper text formatting.

A text command is terminated by the introducer of another 4010 Mode command or by a terminal control code.

> Note: Certain keyboard operations have the same effect as the receipt of <ESC><FF> from the host. Operation of the (Command - Erase Page) key combination clears the screen and places the terminal in text mode with the cursor at "Home". The screen may also be cleared by entering Setup Mode and selecting Erase Graphics on the Graphics Submenu.

Command for Entering Text from the Left Margin of the Line Occupied by the Graphics Beam

Syntax: <CR>{**string**}

Where a **string** consists of printable ASCII characters which constitute a text.

When the terminal is in a 4Ø1Ø Graphics Mode and receives a <CR> code, a text cursor is placed at the left margin, across from the current graphics beam position and subsequently received codes for printable ASCII characters cause these characters to be displayed from the cursor position.

ASCII control codes for cursor positioning (<BS>, <CR>, <HT>, <LF> and <VT>) may be inserted into the text stream.

This text mode is terminated upon receipt of the introducer of another 4Ø1Ø Mode command or of a terminal control code.

Command for Entering Text from the Current Graphics Beam Position

Syntax: <US>{**string**}

Where a **string** consists of printable ASCII characters which constitute a text.

When the terminal is in a 4010 graphics mode and receives a <US> control code, a cursor is placed at the character cell occupied by the graphics beam, and text is entered into the display from that point.

ASCII control codes for cursor movement (<BS>, <CR>, <HT>, <LF> and <VT>) may be inserted into the text string.

This text mode is terminated when the introducer of another 4010 Mode command or a terminal control code is received.

Example:

This command is useful for placing text in relation to graphs already on display when a coordinate pair can be specified to indicate the beginning point of the text. For example, the following commands move the graphics beam to the point (495,390), near the center of the screen, and enter the word "TEXT" beginning at the character cell in which this point falls:

```
<GS>,f/0
<US>TEST
```

ERASING GRAPHICS AND TEXT

The entire screen may be cleared by the operator from the keyboard using the (Command - Erase Page) key combination, without taking the terminal out of 4010 Mode. The screen may also be cleared from the keyboard entering Setup Mode, then selecting the Erase function from the Graphics Submenu and returning to 4010 Mode. The terminal sends the XOFF code to the host when Setup Mode is entered and the XON code when returned to 4010 Mode.

The host can clear the screen by sending the sequence <ESC><FF>.

Individual graphics elements can be erased by redrawing them in the black. To change the drawing color to black it is necessary to go to 4027 Mode and issue a Color Command to select the color register containing the code for black. Once the color has been set, 4010 Mode must be reentered and the proper graphics commands given.

Like graphics, text can be erased by re-entering the characters in the black. In this case, 4027 Mode should be entered and an Attribute Command should be issued to set the color for text to black. 4010 Mode should then be reentered and the proper text commands issued.

Note that the use of this technique can create breaks where display elements to be erased intersect with other display elements.

USING GRAPHICS INPUT (GIN) MODE

Before using the Graphics Input Mode in 4010 Mode operation it is sometimes necessary to go to 4027 Mode to set the GIN source (Input Command) and to set Trailer Codes (Trailer Command). The GIN source is the keyboard unless some other source has been specified. (Optional input devices such as the light pen can be used in 4010 GIN Mode.)

Once settings have been made and the terminal is in 4010 Mode, the receipt of the following control sequence causes entry into GIN Mode:

<ESC><SUB>

Following receipt of this sequence, a crosshair cursor is displayed. The operator may then use the keyboard Arrow keys (or other device) to position the cursor. Tapping an arrow key moves the crosshair one pixel at a time; holding it down moves the cursor continuously.

When the operator has positioned the cursor where he wants it, he may press any alphanumeric key (or operates the switch on an optional device) to position the graphics beam at the crosshair and send a beam position report to the host.

The beam position report consists of a string of five to seven bytes of ASCII code, each byte representing an ordered part of the report as follows:

chr x-high x-low y-high y-low {**trailer**$_1$}{**trailer**$_2$}

Where: **chr** is the ASCII code corresponding to the key pressed

x-high is Ø 1 x_9 x_8 x_7 x_6 x_5

x-low is Ø 1 x_4 x_3 x_2 x_1 $x_Ø$

y-high is Ø 1 y_9 y_8 y_7 y_6 y_5

y-low is Ø 1 y_4 y_3 y_2 y_1 $y_Ø$

trailer is an optional ASCII code, usually a control code

Note that the bytes representing the beam coordinates include tag bits B_6 and B_5 which are set to Ø 1. The values of these bits differ from the values of the tag bits in the corresponding bytes which are sent as graphics command arguments.

Trailer codes are included in the report if they have been selected with the 4Ø27 Mode Trailer command. Trailer codes are normally ASCII control codes and are included to permit the host to terminate GIN Mode and perhaps to perform some other function when it echoes these codes back to the terminal.

While in GIN Mode the terminal is in a Bypass Condition, in which it ignors data from the host until it receives any control code. Accordingly, GIN data, if echoed, would not be accepted by the terminal, but trailer control codes, if subsequently echoed, would be acted upon. The Bypass Condition is automatically cleared upon receipt of an echoed control code. If this control code is a 4Ø1Ø Mode command introducer, GIN Mode is cleared and the mode indicated by the introducer is entered.

For example, if a <CR> is used as a Trailer in a GIN report to the host, and this trailer is echoed by the host, the terminal will enter text mode following transmission of the report, and the text cursor will be positioned at the left margin of the line occupied by the graphics beam.

If trailer codes are not used, the terminal remains in GIN Mode after the transmission of a beam position report and until it receives some control code from the host.

If the display is cleared with the (Command - Erase Page) keys or the Setup Mode Erase Graphics function while the terminal is in 4Ø1Ø GIN Mode, GIN Mode is terminated and trailer codes, if selected, are transmitted but no beam coordinates are transmitted.

The terminal may respond to a status request from the host while it is in GIN Mode. After placing the terminal in GIN Mode, the host may request a report instead of waiting for keyboard input from the terminal indicating cursor/beam position. If the cursor is moved by the operator, but no alphanumeric key is struck, and if the terminal subsequently receives a status request, a report of the cursor position will be transmitted to the host and the graphics beam position will not be modified to agree with the cursor position. GIN Mode will be cleared. For further information, see the section below.

REPORTING STATUS TO THE HOST

The host may request a 4010 Mode status report with the command sequence:

<ESC><ENQ>

The terminal's response to this sequence will vary depending on the 4010 submode (a text mode, graphics mode or GIN mode) in which it is operating at the time.

- If the terminal is in a text submode, the status report will be a string of bytes of the form:

  status x-high$_{TC}$ **x-low**$_{TC}$ **y-high**$_{TC}$ **y-low**$_{TC}$ {**trailer**$_1$}{**trailer**$_2$}

  The coordinates represented are those of the point in the lower left corner of the character cell occupied by the text cursor (TC).

- If the terminal is in a graphics submode (except GIN mode), the status report will be a string of bytes of the form:

  status x-high$_{GB}$ **x-low**$_{GB}$ **y-high**$_{GB}$ **y-low**$_{GB}$ {**trailer**$_1$}{**trailer**$_2$}

  The coordinates represented are those of the graphics beam position (GB).

- If the terminal is in GIN mode, the status report will be a string of bytes of the form:

  x-high$_{CC}$ **x-low**$_{CC}$ **y-high**$_{CC}$ **y-low**$_{CC}$ {**trailer**$_1$}{**trailer**$_2$}

  The coordinates represented are those of the crosshair cursor (CC).

When a **status** byte is included in the report it has the following composition:

Bit	Setting	Meaning
7	Ø	none
6	Ø	Status byte identifier
5	1	Status byte identifier
4	1	none (for a Tektronix terminal this is the hardcopy unit bit, Ø = hardcopy unit available)
3	Ø	none (for a Tektronix terminal this is the vector bit, 1 = vector mode exists)
2	Ø/1	Graph Mode Bit (Ø = Any Graph Mode, 1 = Any Text Mode)
1	Ø/1	Margin Bit (Ø=Margin Ø set, 1=Margin 1 set)
Ø	1	none (for a Tektronix terminal this is the aux unit sensing bit; Ø = some aux device other than a hardcopy unit is connected to the terminal and is functional)

The bytes representing coordinates include tag bits (B_6 and B_5) set to Ø 1.

Trailer codes are included if they have been selected with a 4Ø27 Mode Trailer Command.

After receiving an <ESC><ENQ> sequence, the terminal enters a Bypass Condition, in which it ignors data from the host until it receives any control code. This condition prohibits the status data from being accepted as graphics coordinates or text if echoed back by the host. The host system automatically clears the Bypass Condition if it echoes a trailer code which is a control code.

PRODUCING HARDCOPY OUTPUT

(This section remains incomplete pending finalization of system hardware and software design.)

Several printer option settings, defined in 4027 Mode, affect hardcopy output. These settings include identification of the printer model, choice of two I/O channels, proportional compensation (needed with some printers), image rotation, full or reduced image, leading or trailing form feed, and alphanumeric printing from the host and/or keyboard. (See the LPRint Command, described in Part Four, Command Group I.)

The host can initiate a transfer by sending the sequence

<ESC><ETB>

PROGRAMMING THE HOST

Most existing applications programs designed to supply data in the form of 4Ø1Ø commands will require no modification for use with the 2427 terminal. New programs may be written to make use of the 4Ø1Ø Mode functions described in the preceding sections. The following demonstration programs, one written in FORTRAN code, the other written in BASIC, show how terminal commands for drawing a simple rectangle can be generated. The rectangle has corners at points with coordinates (2ØØ,3ØØ), (2ØØ,4ØØ), (6ØØ,4ØØ), and (6ØØ,3ØØ).

```
1ØØ C
11Ø C     DEMONSTRATION FORTRAN PROGRAM
12Ø C
13Ø       CALL CLR
14Ø       CALL MOV (2ØØ,3ØØ)
15Ø       CALL DRW (2ØØ,4ØØ)
16Ø       CALL DRW (6ØØ,4ØØ)
17Ø       CALL DRW (6ØØ,3ØØ)
18Ø       CALL DRW (2ØØ,3ØØ)
19Ø       STOP
2ØØ       END
21Ø C
22Ø C     END MAIN PROGRAM
23Ø C
24Ø       SUBROUTINE MOV (X,Y)
25Ø C
26Ø C     MOVES GRAPHICS BEAM TO POINT (X,Y) 4Ø1Ø COORDINATES
27Ø C
28Ø       INTEGER X,Y
29Ø       CALL PUTC (29)
3ØØ C
31Ø C     SENDS `GS` CHARACTER (29) FOR ENTRY OF 4Ø1Ø VECTOR MODE
32Ø C
33Ø       CALL DRW (X,Y)
34Ø C
35Ø C     ENCODES AND SENDS (2ØØ,3ØØ) TO POSITION GRAPHICS BEAM
36Ø C
37Ø       RETURN
38Ø       END
39Ø C
4ØØ C
41Ø       SUBROUTINE DRW (X,Y)
42Ø C
43Ø C     ENCODES COORDINATES (X,Y)
44Ø C     ACCORDING TO 4Ø1Ø PROTOCOL AND
45Ø C     SENDS CODE TO TERMINAL
46Ø C
47Ø       INTEGER X,Y
48Ø       IHY = ((Y/32) .AND. 31)+32
49Ø       ILY = (Y .AND. 31)+96
5ØØ       IHX = ((X/32) .AND. 31)+32
51Ø       ILX = (X .AND. 31)+64
```

```
520        CALL PUTC (IHY)
530        CALL PUTC (ILY)
540        CALL PUTC (IHX)
550        CALL PUTC (ILX)
560        RETURN
570        END
580 C
590 C
600        SUBROUTINE PUTC (IC)
610 C
620 C      SENDS ASCII BIN EQU OF INTEGER C TO THE TERMINAL
630 C
640 C      IF IT WOULD BE MORE EFFICIENT TO TRANSMIT CHARATERS TO
650 C      THE TERMINAL IN BLOCKS, A SUBROUTINE OF THIS SORT COULD
660 C      BE USED TO BUFFER CHARACTERS.  ANOTHER SUBROUTINE COULD THEN
670 C      BE INSERTED AT THE END OF THE MAIN PROGRAM TO DUMPT THE
680 C      CONTENTS OF THE BUFFER TO THE TERMINAL
690 C
700        INTEGER IC
710        WRITE (7,100) IC
720    100 FORMAT (`+`,A1)

730 C      A `+` CARRIAGE CONTROL IS NEEDED
740 C      ON SOME COMPUTERS TO STRIP THE `CR` AND `LF` CHARACTERS
750 C
760        RETURN
770        END
780 C
790 C
800        SUBROUTINE CLR
810 C
820 C      CLEARS THE SCREEN BY SENDING `GS``ESC``FF` TO THE TERMINAL
830 C
840        CALL PUTC (29)
850        CALL PUTC (27)
860        CALL PUTC (12)
870        RETURN
880        END
```

An equivalent BASIC program is given below.

```
100 REM  DEMONSTRATION BASIC PROGRAM
110 REM
120 GOSUB 360
130 X=200:Y=300:GOSUB 190
140 X=200:Y=400:GOSUB 260
150 X=600:Y=400:GOSUB 260
160 X=600:Y=300:GOSUB 260
170 X=200:Y=300:GOSUB 260
180 END
190 REM SEND `GS` TO INITIATE 4010 VECTOR MODE
200 REM THEN CALL DRAW TO ENCODE FIRST COORDINATE PAIR
210 REM AND SEND CODE TO TERMINAL TO MOVE GRAPHICS BEAM
220 REM
230 PRINT CHR$ (29);
240 GOSUB 260
250 RETURN
260 REM SUBROUTINE DRAW
270 IHY=((Y/32) AND 31)+32
280 ILY=(Y AND 31)+96
290 IHX=((X/32) AND 31)+32
300 ILX=(X AND 31)+64
310 PRINT CHR$ (IHY);
320 PRINT CHR$ (ILY);
330 PRINT CHR$ (IHX);
340 PRINT CHR$ (ILX);
350 RETURN
360 REM
370 REM SUBROUTINE CLEAR
380 REM CLEARS SCREEN BY SENDING `GS`, `ESC`, `FF`
390 PRINT CHR$ (29);
400 PRINT CHR$ (27);
410 PRINT CHR$ (12);
420 RETURN
```

PART SIX

PART SIX: OPERATION IN THE TRANSPARENT MODE

THE 2427 COMPARED WITH THE 24Ø5

When the 2427 is placed in the Transparent Mode, its operation is similar to that of the Intecolor 24Ø5 terminal. Either ANSI X3.64 type commands or DEC VT52 type commands are acceptable, as determined in Setup Mode.

