TEK PROGRAMMER'S REFERENCE MANUAL

Part No. 070-3806-00
Product Group 12

4663
INTERACTIVE DIGITAL PLOTTER

Tektronix®
COMMITTED TO EXCELLENCE

TEK PROGRAMMER'S REFERENCE MANUAL

Part No. 070-3806-00
Product Group 12

4663

INTERACTIVE DIGITAL PLOTTER

*Please Check for
CHANGE INFORMATION
at the Rear of this Manual*

First Printing JAN 82

Tektronix®
COMMITTED TO EXCELLENCE

WARNING

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

This instrument, in whole or in part, may be protected by one or more U.S. or foreign patents or patent applications. Information provided on request by Tektronix, Inc., P.O. Box 500, Beaverton, Oregon 97077.

MANUAL REVISION STATUS

PRODUCT: 4663 Interactive Digital Plotter

This manual supports the following versions of this product: Serial Numbers B060100 and up and those instruments with Version 5 or greater firmware installed.

REV DATE	DESCRIPTION
JAN 1982	Original Issue. This manual and the 4663 Operator's Manual (070-3807-00) replace an earlier version 4663 Operator's Manual (070-2670-00). This manual also replaces an October 81 preliminary version (061-2638-00).

CONTENTS

Section 4 **SERIAL INTERFACE** **Page**

Section 5 **GPIB INTERFACE** **Page**

Section 6 **INTRODUCTION TO COMMANDS AND ARGUMENTS**

ILLUSTRATIONS

TABLES

ABOUT THIS MANUAL

GOALS OF THIS MANUAL

The *4663 Interactive Digital Plotter Programmer's Reference Manual* is for the programmer who is writing a package of computer subroutines that communicate with the TEKTRONIX 4663 Interactive Digital Plotter (hereafter refered to as "4663" or "Plotter"). This manual describes the Plotter commands and tells how to issue them by entering sequences of ASCII characters at a compatible terminal or controller.

This manual assumes the reader has a knowledge of programming fundamentals and terminology; it is not intended to be a tutorial of programming procedures. Rather, the manual provides instructions about how to control the Plotter using RS-232-C (Serial) communications, General Purpose Interface Bus (GPIB) communications, or by TEKTRONIX 4050 Series BASIC commands. This manual is written so that users with limited computer experience can learn to communicate with the Plotter.

This manual is sequenced with two different colored tabs. Yellow tabs locate major areas of the manual while white tabs indicate command description sections.

ORGANIZATION

This manual is organized with the fundamental concepts and first-time operation in the first six sections; the remaining sections of the manual are devoted to groups of commands. Concepts relevant to the particular command groups are included at the beginning of each command section.

- Section 1 (Introduction) describes the basic graphic concepts used with the Plotter.
- Section 2 (First-Time Operation Through the Serial Interface).
- Section 3 (First-Time Operation Through the GPIB Interface).
- Sections 4 and 5 (Serial Interface and GPIB Interface) provide an overview of specific programming needs for Serial or GPIB interfaces between the Plotter and the host/controller. If you are already well acquainted with programming techniques, then you may wish to skip these sections and go directly to Section 6.
- Section 6 (Introduction to Commands and Arguments) describes the command syntax and conventions used in this manual and should be understood before going to the individual command sections.

- Sections 7 through 16 (command description sections) provide specific information for different groups of Plotter commands. Each command description section begins with a discussion of the major concepts relevant to the commands included in that section. The command description sections are:
  - Serial Interface Commands — Section 7
  - GPIB Interface Commands — Section 8
  - Device Commands — Section 9
  - Alpha Commands — Section 10
  - Graphics Commands — Section 11
  - Transform Commands — Section 12
  - Digitizing Commands — Section 13
  - Macro Commands — Section 14
  - Downloadable Character Commands — Section 15
  - Media Advance Commands — Section 16

While reading this manual, you may find it helpful to refer to the Appendices and the Index located at the back of the manual.

- Command Summary (Serial) — Appendix A
- ASCII Code Chart — Appendix B
- Serial Interface Connector — Appendix C
- Command Summary (GPIB) — Appendix D
- GPIB Description — Appendix E
- Coordinate Conversion Chart — Appendix F
- Default Page, Viewport, and Window Coordinates — Appendix G
- 4662 Emulation Considerations — Appendix H
- Plotter/4014 Command Capability — Appendix I
- Error Types — Appendix J
- Glossary — Appendix K
- Command Index — Appendix L
- Index — Located after the last appendix.

OTHER 4663 MANUALS

The following related manuals are available:

- *4663 Interactive Digital Plotter Operator's Manual*
- *4663 Interactive Digital Plotter Reference Guide*
- *4663 Interactive Digital Plotter Service Manual (Volume 1)*
- *4663 Interactive Digital Plotter Service Manual (Volume 2)*

SYNTAX

The following summarizes the associated conventions and notations used throughout this manual. For more information, refer to Section 6, *Introduction to Commands and Arguments*.

CHARACTERS

ASCII Characters

The 94 printing ASCII characters (numbers, symbols, and uppercase and lowercase letters) are represented by their normal symbols.

The ASCII Control, SPACE, and DELETE characters are each represented by an appropriate single symbol. See the ASCII Code Chart appendix.

NOTE

The ASCII SPACE character is always shown as $^{S}{}_{P}$*, and not as a blank space between printed characters.*

Examples: **& 1 2 A B a b** $^{E}{}_{C}$ $^{S}{}_{P}$ $^{D}{}_{T}$

The last three ASCII characters are ESCAPE, SPACE, and DELETE.

Special Characters

Two symbols are replaced by certain single ASCII characters. The two symbols, $^{A}{}_{T}$ and $^{A}{}_{D}$, represent the attention and address characters, respectively, and are replaced as follows:

$^{A}{}_{T}$ — attention character, replaceable by

$^{E}{}_{C}$ **! ∧** or $^{S}{}_{Y}$

$^{A}{}_{D}$ — address character, replaceable by

A B C D E F G or **H**

LITERAL VS. VARIABLE ELEMENTS

Boldface Type

A literal element, which is entered exactly as shown, is shown in bold.

Regular Type

A variable element, which is replaced by appropriate specific information, is shown in regular type; a single element can be represented by one symbol, one word, or words connected by hyphens.

Examples

$