In this mode of operation, the 2427's graphics command processor (GCP) is idle, except that it examines the incoming data stream for mode change control sequences before passing text and Transparent mode commands between the host and the VT52/ANSI X3.64 command processor (and, by extension, between the host and the keyboard). The terminal's high resolution display memory and display generator are also idle in this mode. Displays are generated using character cell techniques only.

Note: Those who are familiar with the operation of the 24Ø5 terminal should note the following difference between 24Ø5 operation and 2427 Transparent Mode operation: The ASCII <GS> control code, sent to the 2427 by the host, causes the terminal to leave the Transparent Mode and enter 4Ø1Ø Mode. The 24Ø5 performs no special function upon receipt of this code.

TRANSPARENT MODE DISPLAY CAPABILITIES

The Transparent Mode display is generated using character cell techniques by themselves; it can represent either a page of text having 24 lines and 8Ø characters per line, or a graph constructed of plot blocks on an (x,y) coordinate system in which x is an integer between Ø and 159 and y is an integer between Ø and 95.

Eight background and eight foreground colors are selectable on a character cell by character cell basis. Color control is possible with ANSI sequences and with the FG/BG key, used in combination with keys Q, W, E, R, T, Y, U and I. (See the SGR sequences described in this chapter and the keyboard description in Part One for further information.)

The character cells in which characters and plot blocks are represented resolve to a 6 x 12 pixel matrix. The size of the matrix permits the display of the ASCII characters with standard resolution and, in addition, the display of those lower case characters which have descenders in their proper relations to the other characters.

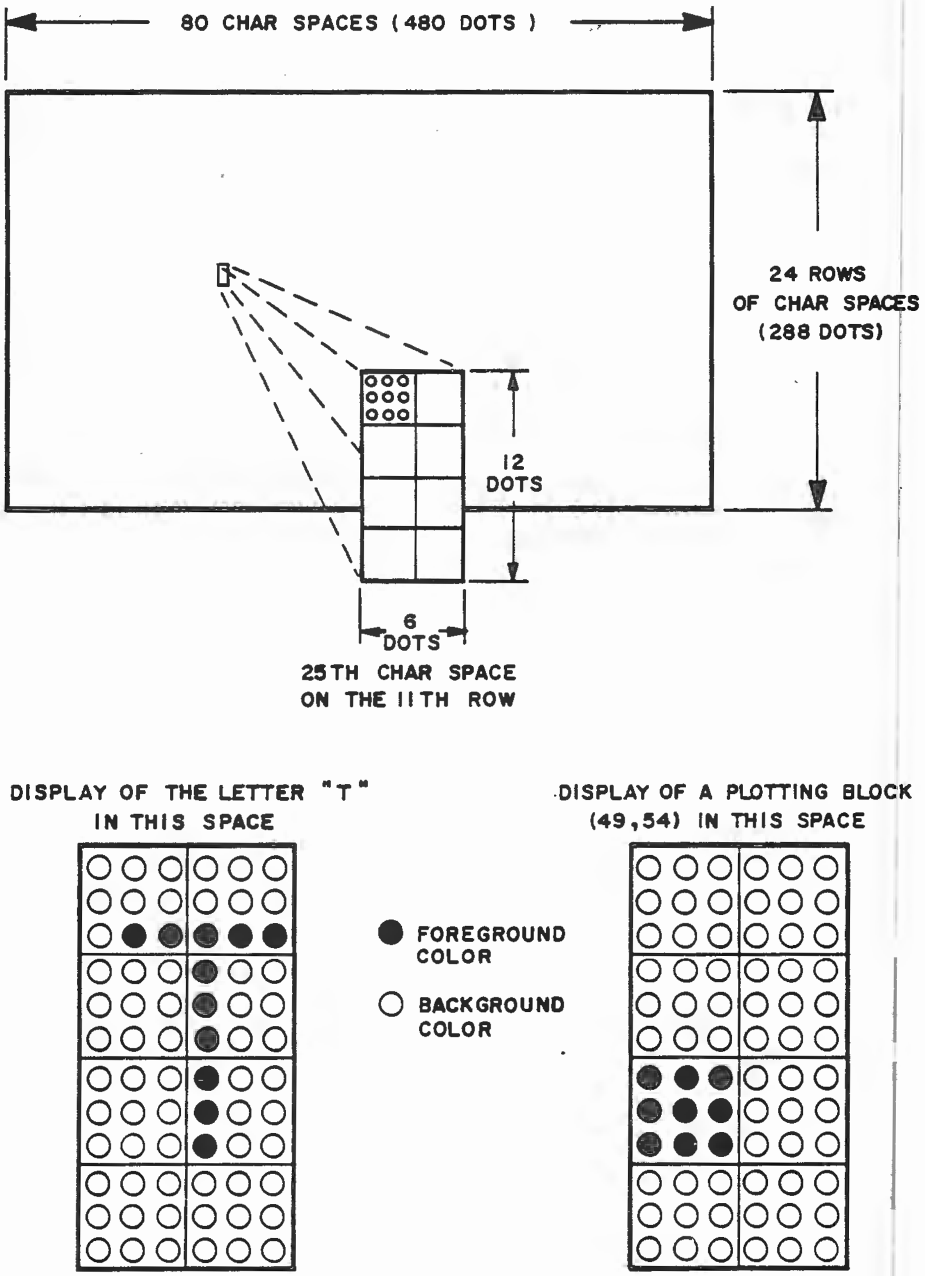
The character set available in Transparent Mode includes the 96 uppercase and lowercase ASCII characters, 32 control code symbols and 32 special graphics characters. The complete character set is represented on a fold-out chart (drawing #1Ø23Ø7) included in the appendices.

The control code symbols are displayable only following the issuance of a special ANSI control sequence. When the terminal is in the Control Representation Mode both ASCII control codes and longer control sequences are represented on the screen, but these codes and sequences are not otherwise acted on by the terminal processors. (See the CRM sequences.)

Switching between the ASCII lowercase characters and the special graphics characters is done in response to <SO> and <SI> control codes, issued by the host or from the keyboard (when local echo is selected). The ASCII uppercase characters are always available. (See the ANSI SCS sequences for further information.)

Graphs constructed of plot blocks can be displayed in response to ANSI command sequences which cause points and lines to be drawn. (See the PLOT commands.) Plot blocks resolve to 3 x 3 dots, and there can be up to eight blocks in a character cell. Blocks can be plotted one at a time, but they are handled by the display generator as groups of one to eight, like characters. Accordingly, when a new block is to be displayed within a given character cell, the contents of the entire cell are changed. For example, if a group of two blocks is already displayed in the given cell it is replaced with a group of three blocks having the proper positions relative to one another. This means that the colors of the blocks are not indepentently controllable.

Transparent Mode Display Organization



ANSI CONTROL SEQUENCES

When operating in the Transparent Mode, the 2427 responds to control sequences formed either according to the VT52 protocol or according to the ANSI (American National Standards Institute) X3.64 standard. VT52 or ANSI X3.64 operation is selected by the operator when the terminal is in the Setup Mode. (See Part Two for details.) ANSI sequences are described in this section of the manual. The VT52 commands are described in a subsequenct section.

The ANSI standard allows commands to include arguments of varying lengths. The arguments are interpreted according to final character in the command sequence. This command format is very flexible. On the one hand, it permits the command set to be expanded without extensive rewriting of terminal software. On the other hand, it permits the sending of commands to a terminal which is not equipped to execute them. Such commands are merely ignored by the terminal.

Command Structure

The structure of an ANSI command is as follows: The command begins with a Control Sequence Introducer. The introducer is followed by a variable number of numeric arguments (parameter values), set off by semicolons. These values are followed by a final character which indicates the operation to be performed.

- The control sequence introducer (CSI) is an ASCII Escape code followed by a left bracket: **<ESC>[**

- A numeric parameter value is always in the range Ø-255 decimal and is given as a string of ASCII numeric characters. Numeric parameters include such things as x and y coordinates.

- A selective parameter value is a special type of numeric value. Selective parameters distinguish subfunctions within the general functions indicated by the command's final control character.

- Parameter values in a sequence are separated by semicolon delimiters only. No delimiter is given between the control sequence introducer and the first parameter value. No delimiter is given between the last parameter value and the final control character.

- The control character is an ASCII upper case or lower case alphabetic character or punctuation mark other than the semicolon. Such a character always terminates the command sequence when received following the CSI. The control character indicates a major terminal function, in terms of which the preceeding parameter values are interpreted. The number of parameters evaluated in a command sequence depends on this character.

<u>Note</u>: An ASCII control code may be embedded in an ANSI command sequence. If the terminal receives a control code following the reception of the CSI and before the reception of the ANSI control character, the control code is treated as not belonging to the sequence and is acted upon as usual; the ANSI sequence is acted upon subsequently and when it has been completely received.

Examples of Valid ANSI Command Sequences

```
                  <ESC>[ 3 A
                  ______ _ _
                     |    | |
                     |    | |
      CSI---------------  | |
                          | |
          Numeric value-------  |
                            |
              Control Char-------
```

This command would cause the terminal to move the cursor up 3 lines. The "3" is interpreted as a number of lines in relation to the "A", which indicates cursor movement upwards.

```
                  <ESC>[ 32 m
                  ______ __ _
                     |     | |
                     |     | |
      CSI---------------   | |
                           | |
          Selective value------  |
                             |
              Control char--------
```

This command would cause the terminal to display subsequently received text with the foreground color green. Note that "32" indicates foreground display in green only in relation to "m". Foreground display in green is treated here as a subfunction of a general function called "select graphic rendition".

```
                     <ESC>[ 2 ; 3 H
                     ______ _ _ _ _
                       |    | | | |
                       |    | | | |
CSI---------------------    | | | |
                            | | | |
  Numeric value--------------  | | |
                               | | |
    Delimiter-------------------  | |
                                  | |
      Numeric value---------------  |
                                    |
        Control Char-----------------
```

This command would cause the terminal's cursor to move to line 2, column 3. Note that the final character "H" indicates cursor movement in terms of lines and columns; accordingly, it determines the number of numeric values which are included.

A Note on Erroneous Commands

Whenever the terminal receives the control sequence introducer <**ESC**>[it treats every subsequently received byte of code other than an ASCII control code as an element of a command. The first alphabetic or punctuation code received following the introducer other than the semicolon code is always treated as the command terminator. A terminated command is either ignored completely (if the terminator does not indicate a function implemented in software) or executed as sent.

If a command sequence does not specify values for all applicable parameters, the terminal supplies default values for those missing from the command. If the command includes too many values, the extra values are ignored. If a semicolon delimiter is omitted, the two numbers which should have been delimited are treated as a single number. If this number is greater than 255, it is treated as 255.

The commands are listed alphabetically by mnemonic.

CBT Cursor Backward Tabulation- HOST to Terminal and Terminal to HOST

<ESC> [Pn Z default value: 1

The Cursor Backward Tabulation (CBT) command sequence moves the current cursor position horizontally in a backward direction to the previous tabulation stops. No parameter value, a parameter value of zero or a parameter value of one moves the current cursor position to the first previous tabulation stop. A paramter value of larger than one moves the current cursor position backward that many tabulation stops.

CHT Cursor Horizontal Tabulation- HOST to Terminal and Terminal to HOST

<ESC> [Pn I default value: 1

The Cursor Horizontal Tabulation (CHT) command sequence moves the current cursor position horizontally in a forward direction to the next tabulation stops. No parameter value, a parameter value of zero or a parameter value of one moves the active position to the next tabulation stop. A parameter value of larger than of one moves the current cursor position forward that many tabulation stops.

CPR Cursor Position Report- Terminal to HOST

<ESC> [Pn ; Pn R

The Cursor Position Report (CPR) sequence reports the current cursor postion to the host computer. The sequence has two parameters, the first specifying the current line number and the second specifying the current column. Both lines and columns are numbered from zero with home position denoted by both parameters being zero.

This control sequence is sent in response to a device status report sent from the host (see DSR, this section).

CRM Control Representation Mode- HOST to Terminal

<ESC> [3 h - to set Mode (SM)

<ESC> [3 l* - to reset Mode (RM)

The Control Representation Mode (CRM) is a parameter that is controlled by the Set Mode (SM) and Reset Mode (RM) ANSI sequences. The reset state causes terminal control commands to be executed and to have no graphic representation in the display. The set state keeps controls from being executed, but instead displays a graphic code for each character received.

*Note: The final character in this sequence is a lower case L.

CTC Cursor Tabulation Control- HOST to Terminal

<ESC> [Pn W   default value: Ø

The Cursor Tabulation Control (CTC) command sequence clears or sets one or more tabulation stops according to the parameter(s). When horizontal tabulation stops are set or cleared, the effect may be only on the active line or it may be on all lines depending upon the state of the Tabulation Stop Mode (TSM). (Refer also to TBC)

Pn	Action
Ø	Sets a horizontal tabulation stop at the current cursor postion (default)
1	Sets a vertical tabulation stop at the current line
2	Clears the horizontal tabulation stop at the current cursor postion
3	Clears the vertical tabulation stop at the current line
4	Clears all horizontal tabulation stops in the current line
5	Clears all horizontal tabulation stops
6	Clears all vertical tabulation stops

CUB Cursor Backward- HOST to Terminal and Terminal to HOST

<ESC> [Pn D   default value: 1

The Cursor Backward (CUB) command sequence moves the current cursor position to the left. The parameter determines the number of positions to be moved. If the parameter is one or zero, the command moves the current cursor position one position to the left. If the parameter value is greater than one, the command moves the current cursor position a distance equal to the parameter. Once the current position reaches the left margin, it moves no further.

CUD Cursor Downward- HOST to Terminal and Terminal to HOST

<ESC> [Pn B   default value: 1

The Cursor Downward (CUD) command sequence moves the current cursor position downward while maintaining the same column position. The parameter determines the number of positions to be moved. If the parameter is one or zero, the command moves the current cursor position one line downward. If the parameter value is greater than one, the command moves the current cursor position a distance equal to the parameter. Once the current position reaches the bottom margin, it moves no further.

CUF Cursor Forward- HOST to Terminal and Terminal to HOST

<ESC> [Pn C   default value: 1

The Cursor Forward (CUF) command sequence moves the current cursor position to the right. The parameter determines the number of positions to be moved. If the parameter is one or zero, the command moves the current cursor position one position to the right. If the parameter value is greater than one, the command moves the current cursor position a distance equal to the parameter. Once the current position reaches the right margin, it moves no further.

CUP Cursor Position- HOST to Terminal and Terminal to HOST

<ESC> [Pn ; Pn H default value: 1

The Cursor Position (CUP) command sequence moves the current cursor position to the position specified by the parameters. The two parameters determine the new position, with the first parameter specifying the line position and the second parameter specifying the column position. If either parameter is one or zero, the command moves the current cursor position to the first line or column respectively. The default position with no parameters is equivalent to a Home Cursor action.

This command is sent by the terminal with the appropriate parameters when the light pen is detected.

CUU Cursor Up- HOST to Terminal and Terminal to HOST

<ESC> [Pn A default value: 1

The Cursor Up (CUU) command sequence moves the current cursor position upward while maintaining the same column position. The parameter determines the number of positions to be moved. If the parameter is one or zero, the command moves the current cursor position one line upward. If the parameter value is greater than one, the command moves the current cursor position a distance equal to the parameter. Once the current position reaches the top margin, it will move no further.

CVT Cursor Vertical Tabulation- HOST to Terminal and Terminal to HOST

<ESC> [Pn Y default value: 1

The Cursor Vertical Tabulation (CVT) command sequence moves the current cursor position downward to the next or following in a series of vertical tabulation stops without altering the horizontal position. No parameter value, a parameter value of zero or a parameter value of one moves the active position to the next line with a vertical tabulation stop. A parameter value of larger than of one moves the active position forward that many tabulation stops.

DA Device Attributes-

<ESC> [Pn c	HOST to Terminal	default value: Ø
<ESC> [? 1 ; Ø c	Terminal to HOST	default value: Ø

If the host requests Device Attributes by sending the above HOST to Terminal sequence with no parameter or with a parameter value of zero, the Terminal will respond with the Terminal to HOST sequence above indicating an Intecolor 24ØØ with no special options present.

DSR Device Status Report- HOST to Terminal and Terminal to HOST

<ESC> [Pn n default value: Ø

The Device Status Report (DSR) command sequence when sent from the HOST, requests either a report of the general status of the terminal or a Cursor Position Report (CPR), depending upon the value of the parameter. This sequence when sent from the terminal, reports the general status of the terminal, depending upon the value of the parameter.

Pn	Action or Meaning
Ø	Reports terminal is ready, no malfunctions
3	Reports terminal is malfunctioning
5	Requests a Status Report (using DSR)
6	Requests a Cursor Position Report (CPR)

ED Erase In Display- HOST to Terminal

<ESC> [Pn J   default value: Ø

The Erase In Display (ED) sequence erases some or all of the characters in the display according to the value of the parameter.

Pn	Action
Ø	Erases from the active position to the end of the display screen, inclusive
1	Erases from the start of the display screen to the active position, inclusive
2	Erases all of the display screen

EL Erase In Line- HOST to Terminal

<ESC> [Pn K   default value: Ø

The Erase In Line (EL) sequence erases some or all of the characters in the current line according to the value of the parameter.

Pn	Action
Ø	Erases from the active position to the end of the current line, inclusive
1	Erases from the start of the current line to the active position, inclusive
2	Erases all of the current line

FETM Format Effector Transfer Mode- HOST to Terminal

<ESC> [14 h - to Set Mode (SM)

<ESC> [14 l* - to Reset Mode (RM)

The Format Effector Transfer Mode (FETM) is a parameter that is controlled by the Set Mode (SM) and Reset Mode (RM) ANSI sequences. This parameter is the same parameter that is controlled by the Color Printer option in the SET-UP mode. The reset state allows the terminal to pass command sequences on to the auxiliary device. The set state causes the terminal to suppress sending command sequences on to the auxiliary device. (See Color Printer option in SET-UP Mode in Chapter 1.)

*Note: The last character in the sequence is a lower case L.

HTS Horizontal Tabulation Set- HOST to Terminal and Terminal to HOST

<ESC> H

The Horizontal Tabulation Set (HTS) command sequence sets one horizontal tabulation stop at the active position. This command sets the tabulation stop on the active line only or on all lines depending upon the state of the Tabulation Stop Mode (TSM).

HVP Horizontal & Vertical Position- HOST to Terminal and Terminal to HOST

<ESC> [Pn ; Pn f default value: 1

The Horizontal & Vertical Position (HVP) command sequence moves the current cursor position to the position specified by the parameters. The two parameters determine the new position, with the first parameter specifying the line position and the second parameter specifying the column position. If either parameter is one or zero, the command moves the current cursor position to the first line or column respectively. The default position with no parameters is equivalent to a Home Cursor action.

ED Erase In Display- HOST to Terminal

<ESC> [Pn J   default value: Ø

The Erase In Display (ED) sequence erases some or all of the characters in the display according to the value of the parameter.

Pn	Action
Ø	Erases from the active position to the end of the display screen, inclusive
1	Erases from the start of the display screen to the active position, inclusive
2	Erases all of the display screen

EL Erase In Line- HOST to Terminal

<ESC> [Pn K   default value: Ø

The Erase In Line (EL) sequence erases some or all of the characters in the current line according to the value of the parameter.

Pn	Action
Ø	Erases from the active position to the end of the current line, inclusive
1	Erases from the start of the current line to the active position, inclusive
2	Erases all of the current line

FETM Format Effector Transfer Mode- HOST to Terminal

<ESC> [14 h   - to Set Mode (SM)

<ESC> [14 l*   - to Reset Mode (RM)

The Format Effector Transfer Mode (FETM) is a parameter that is controlled by the Set Mode (SM) and Reset Mode (RM) ANSI sequences. This parameter is the same parameter that is controlled by the Color Printer option in the SET-UP mode. The reset state allows the terminal to pass command sequences on to the auxiliary device. The set state causes the terminal to suppress sending command sequences on to the auxiliary device. (See Color Printer option in SET-UP Mode in Chapter 1.)

*Note: The last character in the sequence is a lower case L.

HTS Horizontal Tabulation Set- HOST to Terminal and Terminal to HOST

<ESC> H

The Horizontal Tabulation Set (HTS) command sequence sets one horizontal tabulation stop at the active position. This command sets the tabulation stop on the active line only or on all lines depending upon the state of the Tabulation Stop Mode (TSM).

HVP Horizontal & Vertical Position- HOST to Terminal and Terminal to HOST

<ESC> [Pn ; Pn f   default value: 1

The Horizontal & Vertical Position (HVP) command sequence moves the current cursor position to the position specified by the parameters. The two parameters determine the new position, with the first parameter specifying the line position and the second parameter specifying the column position. If either parameter is one or zero, the command moves the current cursor position to the first line or column respectively. The default position with no parameters is equivalent to a Home Cursor action.

IND Index- HOST to Terminal and Terminal to HOST

<ESC> D

The Index (IND) command sequence moves the active position down one line without changing the horizontal position. If the active position is at the bottom margin, a Scroll Up (SU) is performed.

KAM Keyboard Action Mode- HOST to Terminal

<ESC> [2 h - to Set Mode (SM)

<ESC> [2 l* - to Reset Mode (RM)

The Keyboard Action Mode (KAM) is a parameter that is controlled by the Set Mode (SM) and Reset Mode (RM) ANSI sequences. The reset state enables the keyboard. The set state disables the keyboard. When the keyboard is disabled, the keyclick will not sound.

*Note: The last character in the sequence is a lower case L.

LNM Line Feed New Line Mode- HOST to Terminal

<ESC> [2Ø h - to Set Mode (SM)

<ESC> [2Ø l* - to Reset Mode (RM)

The Line Feed New Line Mode (LNM) is a parameter that is controlled by the Set Mode (SM) and Reset Mode (RM) ANSI sequences. The reset state causes the interpretation of the Line Feed (LF) to imply only vertical movement of the active position. The set state causes the Line Feed to imply movement to the first position of the following. This is the New Line option. This mode does not affect the Vertical Tabulation (VT), Form Feed (FF), Index (IND), or Next Line (NEL).

*Note: The last character in the sequence is a lower case L.

MC Media Copy- HOST to Terminal

<ESC> [Pn i   default value: Ø

The Media Copy (MC) command sequence initiates and controls the transfer of data from the terminal to the auxiliary device according to the value of the parameter.

Pn	Action
Ø	Copies the entire contents of the display screen to the auxiliary device
4	Turns off copying of received data to the auxiliary device (Terminates Media Copy)
5	Turns on copying of received data to the auxiliary device and turns off copying of received data to display screen
6	Turns off copying of received data to the auxiliary device (Terminates Media Copy)
7	Turns on copying of received data to the auxiliary device but data continues to be copied to display screen

During Media Copy (MC), if XON/XOFF is enabled the Intecolor 24ØØ sends XOFF and XON codes to synchronize the character rate from the host computer with the character rate of the auxiliary device.

NEL Next Line- HOST to Terminal and Terminal to HOST

<ESC> E

The Next Line (NL) command sequence moves the current cursor position to the first position on the next line downward.

NP Next Page- HOST to Terminal

<ESC> [U

The Next Page (NP) command sequence displays the second page of the display memory on the terminal screen. The cursor position, the current foreground and background colors and the vertical margins for the first page are saved in memory, and these same parameters for the second page become active. These parameters for the second page are saved when a Previous Page (PP) command is received.

PALN (Private) Screen Alignment Display- HOST to Terminal

<ESC> # 8

The Screen Alignment Display (PALN) command sequence causes the display page to be filled with the character which preceeds the cursor at the time the command is issued. If the cursor is in the HOME position, a white page is displayed. This command is used for maintenance and screen alignment.

PATM (Private) ANSI Terminal Mode - HOST to Terminal

<ESC> [? 2 h - to Set Mode (PSM)

<ESC> [? 2 l* - to Reset Mode (PRM)

The ANSI Terminal Mode (PATM) allows the host computer to specify whether the terminal will process the standard ANSI commands specified in this section of the manual or the alternate VT52 type commands specified in the following section. The set state specifies ANSI commands.

PAWM (Private) Auto Wrap Mode- HOST to Terminal

<ESC> [? 7 h - to Set Mode (PSM)

<ESC> [? 7 l* - to Reset Mode (PRM)

The Auto Wrap Mode (PAWM) allows the host computer to set or reset the Wrap Mode. This is the same feature that is set in the SET-UP mode. (see WRAP in SET-UP) In the set mode, the 81st and following characters of a line will be displayed on the following line. In the reset mode, the 81st and following characters of a line will be overlaid onto the 80th position of the current line.

PCDP (Private) Cursor Display Mode - Host to Terminal

<ESC> [> 5 h - to Set Mode (PSM) or blank cursor

<ESC> [> 5 l* - to Reset (PRM) or display cursor

Alternate sequences:

<ESC> [1 v - to blank cursor

<ESC> [Ø v - to display cursor

These command sequences are for use on terminals with V2.6 and subsequent software. Terminals with earlier versions ignore these sequences. When the cursor is blanked ("blind" cursor mode) it still exists for the software and can still be moved, etc. with standard cursor commands. The PCKP commands thus affect only the visibility of the cursor to the operator.

*Lowercase L.

PCKM (Private) Cursor Keys Mode

<ESC> [? 1 h - to Set Mode (PSM)

<ESC> [? 1 l* - to Reset Mode (PRM)

The cursor keys mode (PCKM) allows the host to specify which of two sets of character sequences will be sent by the terminal when the cursor positioning keys are pressed. In the set mode, the cursor positioning keys send application functions. In the reset mode, the cursor positioning keys send ANSI cursor control commands. (See Cursor Control Keys in Chapter 1.)

*Note: The last character in each sequence is a lower case L. In the PCKM sequence, the L is preceeded by a one.

PCNF (Private) Confidence Test- HOST to Terminal

<ESC> [Pn y

The Confidence Test (PCNF) command sequence invokes one or more of a set of tests to provide confidence of proper terminal operation. Some of these tests are also executed automatically at power-up (see **Self-Test** in chapter one).

The parameter specified in the command sequence determines the type of test to be executed. Combinations of the following tests may also be performed by summing the values shown. A value of more than 127 causes the specified test(s) to be performed repeatedly, either until failure or until power is turned off. The only other way to terminate such a repeating test is to hit the unshifted 'SETUP' key once. The terminal will then return to the normal terminal mode.

Pn	Test to Execute
Ø or 1	Read-Only-Memory test Setup Memory test, Screen Memory test, Host data loop-back test
2	Host Data loop back test
4	Keyboard/Printer interface loop-back test
128	Repeat selected test(s) until failure or power-off
255	Repeat **ALL** tests until failure or power-off

These tests can only indicate a so-called "non-fatal" error. Such an error implies that only a portion of the terminal has a problem, and that enough of the terminal's internal mechanism is still working to provide an error indication. No response can be guaranteed if "fatal" errors exist.

If any errors are found while executing a given test, the screen will be cleared and a single letter or symbol printed in the upper left-hand corner. The type of character printed will signify the nature of the problem, as follows:

ERROR	CODE PRINTED
Read-Only-Memory	@
Setup Memory	A
Screen Memory	C
Host I/O	G
Keyboard/Printer	O

Other letters or symbols indicate combinations of these errors. Such a case is treated more fully in the **Intecolor 24ØØ Maintenance Manual.**

PDFK (Private) Default Function Key Mode - HOST to Terminal

<ESC> [? 29 h - to Set Mode (PSM)

<ESC> [? 29 l* - to Reset Mode (PRM)

The Default Function Key Mode allows the host to specify whether function keys transmit their default values or their user-assigned values. The set state causes the function keys to transmit their default values. The reset state causes the function keys to transmit their user-assigned values. This mode does not alter the user definitions of function keys.

PFNR (Private) Function Key Rate Mode - Host to Terminal

<ESC> [? 31 h - to Set Mode (Fast Function Keys)

<ESC> [? 31 l* - to Reset Mode (Slow Function Keys)

These sequences allow the adjustment of the rate at which the characters belonging to sequences generated by function keys are transmitted to the host. In the slow mode, the characters in a function key sequence are transmitted at the rate of 6Ø per second. In the fast mode, the characters belonging to a function key sequence are transmitted as fast as the terminal's processing time and the baud rate will allow. Rate control is useful when a host application program uses the time between characters to distinguish text characters from characters in a command sequence.

PIDT (Private) Identify Terminal- HOST to Terminal

<ESC> Z

The Identify Terminal (PIDT) command sequence evokes the same response as the ANSI Device Attributes (DA) command sequence.

PKPA (Private) Keypad Application Mode- HOST to Terminal

<ESC> =

The Keypad Application Mode (PKPA) command sequence places the terminal in the Keypad Application Mode, which means that the auxiliary keypad keys transmit the control sequences defined in the table on page 1.17.

*Note: The last character in the sequence is a lower case L.

PKPN (Private) Keypad Numeric Mode- HOST to Terminal

<ESC> >

The Keypad Numeric Mode (PKPN) command sequence places the terminal in the Keypad Numeric Mode, which means that the numeric keypad keys transmit the ASCII codes that correspond to the characters on the keycaps.

PLOT (Private) Plot Graphics Commands- HOST to Terminal

<ESC> [Pn ; Pn ; Pn {; Pn} z

The Plot Graphics Commands (PLOT) provide you with the capability to plot points and vectors on the display screen. The basic form of the command is shown above. Up to four parameters may be present.

The first parameter specifies the type of plot to be done. There are three different plot commands, Absolute Point Plot, Relative Move, and Relative Draw. These commands represent the minimum considered necessary to easily create simple graphic displays. The three commands and their parameter values are as follows:

(1) Absolute Point Plot
(2) Relative Move
(3) Relative Draw

The second, third and fourth parameters are interpreted differently for each plot mode. Certain ranges of parameters are expected by each plot mode. If a parameter is out of range the mode is aborted and the command is ignored.

The screen can be viewed as a graph with an X-axis and a Y-axis. The X-axis runs along the bottom of the screen and the Y-axis runs along the left edge of the screen. Both the X and the Y coordinates must be positive integers. A specific location on the screen is expressed as an X,Y coordinate pair. The maximum resolution is 160 horizontal by 96 vertical points known as pixels. Horizontal coordinates are expressed as positive integers in the range 0...159, and vertical coordinates are expressed as positive integers in the range 0...95.

Absolute Point Plot

<ESC> [1 ; $Pn_{x\text{-coord}}$; $Pn_{y\text{-coord}}$ z

where:

$Pn_{x\text{-coord}}$ is in the range Ø...159

$Pn_{y\text{-coord}}$ is in the range Ø...95

The Absolute Point Plot command plots a point is at a specified location on the screen. The current foreground and background colors are used. If points with a different foreground or background color are already plotted in the same character position, they will be changed to the current foreground and background colors. When a point is plotted, an internally maintained reference point is defined as having the coordinates of the point just plotted. **Relative Move** and **Relative Draw** commands will be relative to this point, which will be referred to as the current plot position.

Relative Move

<ESC> [2 ; Pn_s ; Pn_{dx} ; Pn_{dy} z

where:

Pn_s has one of the following values:

```
    Ø  -  for both delta x and delta y signed positive
    1  -  for delta x signed positive and delta y negative
    2  -  for delta x signed negative and delta y positive
    3  -  for both delta x and delta y signed negative
```

Pn_{dx} (delta x) is in the range Ø...159

Pn_{dy} (delta y) is in the range Ø...95

The Relative Move command moves a specified distance on the screen from the current plot position, then redefines the current plot position as having the coordinates of the new location. Nothing is actually plotted in this mode; only the current plot position is changed. If the specified movement would place the current plot position outside the boundary of the screen, the command sequence is aborted, and the current plot position is not changed.

Relative Draw

<ESC> [3 ; Pn_s ; Pn_{dx} ; Pn_{dy} z

where:

Pn_s has one of the following values:
Ø - for both delta x and delta y signed positive
1 - for delta x signed positive and delta y negative
2 - for delta x signed negative and delta y positive
3 - for both delta x and delta y signed negative

Pn_{dx} (delta x) is in the range Ø...159

Pn_{dy} (delta y) is in the range Ø...95

The Relative Draw command draws a vector from the current plot position for a specified distance on the screen, then redefines the current plot position as having the coordinates of the new location. This mode redefines the position of the current plot position. If the specified movement would place the current plot position outside the boundary of the screen, the command sequence is aborted, and the current plot position is not changed.

PMRG (Private) Set Margins- HOST to Terminal and Terminal to HOST

<ESC> [Pn ; Pn r

The Set Margins (PMRG) command sequence sets the top and bottom margins of the scrolling region. The first parameter is the top limit and the second parameter is the bottom limit. Both parameters must be a numeric value between 1 and 24, inclusive, and the bottom limit must be at least one greater than the top limit. Once the scrolling region is set by this command, only the lines within the scrolling region will scroll, any lines above or below the scrolling region will remain on the display.

Messages to the operator may be displayed outside of the scrolling region by use of the CUP sequence. The cursor position within the scrolling region must first be saved with the Save Cursor Position (PCSP) command. It can then be restored to the previous position with the Restore Cursor Position (PRCP) command.

POM (Private) Origin Mode

<ESC> [? 6 h - to Set Mode (PSM)

<ESC> [? 6 l* - to Reset Mode (PRM)

The Origin Mode (POM) allows the host to specify where origin position is relative to the vertical margins. The reset state causes the origin to be the upper-left character position on the screen. The set state causes the origin to be at the upper left position within the margins. In the set state the cursor will not cross margin boundaries.

*Note: The last character in the sequence is a lower case L.

PP Previous Page- HOST to Terminal

<ESC> [V

The Previous Page (PP) command sequence displays the first page of the display memory on the terminal screen. The cursor position, the current foreground and background colors and the vertical margins for the second page are saved in memory, and these same parameters for the first page become active. These parameters for the first page are saved when a Next Page (NP) command is received.

PPFN (Private) Program Function Keys - Host to Terminal

<ESC> [Pn_{key} ; Pn_1 {; Pn_2 . . . ; Pn_n} s

where:

Pn_{key} is a decimal number which identifies the function key being programmed. The numbers used for the keys in these commands are as follows:

Key	Unshifted	Shifted	With Control
FØ - F23	Ø - 23	Unshifted value plus 32	Unshifted value plus 64

Pn_1 - Pn_n are the decimal equivalents of the ASCII codes for the alphanumeric characters and control symbols in the string assigned to the function key by the command. See Appendix A of this manual for a listing of the ASCII Decimal Equivalents (ADE). The string assigned to a function key may not contain more than 4Ø codes.

This command is used when the programmer wishes to have the host define the assignments to one or more of the function keys. A separate command sequence must be issued for each key definition.

The sequence <ESC> [s may be sent to the terminal from the host in order to clear all current function key assignments.

Note that if the function key assignments are cleared or if any function key is defined by the host with a sequence including parameter values, all previous assignments, or the previous assignment for a particular key will be destroyed, unless the assignments have been made by the terminal operator from the Setup Mode and have been stored in the terminal's permanent Setup Memory. (See page 18 of this manual for details.) If a previously defined assignment has been stored in permanent memory, that assignment may be restored by the operator following the use of these commands. The RIS sequence (ESC c) is used for restoration.

This command is executed by terminals with V2.6 and subsequent software only. Terminal's with earlier software ignore the command.

PRCP (Private) Restore Cursor Position– HOST to Terminal

<ESC> 8

The Restore Cursor Position (PRCP) command sequence restores the active cursor to the position (vertical and horizontal) previously saved with the PSCP command. Video attributes (background/foreground color, underline, and blinking character) and the GØ and Gl character set assignments (SCS commands) which were in effect at the time the position was stored are also restored.

PRKC (Private) Repeat Key Control - Host to Terminal

<ESC> [? 8 l* - inhibits arrow key auto repeat

<ESC> [? 8 h - enables arrow key auto repeat

The arrow keys are normally repeating keys, i. e., when held down they send their codes repeatedly. When used for scrolling in text processing, the use of these keys can cause buffer overflow unless their repeating capability is disabled. This command permits the repeat function to be switched in or out at will.

*Lower case L.

PRM (Private) Reset Mode- HOST to Terminal

<ESC> [? Pn l*

The Private Reset Mode (PRM) command sequence resets one or more of the private modes within the terminal as specified by the selective parameter. At this time only one mode is implemented within the Intecolor 24ØØ Private Modes. The remaining parameter values are for future expansion.

The valid selective value is:

Pn	Mode	Function
1	PCKM	Cursor Keys
2	PATM	ANSI Terminal Mode
6	POM	Origin Mode
7	PAWM	Auto Wrap capability
29	PDFK	Default Function Key Mode

These parameters are set by the Private Set Mode command (PSM).

*Note: The last character in the sequence is a lower case L.

PRPT (Private) Request/Report Terminal Parameters-

<ESC> [Pn x HOST to Terminal

<ESC> [$Pn_1;Pn_2;Pn_3;Pn_4;Pn_5;Pn_6;Pn_7$ x Terminal to HOST

The Request Terminal Parameters (first sequence above) is sent by the host computer with a parameter value of zero or one, which will request a response of Report Terminal Parameters from the terminal. Any parameter value other than zero or one causes the request to be rejected by the terminal. The Intecolor 2400 responds with a Report Terminal Parameters sequence (second sequence above). The five parameters inform the host computer of various parameter settings within the terminal. The meanings of the parameter values are included below:

Parameter Position	Value	Meaning
First	2	This message is a report
Second	1	No Parity Set
	4	Parity is Set and Odd
	5	Parity is Set and Even
Third	2	Seven Bits Per character
Fourth	000	50 Baud Transmit Rate
	008	75
	016	110
	024	134.5
	032	150
	040	200
	048	300
	056	600
	060	1050
	064	1200
	072	1800
	080	2000
	088	2400
	104	4800
	112	9600
	120	19200
Fifth	*	* Baud Receive Rate
Sixth	1	Bit rate multiplier is 16
Seventh	0	Flags (always 0)

*Note: The values of the fifth parameter are the same as those of the fourth.

PSCP (Private) Save Cursor Position- HOST to Terminal

<ESC> 7

The Save Cursor Position (PSCP) command sequence saves the current cursor position (vertical and horizontal) and video attributes (background/foreground color, underline, and blinking character) in memory, from which they may be recovered at a later time by the Restore Cursor Position (PRCP) command. The assignments to the GØ and Gl character sets, made with the (SCS) sequences, are also stored for later recovery with the (PRCP) command.

PSM (Private) Set Mode- HOST to Terminal

<ESC> [? Pn h

The Private Set Mode (PSM) command sequence sets one or more of the private modes within the terminal as specified by the selective parameter. At this time only one mode is implemented within the Intecolor 24ØØ Private Modes. The remaining parameter values are for future expansion.

The valid selective value is:

Pn	Mode	Function
1	PCKM	Cursor Keys
2	PATM	ANSI Terminal Mode
6	POM	Origin Mode
7	PAWM	Auto Wrap capability
29	PDFK	Default Function Key Mode

These parameters are reset by the Private Reset Mode command (PRM).

PU1 Private Use One - Host to Terminal/Terminal to Printer

<ESC> Q chr

This sequence, together with the SS2 and SS3 sequences, is used to transmit or receive plot block clusters. The last element of the sequence is an ASCII character corresponding to a specific plot block cluster. See below on the SS2 and SS3 sequences.

RI Reverse Index- HOST to Terminal and Terminal to HOST

<ESC> M

The Reverse Index (RI) command sequence moves the current cursor position to the same horizontal position on the preceding line. If the cursor position is at the top margin, a Scroll Down (SD) is performed.

RIS Reset to Initial State- HOST to Terminal

<ESC> c

The Reset to Initial State (RIS) command sequence resets the Intecolor 24ØØ to its power-on state. This command sequence clears all character buffers, unlocks the keyboard, and sets the video attributes to the default values (normal characters, green on black background). The terminal assumes the set up configuration stored in permanent memory.

RM Reset Mode- HOST to Terminal

<ESC> [Pn ; Pn ... Pn l*

The Reset Mode (RM) command sequence resets one or more of the ANSI modes within the terminal as specified by the selective parameter. Each mode to be reset is specified by a separate parameter. Up to nine parameters may be included in one command sequence. The valid selective values are:

*Note: The last character in the sequence is a lower case L.

Pn	Mode	Function
2	KAM	Keyboard Action Mode - if set, this mode locks the keyboard
3	CRM	Control Representation Mode - if set, this mode causes controls to have a graphic representation on the display screen
12	SRM	Send-Receive Mode - if reset, each character entered at the keyboard is displayed on the display screen (see Echo feature in SET-UP Mode)
14	FETM	Format Effector Transfer Mode - if reset, command sequences and control characters are included in the data stream passed on to the auxiliary device
18	TSM	Tabulation Stop Mode - if reset, the setting and clearing of horizontal tabulation stops apply to the corresponding character positions in all lines. If set, the horizontal tabulation stops apply to each line independently.
20	LNM	Line Feed New Line Mode - if reset, the linefeed implies only vertical movement of the current cursor position. If set, the linefeed implies movement to the first position of the following line.

These parameters are set by the Set Mode command (SM).

SCS Select Character Set-

This command assigns one of three character sets to the GØ character set or to the G1 character set according to the commands listed below. The SI (ØF Hex) and SO (ØE Hex) control codes select which of the two assigned character sets are active at any one time. The SI control code enables the GØ character set and the SO control code enables the G1 character set.

SCS Commands

GØ Sets	G1 Sets	Name
<ESC> (A	<ESC>) A	U.K. Set
<ESC> (B	<ESC>) B	ASCII Set
<ESC> (Ø	<ESC>) Ø	Special Graphics

SGR Select Graphic Rendition-

<ESC> [Pn ; Pn ... Pn m

The Select Graphic Rendition (SGR) command sequence invokes the graphic rendition specified by the parameter(s). All characters following this command in the data stream are rendered according to the selected graphic rendition until the next occurrence of the Select Graphic Rendition (SGR) command. Up to nine parameters may be sent in one command sequence.

Pn	Meaning
Ø	Primary Rendition (green letters on black background)
1	Bold Characters (white letters on black background)
2	Faint Characters (blue letters on black background)
4	Underscore
5	Blink
6	Blink
7	Reverse Image (background and foreground colors swapped)
8	Concealed Characters
3Ø	Black Foreground
31	Red Foreground
32	Green Foreground
33	Yellow Foreground
34	Blue Foreground
35	Magenta Foreground
36	Cyan Foreground
37	White Foreground
4Ø	Black Background
41	Red Background
42	Green Background
43	Yellow Background
44	Blue Background
45	Magenta Background
46	Cyan Background
47	White Background

SM Set Mode- HOST to Terminal

<ESC> [Pn ; Pn ... Pn h

The Set Mode (SM) command sequence sets one or more of the ANSI modes within the terminal as specified by the selective parameter. Each mode to be set is specified by a separate parameter. Up to nine parameters may be included in one command sequence. The valid selective values are:

Pn	Mode	Function
2	KAM	Keyboard Action Mode - if set, this mode locks the keyboard
3	CRM	Control Representation Mode - if set, this mode causes controls to have a graphic representation on the display screen
12	SRM	Send-Receive Mode - if reset, each character entered at the keyboard is displayed on the display screen (see Echo feature in SET-UP Mode)
14	FETM	Format Effector Transfer Mode - if reset, command sequences and control characters are included in the data stream passed on to the auxiliary device
18	TSM	Tabulation Stop Mode - if reset, the setting and clearing of horizontal tabulation stops apply to the corresponding character positions in all lines. If set, the horizontal tabulation stops apply to each line independently.
2Ø	LNM	Line Feed New Line Mode - if reset, the linefeed implies only vertical movement of the current cursor position. If set, the linefeed implies movement to the first position of the following line.

These parameters are reset by the Reset Mode command (RM).

SRM Send-Receive Mode- HOST to Terminal

<ESC> [12 h - to Set Mode (SM)

<ESC> [12 l* - to Reset Mode (RM)

The Send-Receive Mode (SRM) is a parameter that is controlled by the Set Mode (SM) and Reset Mode (RM) ANSI sequences. When the SRM is in the reset state there is a logical connection between the keyboard and the display screen which causes the terminal to display or echo each character or command that is entered at the keyboard. However, when the SRM is in the set state the keyboard and the display screen are logically independent from each other which causes the terminal to send the keyboard input to the host only, which means the host must echo anything that is to be displayed on the terminal.

*Note: The last character in the sequence is a lower case L.

SS2 Single Shift Two - Host to Terminal/Terminal to Printer

<ESC> N chr

This sequence (together with the sequences PU1 and SS3) is used to transmit and receive CRT displays composed of plot block clusters. The last element in the sequence is an ASCII character (2Ø Hex - 7E Hex). When this character is specified, the sequence defines one of the 256 clusters displayable in a character space on the screen. The table on the pages following gives the sequence assigned to each cluster.

When a PU1, SS2 or SS3 sequence is received at the terminal's Host port, the corresponding plot block cluster is displayed at the current cursor position with the current foreground and background colors.

When the contents of a page of display memory are transmitted to a printer using the Page Print key, every memory representation of a plot block cluster is converted to a PU1, SS2 or SS3 ASCII sequence before being transmitted. Accordingly, displays containing points and vectors can be reproduced on an appropriately equipped and programmed intelligent printer by using these sequences. In general, they should not be used when the printer connected does not respond to ANSI control sequences.

Note: The PU1, SS2 and SS3 sequences are interpreted and generated by terminals with V2.5 and subsequent software only.

SS3 Single Shift Three - Host to Terminal/Terminal to Printer

<ESC> O chr

This sequence (together with the PU1 and SS2 sequences) is used to transmit and receive CRT displays composed of plot block clusters. See the description above (SS2) and the pages following.

Table of PU1, SS2 and SS3 Sequences

In the tables below and on the following pages, each of the 256 displayable plot block clusters is correlated with a PU1, SS2 or SS3 sequence. To determine the sequence for a given cluster, first look to the top of the column in which it appears to find the sequence introducer (ESC N, ESC O or ESC Q). Then look to the left of the row in which it appears to find the ASCII character which closes the sequence.

	ESC N	ESC O	ESC Q		ESC N	ESC O	ESC Q		ESC N	ESC O	ESC Q
SP	[] [] [] [] [] [] [] []	█ █ █ [] █ █ █ []	[] █ █ █ █ [] █ █	+	█ [] █ [] [] [] █ []	[] [] █ █ [] █ █ []	█ [] [] [] [] █ █ █	6	[] █ █ [] █ [] [] []	█ █ [] █ █ █ [] []	[] █ [] [] █ █ [] █
!	█ [] [] [] [] [] [] []	[] [] [] █ [] █ [] []	█ █ █ █ █ [] █ █	,	[] [] [] [] █ [] █ []	█ [] █ █ [] █ █ []	[] [] █ [] [] █ █ █	7	█ █ █ [] █ [] [] []	[] █ █ █ █ █ [] []	█ █ [] [] █ █ [] █
"	[] [] █ [] [] [] [] []	█ [] [] █ [] █ [] []	[] [] [] [] [] █ [] █	-	█ [] [] [] █ [] █ []	[] [] [] █ █ █ █ []	█ [] █ [] [] █ █ █	8	[] █ [] [] [] [] █ []	█ █ █ █ █ █ [] []	[] █ █ [] █ █ [] █
#	█ [] █ [] [] [] [] []	[] [] █ █ [] █ [] []	█ [] [] [] [] █ [] █	.	[] [] █ [] █ [] █ []	█ [] [] █ █ █ █ []	[] [] [] [] █ █ █ █	9	█ █ [] [] [] [] █ []	[] █ [] █ [] █ █ []	█ █ █ [] █ █ [] █
$	[] [] [] [] █ [] [] []	█ [] █ █ [] █ [] []	[] [] █ [] [] █ [] █	/	█ [] █ [] █ [] █ []	[] [] █ █ █ █ █ []	█ [] [] [] █ █ █ █	:	[] █ █ [] [] [] █ []	█ █ [] █ [] █ █ []	[] █ [] [] [] █ █ █
%	█ [] [] [] █ [] [] []	[] [] [] █ █ █ [] []	█ [] █ [] [] █ [] █	0	[] █ [] [] [] [] [] []	█ [] █ █ █ █ █ []	[] [] █ [] █ █ █ █	;	█ █ █ [] [] [] █ []	[] █ █ █ [] █ █ []	█ █ [] [] [] █ █ █
&	[] [] █ [] █ [] [] []	█ [] [] █ █ █ [] []	[] [] [] [] █ █ [] █	1	█ █ [] [] [] [] [] []	[] █ [] █ [] █ [] []	█ [] █ [] █ █ █ █	<	[] █ [] [] █ [] █ []	█ █ █ █ [] █ █ []	[] █ █ [] [] █ █ █
'	█ [] █ [] █ [] [] []	[] [] █ █ █ █ [] []	█ [] [] [] █ █ [] █	2	[] █ █ [] [] [] [] []	█ █ [] █ [] █ [] []	[] █ [] [] [] █ [] █	=	█ █ [] [] █ [] █ []	[] █ [] █ █ █ █ []	█ █ █ [] [] █ █ █
(	[] [] [] [] [] [] █ []	█ [] █ █ █ █ [] []	[] [] █ [] █ █ [] █	3	█ █ █ [] [] [] [] []	[] █ █ █ [] █ [] []	█ █ [] [] [] █ [] █	>	[] █ █ [] █ [] █ []	█ █ [] █ █ █ █ []	[] █ [] [] █ █ █ █
)	█ [] [] [] [] [] █ []	[] [] [] █ [] █ █ []	█ [] █ [] █ █ [] █	4	[] █ [] [] █ [] [] []	█ █ █ █ [] █ [] []	[] █ █ [] [] █ [] █	?	█ █ █ [] █ [] █ []	[] █ █ █ █ █ █ []	█ █ [] [] █ █ █ █
*	[] [] █ [] [] [] █ []	█ [] [] █ [] █ █ []	[] [] [] [] [] █ █ █	5	█ █ [] [] █ [] [] []	[] █ [] █ █ █ [] []	█ █ █ [] [] █ [] █	@	[] [] [] █ [] [] [] []	█ █ █ █ █ █ █ []	[] █ █ [] █ █ █ █

Table of PU1, SS2 and SS3 Sequences, cont.

	ESC N	ESC O	ESC Q		ESC N	ESC O	ESC Q		ESC N	ESC O	ESC Q
A	■[] []■ [][] [][]	[][] [][] [][] []■	■■ ■[] ■■ ■■	L	[][] []■ ■[] ■[]	■[] ■[] [][] ■■	[][] ■■ []■ ■■	W	■■ ■■ ■[] [][]	[]■ ■[] ■[] []■	■■ []■ ■■ []■
B	[][] ■■ [][] [][]	■[] [][] [][] []■	[][] []■ []■ []■	M	■[] []■ ■[] ■[]	[][] [][] ■[] ■■	■[] ■■ []■ ■■	X	[]■ []■ [][] ■[]	■■ ■[] ■[] []■	[]■ ■■ ■■ []■
C	■[] ■■ [][] [][]	[][] ■[] [][] []■	■[] []■ []■ []■	N	[][] ■■ ■[] ■[]	■[] [][] ■[] ■■	[][] []■ ■■ ■■	Y	■■ []■ [][] ■[]	[]■ [][] [][] ■■	■■ ■■ ■■ []■
D	[][] []■ ■[] [][]	■[] ■[] [][] []■	[][] ■■ []■ []■	O	■[] ■■ ■[] ■[]	[][] ■[] ■[] ■■	■[] []■ ■■ ■■	Z	[]■ ■■ [][] ■[]	■■ [][] [][] ■■	[]■ []■ []■ ■■
E	■[] []■ ■[] [][]	[][] [][] ■[] []■	■[] ■■ []■ []■	P	[]■ []■ [][] [][]	■[] ■[] ■[] ■■	[][] ■■ ■■ ■■	[	■■ ■■ [][] ■[]	[]■ ■[] [][] ■■	■■ []■ []■ ■■
F	[][] ■■ ■[] [][]	■[] [][] ■[] []■	[][] []■ ■■ []■	Q	■■ []■ [][] [][]	[]■ [][] [][] []■	■[] ■■ ■■ ■■	\	[]■ []■ ■[] ■[]	■■ ■[] [][] ■■	[]■ ■■ []■ ■■
G	■[] ■■ ■[] [][]	[][] ■[] ■[] []■	■[] []■ ■■ []■	R	[]■ ■■ [][] [][]	■■ [][] [][] []■	[]■ []■ []■ []■	]	■■ []■ ■[] ■[]	[]■ [][] ■[] ■■	■■ ■■ []■ ■■
H	[][] []■ [][] ■[]	■[] ■[] ■[] []■	[][] ■■ ■■ []■	S	■■ ■■ [][] [][]	[]■ ■[] [][] []■	■■ []■ []■ []■	^	[]■ ■■ ■[] ■[]	■■ [][] ■[] ■■	[]■ []■ ■■ ■■
I	■[] []■ [][] ■[]	[][] [][] [][] ■■	■[] ■■ ■■ []■	T	[]■ []■ ■[] [][]	■■ ■[] [][] []■	[]■ ■■ []■ []■	_	■■ ■■ ■[] ■[]	[]■ ■[] ■[] ■■	■■ []■ ■■ ■■
J	[][] ■■ [][] ■[]	■[] [][] [][] ■■	[][] []■ []■ ■■	U	■■ []■ ■[] [][]	[]■ [][] ■[] []■	■■ ■■ []■ []■	`	[][] [][] []■ [][]	■■ ■[] ■[] ■■	[]■ ■■ ■■ ■■
K	■[] ■■ [][] ■[]	[][] ■[] [][] ■■	■[] []■ []■ ■■	V	[]■ ■■ ■[] [][]	■■ [][] ■[] []■	[]■ []■ ■■ []■	a	■[] [][] []■ [][]	[][] []■ [][] []■	■■ ■■ ■■ ■■

Table of PU1, SS2 and SS3 Sequences, cont.

	ESC N	ESC O		ESC N	ESC O		ESC N	ESC O
b	[] [] ■ [] [] ■ [] []	■ [] [] ■ [] [] [] ■	m	■ [] [] [] ■ ■ ■ []	[] [] [] ■ ■ [] ■ ■	x	[] ■ [] [] [] ■ ■ []	■ ■ ■ ■ ■ [] [] ■
c	■ [] ■ [] [] ■ [] []	[] [] ■ ■ [] [] [] ■	n	[] [] ■ [] ■ ■ ■ []	■ [] [] ■ ■ [] ■ ■	y	■ ■ [] [] [] ■ ■ []	[] ■ [] ■ [] [] ■ ■
d	[] [] [] [] ■ ■ [] []	■ [] ■ ■ [] [] [] ■	o	■ [] ■ [] ■ ■ ■ []	[] [] ■ ■ ■ [] ■ ■	z	[] ■ ■ [] [] ■ ■ []	■ ■ [] ■ [] [] ■ ■
e	■ [] [] [] ■ ■ [] []	[] [] [] ■ ■ [] [] ■	p	[] ■ [] [] [] ■ [] []	■ [] ■ ■ ■ [] ■ ■	{	■ ■ ■ [] [] ■ ■ []	[] ■ ■ ■ [] [] ■ ■
f	[] [] ■ [] ■ ■ [] []	■ [] [] ■ ■ [] [] ■	q	■ ■ [] [] [] ■ [] []	[] ■ [] ■ [] [] [] ■	|	[] ■ [] [] ■ ■ ■ []	■ ■ ■ ■ [] [] ■ ■
g	■ [] ■ [] ■ ■ [] []	[] [] ■ ■ ■ [] [] ■	r	[] ■ ■ [] [] ■ [] []	■ ■ [] ■ [] [] [] ■	}	■ ■ [] [] ■ ■ ■ []	[] ■ [] ■ ■ [] ■ ■
h	[] [] [] [] [] ■ ■ []	■ [] ■ ■ ■ [] [] ■	s	■ ■ ■ [] [] ■ [] []	[] ■ ■ ■ [] [] [] ■	~	[] ■ ■ [] ■ ■ ■ []	■ ■ [] ■ ■ [] ■ ■
i	■ [] [] [] [] ■ ■ []	[] [] [] ■ [] [] ■ ■	t	[] ■ [] [] ■ ■ [] []	■ ■ ■ ■ [] [] [] ■			
j	[] [] ■ [] [] ■ ■ []	■ [] [] ■ [] [] ■ ■	u	■ ■ [] [] ■ ■ [] []	[] ■ [] ■ ■ [] [] ■			
k	■ [] ■ [] [] ■ ■ []	[] [] ■ ■ [] [] ■ ■	v	[] ■ ■ [] ■ ■ [] []	■ ■ [] ■ ■ [] [] ■			
l	[] [] [] [] ■ ■ ■ []	■ [] ■ ■ [] [] ■ ■	w	■ ■ ■ [] ■ ■ [] []	[] ■ ■ ■ ■ [] [] ■			

TBC Tabulation Clear- HOST to Terminal

<ESC> [Pn g

The Tabulation Clear (TBC) command sequence clears one or more tabulation stops according to the parameter(s). When horizontal tabulation stops are cleared, the effect may be only on the active line or it may be on all lines depending upon the state of the Tabulation Stop Mode (TSM). (Refer also to CTC)

Pn	Action
Ø	Clears the horizontal tabulation stop at the current cursor position (default)
1	Clears the vertical tabulation stop at the current line
2	Clears all horizontal tabulation stops in the current line
3	Clears all horizontal tabulation stops
4	Clears all vertical tabulation stops

TSM Tabulation Stop Mode- HOST to Terminal

<ESC> [18 h   - to Set Mode (SM)

<ESC> [18 l*   - to Reset Mode (RM)

The Tabulation Stop Mode (TSM) is a parameter that is controlled by the Set Mode (SM) and Reset Mode (RM) ANSI sequences. When the Tabulation Stop Mode is in the reset state the setting and clearing of horizontal tabulation stops applies to the corresponding character position on all the lines. When the TSM is in the set state the horizontal taulation stops are independent for each line, e.g. line 1 might have tabulation stops at columns 5, 25, and 5Ø, while line 2 might have tabulation stops at columns 1Ø, 3Ø, 5Ø, and 75.

*Note: The last character in the sequence is a lower case L.

VTS Vertical Tabulation Set- HOST to Terminal and Terminal to HOST

<ESC> J

The Vertical Tabulation Set (VTS) command sequence sets a vertical tabulation stop at the current line.

VT52™ EMULATION IN TRANSPARENT MODE

In normal Transparent Mode operation host/terminal communications are handled according to the ANSI X3.64 protocol. However, the terminal's Transparent Mode operation can be adapted to DEC VT52 compatible systems which use a restricted version of the ANSI protocol along with some device specific commands.

VT52 Mode is selected by entering Setup Mode and setting item N to "NO" and then entering Transparent Mode by one of the means described in Part Three of the manual ("On-Line Mode Transitions"). Alternately, VT52 Mode can be selected while the terminal is in ANSI Transparent Mode by having the host send the ANSI PATM sequence <**ESC**>**[?2h.**

Once the terminal is in VT52 Transparent Mode, it may be returned to ANSI Transparent Mode in any one of the following three ways:

1) by having the host send the ANSI PATM sequence <**ESC**>**[?2l**[*]

2) by having the host send the VT52 sequence <**ESC**><

3) by selecting Setup Mode, setting Setup Menu item N to "YES", and returning the terminal to Transparent Mode

Command Sequences which Replace ANSI Sequences

While in VT52 Transparent Mode the terminal responds to all ANSI sequences described in the preceeding section except those listed by mnemonic in the left column of the table below. The ANSI sequences listed are replaced with the VT52 sequences listed in the right column.

ANSI Mnemonic	Function	VT52 Sequence
CUU	Cursor Up	<ESC> A
CUD	Cursor Down	<ESC> B
CUF	Cursor Right	<ESC> C
CUB	Cursor Left	<ESC> D
CUP	Cursor to Home	<ESC> H
ED	Erase to end of screen	<ESC> J
EL	Erase to end of line	<ESC> K
HVP	Direct cursor address	<ESC> Ylc[**]
PCDP	Enter Blind Cursor Mode	<ESC> x 5
PCDP	Exit Blind Cursor Mode	<ESC> y 5

[*] Note: Lower case L.

[**] Note: l = the ASCII character whose decimal equivalent is the line number plus 32
c = the ASCII character whose decimal equivalent is the column number plus 32

New Command Sequences

In VT52 Transparent Mode the terminal responds to the following additional command sequences:

Select Special Graphics character set <ESC> F

This sequence is equivalent to a (Control N) entered at the keyboard or <SO> received from the host.

Select ASCII character set <ESC> G

This sequence is equivalent to a (Control O) entered at the keyboard or <SI> received from the host.

Reverse line feed <ESC> I

Enter ANSI mode <ESC> <

Alternate Key Codes

In VT52 Transparent Mode operation, certain of the terminal's keys generate special codes. These codes are listed below.

Cursor (Arrow) Keys

Up Arrow	<ESC> A
Down Arrow	<ESC> B
Right Arrow	<ESC> C
Left Arrow	<ESC> D
Home	<ESC> H

Auxiliary Keypad Keys

Key	Keypad Numeric Mode	Keypad Application Mode
Ø	Ø	<ESC> ?p
1	1	<ESC> ?q
2	2	<ESC> ?r
3	3	<ESC> ?s
4	4	<ESC> ?t
5	5	<ESC> ?u
6	6	<ESC> ?v
7	7	<ESC> ?w
8	8	<ESC> ?x
9	9	<ESC> ?y
-	-	<ESC> ?m
,	,	<ESC> ?l
.	.	<ESC> ?n
ENTER	<CR>	<ESC> ?M
PF1	<ESC> P	<ESC> P
PF2	<ESC> Q	<ESC> Q
PF3	<ESC> R	<ESC> R
PF4	<ESC> S	<ESC> S

Erase Keys

Erase Page	<ESC> J
Erase Line	<ESC> K

APPENDICES

THE ASCII CODE

Hex			Hex			Hex			Hex		
Code	Char	ADE	Code	Char	ADE	Code	Char	ADE	Code	Char	ADE
00	NUL	0	20	SP	32	40	@	64	60	'	96
01	SOH	1	21	!	33	41	A	65	61	a	97
02	STX	2	22	"	34	42	B	66	62	b	98
03	ETX	3	23	#	35	43	C	67	63	c	99
04	EOT	4	24	$	36	44	D	68	64	d	100
05	ENQ	5	25	%	37	45	E	69	65	e	101
06	ACK	6	26	&	38	46	F	70	66	f	102
07	BEL	7	27	'	39	47	G	71	67	g	103
08	BS	8	28	(	40	48	H	72	68	h	104
09	HT	9	29	)	41	49	I	73	69	i	105
0A	LF	10	2A	*	42	4A	J	74	6A	j	106
0B	VT	11	2B	+	43	4B	K	75	6B	k	107
0C	FF	12	2C	,	44	4C	L	76	6C	l	108
0D	CR	13	2D	-	45	4D	M	77	6D	m	109
0E	SO	14	2E	.	46	4E	N	78	6E	n	110
0F	SI	15	2F	/	47	4F	O	79	6F	o	111
10	DLE	16	30	0	48	50	P	80	70	p	112
11	DC1*	17	31	1	49	51	Q	81	71	q	113
12	DC2	18	32	2	50	52	R	82	72	r	114
13	DC3**	19	33	3	51	53	S	83	73	s	115
14	DC4	20	34	4	52	54	T	84	74	t	116
15	NAK	21	35	5	53	55	U	85	75	u	117
16	SYN	22	36	6	54	56	V	86	76	v	118
17	ETB	23	37	7	55	57	W	87	77	w	119
18	CAN	24	38	8	56	58	X	88	78	x	120
19	EM	25	39	9	57	59	Y	89	79	y	121
1A	SUB	26	3A	:	58	5A	Z	90	7A	z	122
1B	ESC	27	3B	;	59	5B	[	91	7B	{	123
1C	FS	28	3C	<	60	5C	\	92	7C	|	124
1D	GS	29	3D	=	61	5D	]	93	7D	}	125
1E	RS	30	3E	>	62	5E	^	94	7E	~	126
1F	US	31	3F	?	63	5F	_	95	7F	DEL	127

*XON

**XOFF

Expanded Table of ASCII Control Codes

ASCII Symbol	Keyboard Keys	Conventional Meaning	Binary Code	Hex Code	Octal Code	Decimal Code
<NUL>	(Control @)	Time Delay	P0000000	00	000	0
<SOH>	(Control A)	Start of Header	P0000001	01	001	1
<STX>	(Control B)	Start of Text	P0000010	02	002	2
<ETX>	(Control C)	End of Text	P0000011	03	003	3
<EOT>	(Control D)	End of Transmission	P0000100	04	004	4
<ENQ>	(Control E)	Enquire	P0000101	05	005	5
<ACK>	(Control F)	Acknowledge	P0000110	06	006	6
<BEL>	(Control G)	Sound Bell	P0000111	07	007	7
<BS>	BACKSPACE					
	(Control H)	Backspace	P0001000	08	010	8
<HT>	TAB)					
	(Control I)	Horizontal Tab	P0001001	09	011	9
<LF>	LINEFEED					
	(Control J)	Linefeed	P0001010	0A	012	10
<VT>	(Control K)	Vertical Tab	P0001011	0B	013	11
<FF>	(Control L)	Form Feed	P0001100	0C	014	12
<CR>	RETURN					
	(Control M)	Carriage Return	P0001101	0D	015	13
<SO>	(Control N)	Shift Out	P0001110	0E	016	14
<SI>	(Control O)	Shift In	P0001111	0F	017	15
<DLE>	(Control P)	Data Link Escape	P0010000	10	020	16
<DC1>	(Control Q)	Device Control 1				
		XON	P0010001	11	021	17
<DC2>	(Control R)	Device Control 2	P0010010	12	022	18
<DC3>	(Control S)	Device Control 3				
		XOFF	P0010011	13	023	19
<DC4>	(Control T)	Device Control 4	P0010100	14	024	20
<NAK>	(Control U)	Negative Acknowledge	P0010101	15	025	21
<SYN>	(Control V)	Synchronization	P0010110	16	026	22
<ETB>	(Control W)	End Transmission				
		Block	P0010111	17	027	23
<CAN>	(Control X)	Cancel	P0011000	18	030	24
<EM>	(Control Y)	End of Medium	P0011001	19	031	25
<SUB>	(Control Z)	Substitute	P0011010	1A	032	26
<ESC>	ESC					
	(Control [)	Escape (CSI)	P0011011	1B	033	27
<FS>	(Control \)	Field Separator	P0011100	1C	034	28
<GS>	(Control])	Group Separator	P0011101	1D	035	29
<RS>	(Control ^)	Record Separator	P0011110	1E	036	30
<US>	(Control _)	Unit Separator	P0011111	1F	037	31

GLOSSARY OF TERMS (ANSI)

Active position - The character position in the visual display that is to display the graphic symbol representing the next graphic character.

Character position - That portion of a visual display which is displaying or is capable of displaying a graphic symbol.

Control character - A character whose occurrence in a particular context initiates, modifies, or stops a control function.

Control function - An action that affects the processing, transmission, format, or interpretation of data.

Control sequence - A sequence of characters that is used for control purposes to perform a control function, that begins with the control sequence introducer (CSI) control, and that may contain a parameter string.

Control sequence introducer (CSI) - An escape sequence that provides supplementary controls and that is itself a prefix affecting the interpretation of a limited number of contiguous characters.

Control string - A string of characters that is used to perform a control function and that is delimited by an opening and closing delimiter control.

Current Cursor Position - The next position on the visual display where a graphic character will be placed. (see Active Position)

Cursor - A visual representation of the active position which is a blinking underline.

Cursor control - An editor function that moves the active position.

Default - A function-dependent value that is assumed when no explicit value, or a value of Ø, is specified.

Display - The current active area of the screen, i.e., the area inside the scrolling region, or the entire screen if no scrolling region is specified.

Editor function - A control that affects the layout or positioning of previously entered or received information in a printing or cathode ray tube device and that is intended to be interpreted and executed without remaining in the data stream. (See format effector.)

Escape character (ESC) - A control character that provides supplementary characters (code extension) and that is itself a prefix affecting the interpretation of a limited number of contiguous characters.

Escape sequence - A sequence of characters that is used for control purposes to perform a control function and whose first character is the escape <ESC> control character.

Final character - A character whose bit combination terminates an escape or control sequence.

Format effector - A control that affects the layout or positioning of information on the screen and that may remain in the data stream subsequent to interpretation and processing. (See editor function.)

Graphic character - A character, other than a control character, that has a visual representation normally handwritten, printed, or displayed.

Home - The character position at the origin, which is the uppermost and leftmost position within the display.

Numeric parameter - A string of bit combinations that represent a number.

Parameter - (1) A string of one or more characters that represent a single value: (2) The value so represented. (Designated by Pn).

Parameter string - A string of characters that represent one or more parameter values.

Selective parameter - A string of bit combinations that selects a subfunction from a specified list of subfunctions.

4027 MODE COMMAND SUMMARY

(Alphabetical by Command Keyword)

<u>Note</u>: All keywords must be immediately preceeded by the command character, usually the exclamation point "**!**". Commands are terminated with a <CR>.

<u>Keyword</u>	<u>Parameters</u>	<u>Group</u>	<u>Page</u>
ATTribute	{**font**} {**logical**} {**visual**{,**-visual**}}	E	4.74
BACktab	{**count**}	F	4.97
BEL		A	4.9
BITs	{**number**}	G	4.102
CIRcle	**rad**{,**start**{,**end**{,**seg**}}}	C	4.42
CITalics	{**angle**}	E	4.76
CLEar		H	4.112
CMOde	{**setting**}	D	4.58
COLor	**Cn**$_{vector}${,**Cn**$_{boundary}$}	B	4.31
COMmand	**chr**	A	4.10
CROtate	{**angle**}	E	4.78
CSCale	{**width**{,**height**}}	E	4.79
CSPace	{**type**}	E	4.80
CVEctor	**x-dist**{,**y-dist**}	E	4.81
CVSymbol	**chr**,**a**$_1$,**a**$_2$,**x**$_1$,**y**$_1${,**x**$_2$,**y**$_2$ ...,**x**$_n$,**y**$_n$}	E	4.83
DFLine	**Pon**$_1$,**Poff**$_2${,**Pon**$_3$,**Poff**$_4$ ...,**Pon**$_7$,**Poff**$_8$}	B	4.30
DFOnt	**font**	E	4.86
DISable		C	4.54
DOWn	{**count**}	F	4.98
ENAble	{**seq**}	C	4.51

ERAse	{area{,background$_{gr}$}	B	4.33
EXPand	macro	H	4.112
FONt	{font$_{or}$,}font$_{dest}${,Cn$_{bg}$,Cn$_{fg}$}	E	4.87
GRAphic	row$_{b}$,row$_{e}${,col$_{b}${,col$_{e}$}}	B	4.23
GTEst		A	4.11
HCOpy		I	4.114
INK	{setting}	C	4.53
JUMp	{row{,col}}	F	4.100
LEArn	identifier{,string$_{1}$, ... ,string$_{n}$,}	H	4.110
LEFt	{count}	F	4.98
LINe	{type}	B	4.29
LPRinter	{pr} {chan} {sq} {rot} {rdc} {line} {ff}	I	4.115
MAP	Cn,hue{,light{,sat}}	D	4.61
MARgins	{left{,right}}	A	4.12
MIX	Cn{,red{,grn{,blu}}}	D	4.67
MONitor	{lines} {source{,source}}	B	4.22
MOVe	x$_{1}$,y$_{1}$,x$_{2}$,y$_{2}$,x$_{3}$,y$_{3}$	G	4.104
PATtern	Pn{,Cn$_{bg}$},Cn$_{fg1}$,row$_{1}$ ...,row$_{14}$ {,Cn$_{fg2}$,row$_{1}$ ... ,row$_{14}$}	D	4.69
PIE	rad{,start{,end{,seg}}}	C	4.45
POLygon	xb,yb{,xv$_{1}$,yv$_{1}$ ... ,xv$_{n-1}$,yv$_{n-1}$}	C	4.47
RASter	data	G	4.105
READback	count	G	4.107
REPort	status	A	4.13
RIGht	{count}	F	4.99
RMAp	Cn,d-hue{,d-light{,d-sat}}	D	4.65
RPOlygon	dxb,dyb{,dxv$_{1}$,dyv$_{1}$ ... ,dxv$_{n-1}$,dyv$_{n-1}$}	C	4.49
RVEctor	dxb,dyb{,dxe$_{1}$,dye$_{1}$ ... ,dxe$_{n}$,dye$_{n}$}	C	4.40
SFOnt	{chr$_{b}${,chr$_{e}$}}	E	4.89

SHRink	{factor}	B	4.27
SNOopy	{setting)	A	4.15
STOps	{stop$_1$ ... ,stop$_n$}	A	4.16
STRing	,string$_1$,{,string$_2$, ... ,string$_n$,}	E	4.92
SYMbol	chr,font{,Cn$_{bg}$},Cn$_{fg1}$,row$_1$ ...,row$_{14}$ {,Cn$_{fg2}$,row$_1$ ... ,row$_{14}$}	E	4.9Ø
SYStat		A	4.17
TAB	{count}	F	4.97
TESt		A	4.18
UP	{count}	F	4.99
VECtor	xb,yb{,xe$_1$,ye$_1$ ... ,xe$_n$,ye$_n$}	C	4.38
WORkspace	{lines} {source{,source}}	B	4.2Ø

4010 MODE COMMAND SUMMARY

Mode Specific Commands

Command Character(s)	Parameter(s)	Function
<CR>	{**string**}	Display text beginning at left margin on line occupied by graphics beam
<FS>	**string**$_1${...**string**$_n$}	Plot points at given coordinates
<GS>	**string**$_1${...**string**$_n$}	Move graphics beam and draw vectors at given coordinates
<US>	{**string**}	Display text beginning at current graphics beam position
<ESC><ENQ>		Send status to the host and set Bypass Condition
<ESC><ETB>		Transmit contents of screen memory (optional I/O Connector required)
<ESC><FF>	{**string**}	Erase screen and display text beginning at home position
<ESC><SUB>		Enter Graphics Input Mode
<ESC>/f		Move crosshair cursor to graphics beam position
<ESC>	**type**	Set line type for vectors

TABLES FOR FINDING ASCII CODES OF 4010 COORDINATES

Given an (x,y) coordinate pair expressed as decimal integers, the y-high and x-high codes can be found directly from the table below.

x or y in the range	ASCII character for x-high or y-high
0-31	SP
32-63	!
64-95	"
96-127	#
128-159	$
160-191	%
192-223	&
224-255	'
256-287	(
288-319	)
320-351	*
352-383	+
384-415	,
416-447	-
448-479	.
480-511	/
512-543	0
544-575	1
576-607	2
608-639	3
640-671	4
672-703	5
704-735	6
736-767	7
768-799	8
800-831	9
832-863	:
864-895	;
896-927	<
928-959	=
960-991	>
992-1023	?

To find the y-low code for a given decimal y value, note the range in which the value falls in the table on the preceeding page. Subtract the number with which the range begins from the given y value to find its position in the range. Then use the position number to find the y-low code in the table below.

To find the x-low code for a given decimal x value, the proceedure is the same as for the y-low code.

x's or y's position within a given range	ASCII Character for y-low	ASCII Character for x-low
Ø	`	@
1	a	A
2	b	B
3	c	C
4	d	D
5	e	E
6	f	F
7	g	G
8	h	H
9	i	I
1Ø	j	J
11	k	K
12	l	L
13	m	M
14	n	N
15	o	O
16	p	P
17	q	Q
18	r	R
19	s	S
2Ø	t	T
21	u	U
22	v	V
23	w	W
24	x	X
25	y	Y
26	z	Z
27	{	[
28	|	\
29	}	]
3Ø	~	^
31	<DEL>*	_

*The sequence <ESC>? may be substituted for <DEL> if necessary.

SUMMARY OF

TRANSPARENT MODE ANSI CONTROL SEQUENCES

(In order by Mnemonic)

CBT Cursor Backward Tabulation-	ESC [Pn Z
CHT Cursor Horizontal Tabulation-	ESC [Pn I
CPR Cursor Position Report-	ESC [Pn ; Pn R
CRM Control Representation Mode-	ESC [3 h ESC [3 l*
CTC Cursor Tabulation Control-	ESC [Pn W
CUB Cursor Backward-	ESC [Pn D
CUD Cursor Downward-	ESC [Pn B
CUF Cursor Forward-	ESC [Pn C
CUP Cursor Position-	ESC [Pn ; Pn H
CUU Cursor Up-	ESC [Pn A
CVT Cursor Vertical Tabulation-	ESC [Pn Y
DA Device Attributes-	ESC [Pn c
DSR Device Status Report-	ESC [Pn n
ED Erase In Display-	ESC [Pn J
EL Erase In Line-	ESC [Pn K
HTS Horizontal Tabulation Set-	ESC H
HVP Horizontal & Vertical Position-	ESC [Pn ; Pn f
IND Index-	ESC D
KAM Keyboard Action Mode-	ESC [2 h ESC [2 l*
LNM Line Feed New Line Mode-	ESC [2Ø h ESC [2Ø l*

MC	Media Copy-	ESC [Pn i
NEL	Next Line-	ESC E
NP	Next Page-	ESC [U
PALN	(Private) Screen Alignment Display-	ESC # 8
PATM	(Private) ANSI Terminal Mode-	ESC [? 2 h ESC [? 2 l*
PAWM	(Private) Auto Wrap Mode-	ESC [? 7 h ESC [? 7 l*
PCDP	(Private) Cursor Display Mode-	ESC [> 5 h ESC [> 5 l*
		or ESC [1 v ESC [Ø v
PCKM	(Private) Cursor Keys Mode-	ESC [? 1 h ESC [? 1 l*
PCNF	(Private) Confidence Test-	ESC [Pn y
PDFK	(Private) Default Function Key Mode-	ESC [? 29 h ESC [? 29 l*
PFNR	(Private) Function Key Rate Mode-	ESC [? 31 h ESC [? 31 l*
PIDT	(Private) Identify Terminal-	ESC Z
PKPA	(Private) Keypad Application Mode-	ESC =
PKPN	(Private) Keypad Numeric Mode-	ESC >
PLOT	(Private) Plot Graphics Commands-	ESC [Pn;Pn;Pn;Pn z
PMRG	(Private) Set Margins-	ESC [Pn ; Pn r
POM	(Private) Origin Mode-	ESC [? 6 h ESC [? 6 l*
PP	Previous Page-	ESC [V
PPFN	(Private) Program Function Keys-	ESC [Pn_{key} $\{;Pn_1 \ldots ;Pn_n\}$ s
PRCP	(Private) Restore Cursor Position-	ESC 8
PRKC	(Private) Repeat Key Control	ESC [? 8 h ESC [? 8 l*
PRM	(Private) Reset Mode-	ESC [? Pn l*

PRPT (Private) Request/Report Terminal Parameters-	ESC [Pn ; ... Pn x
PSCP (Private) Save Cursor Position-	ESC 7
PSM (Private) Set Mode-	ESC [? Pn h
PU1 (Private) Use One-	ESC Q chr
RI Reverse Index-	ESC M
RIS Reset to Initial State-	ESC c
RM Reset Mode-	ESC [Pn ; ... Pn l*
SGR Select Graphic Rendition-	ESC [Pn ; ... Pn m
SM Set Mode-	ESC [Pn ; ... Pn h
SRM Send-Receive Mode-	ESC [12 h ESC [12 l*
SS2 Single Shift Two	ESC N chr
SS3 Single Shift Three	ESC O chr
TBC Tabulation Clear-	ESC [Pn g
TSM Tabulation Stop Mode-	ESC [18 h ESC [18 l*
VTS Vertical Tabulation Set-	ESC J

*Note: The character is a lower case L.

TERMINAL CONTROL CODES AND THEIR ACTIONS

Code	Selectable Recognition	Function
<BEL>		Causes terminal's audible indicator to sound
<BS>		Causes cursor to move one space to the left
<CAN>*		Causes the terminal to enter Transparent Mode when it is in 4010 Mode
<CR>		Causes the cursor to move to the left margin of the current line. If item D "New Line" is selected in Setup Mode, <CR> is interpreted as <CR><LF>.
<DC1>	Yes	Causes the terminal to resume sending to the host (same as XON)
<DC3>	Yes	Causes the terminal to stop sending to the host (same as XOFF)
<ESC> c**		Causes the terminal to be reset.
<ESC>"6g		Causes the terminal to enter 4027 Mode when it is in 4010 Mode. This sequence is normally sent by the host.
<ESC>"7g		Causes the terminal to enter 4010 Mode when it is in 4027 Mode. This sequence is normally sent by the host.
<ESC>"0g		Causes the terminal to enter Transparent Mode when in 4010 or 4027 Mode. This sequence is normally sent by the host.
<ENQ>	Yes	Causes the terminal to send its answerback message to the host.
<FF>		Same as <CR>
<GS>		Causes the terminal to enter 4010 Mode when it is in Transparent Mode

*This code can be changed by means of selections made in Setup Mode (Graphics Submenu).

**This code should be used only with terminals having V2.8 and subsequent software. It will not work on V2.7 terminals.

<HT> Causes cursor to move one tab stop to the right

<LF> Causes the cursor to move down one line

<SI> Causes the terminal's alternate character set to be used for text display. When the terminal is in 4Ø27 Mode, only the monitor display is affected. (See SCS in Part Six.)

<SO> Causes the terminal to revert to its normal character set for text display (monitor only in 4Ø27 Mode).

<VT> Causes the cursor to move up one line.

Software Problem Report

Use this form to report errors or problems in Intecolor Corporation's software products. Please report only one problem per form.

Mail to:

Intecolor Corporation
Software Development
225 Technology Park/Atlanta
Norcross, Georgia 30092

Date ______________________

Your name and address:

____

____

____

____

Program Name ____________________

Program Version __________________

Purchase Date ____________________

ISC unit serial number ___________

Phone number where you can be reached during

the working day ______________________

Description of the problem and the circumstances surrounding its occurrence. It is best if the problem can be reduced to a simple test case. Include a diskette or write on the back of this form if you wish.

____

____

____

____

____

____

____

____

____

(Please do not write below this line)

Disposition

____

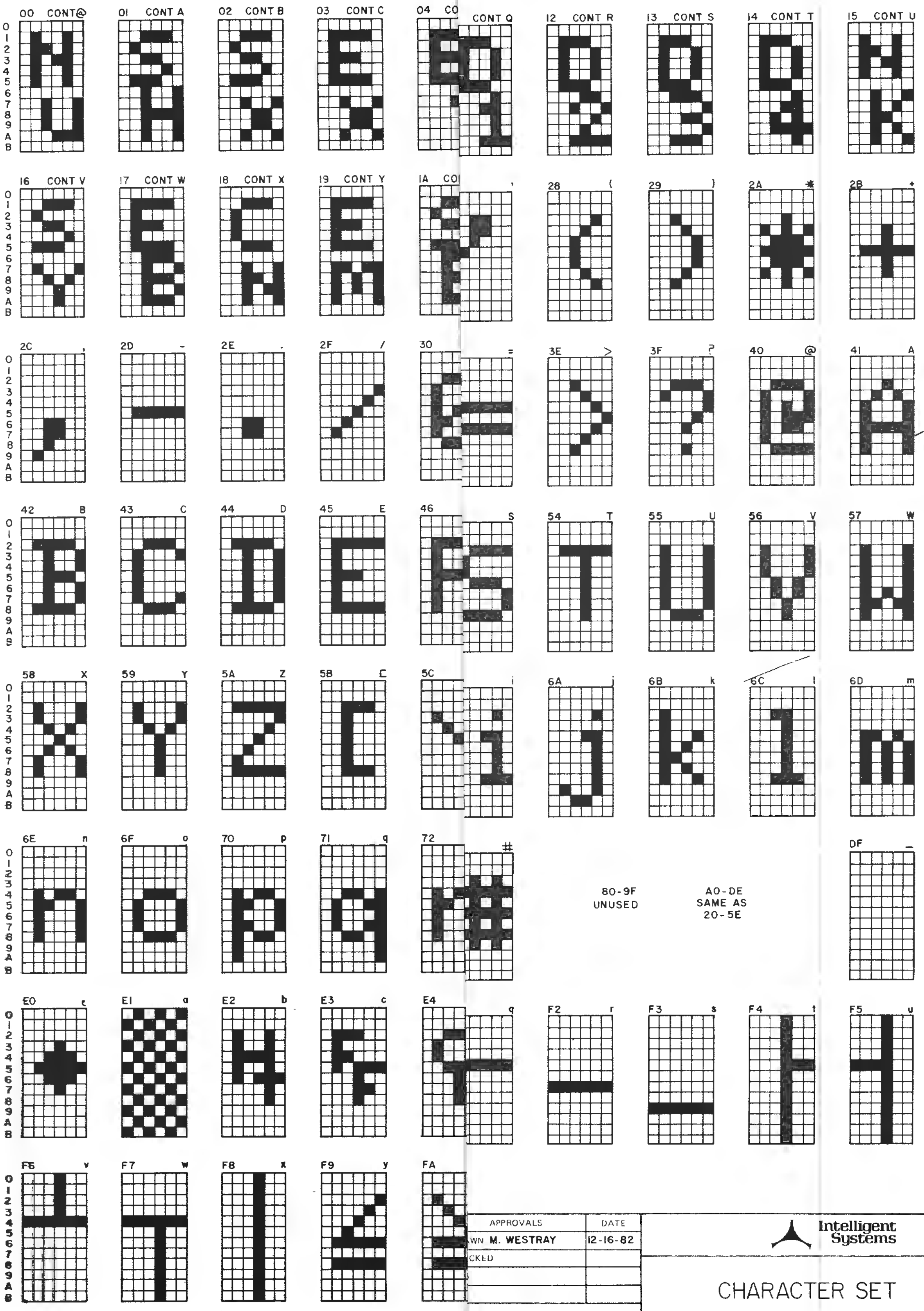
____

____

Sequence ________ Date ________________

ECN ___________ Programmer ____________

REVISIONS				
REV	DESCRIPTION	ECN	DATE	APPROVED
1	ENG. RELEASE	2912	1·11·83	REB 1/11/83



APPROVALS	DATE
WN M. WESTRAY	12-16-82
CKED	

PROPRIETARY

document, submitted in confidence, ains proprietary information which shall be reproduced or transferred to other ments or disclosed to others or used for afacturing or any other purpose without written permission of Intelligent Systems

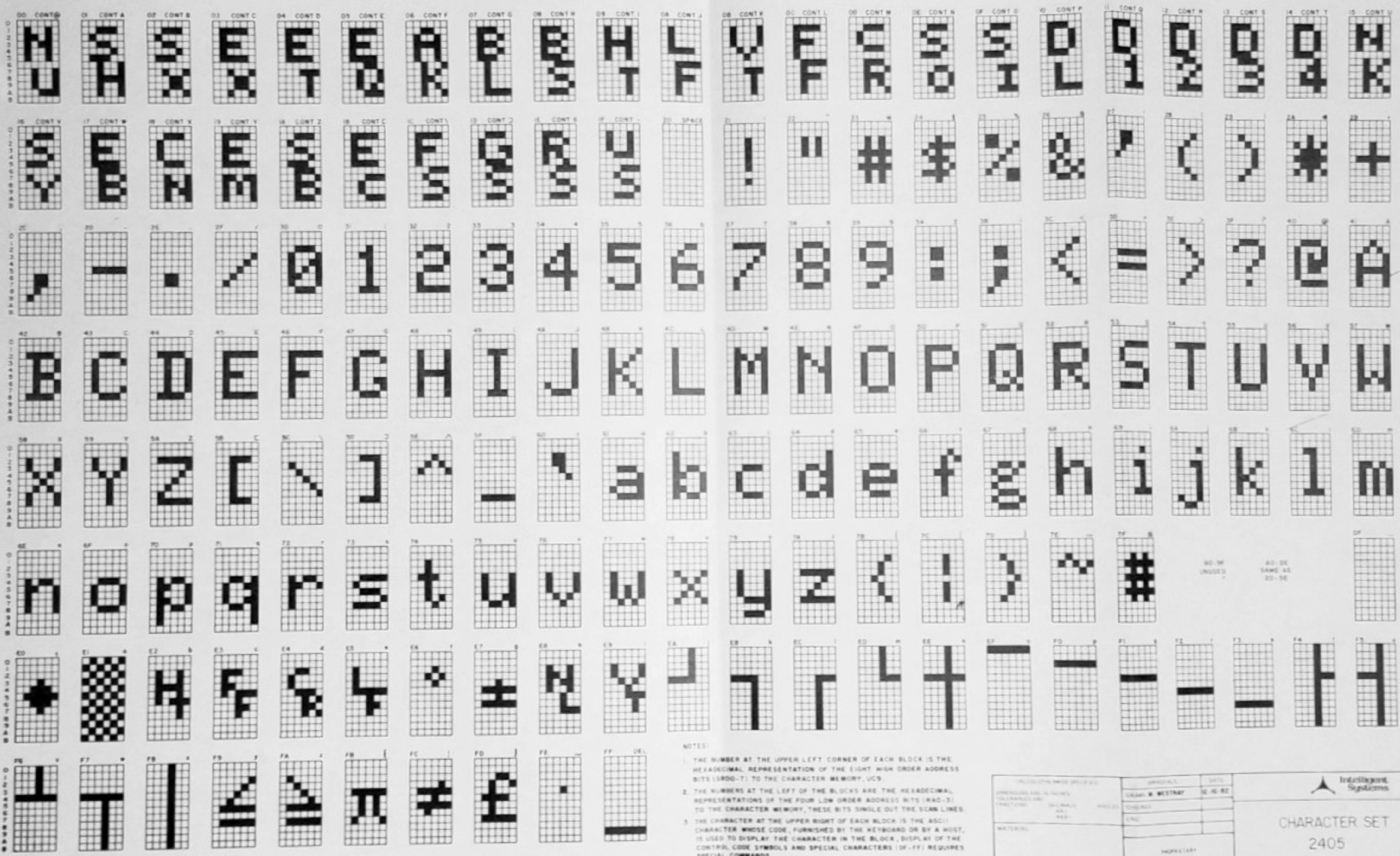
Intelligent Systems

CHARACTER SET
2405

SIZE D | DRAWING NO 102307 | REV 1

SCALE | SHEET 1 OF 1

REV	DESCRIPTION	ECO	DATE	APPROVED
1	ENG. RELEASE	2912	1-11-83	[illegible]



NOTES:

1. THE NUMBER AT THE UPPER LEFT CORNER OF EACH BLOCK IS THE HEXADECIMAL REPRESENTATION OF THE EIGHT HIGH ORDER ADDRESS BITS (SRDO-7) TO THE CHARACTER MEMORY, UC9.
2. THE NUMBERS AT THE LEFT OF THE BLOCKS ARE THE HEXADECIMAL REPRESENTATIONS OF THE FOUR LOW ORDER ADDRESS BITS (RAO-3) TO THE CHARACTER MEMORY. THESE BITS SINGLE OUT THE SCAN LINES.
3. THE CHARACTER AT THE UPPER RIGHT OF EACH BLOCK IS THE ASCII CHARACTER WHOSE CODE, FURNISHED BY THE KEYBOARD OR BY A HOST, IS USED TO DISPLAY THE CHARACTER IN THE BLOCK. DISPLAY OF THE CONTROL CODE SYMBOLS AND SPECIAL CHARACTERS (0F-FF) REQUIRES SPECIAL COMMANDS.

UNLESS OTHERWISE SPECIFIED
DIMENSIONS ARE IN INCHES
TOLERANCES ARE:
FRACTIONS DECIMALS ANGLES
.XX
.XXX

MATERIAL

FINISH

DO NOT SCALE DRAWING

APPROVALS	DATE
DRAWN R. WESTRAP	12-16-82
CHECKED	
ENG	

PROPRIETARY

This document, delivered in confidence, contains proprietary information which shall not be reproduced or transferred to other documents or disclosed to others or used for manufacturing or any other purpose without prior written permission of Intelligent Systems.

Intelligent Systems

CHARACTER SET
2405

SIZE		DRAWING NO.	REV
D		102307	1

SCALE SHEET 1 OF 1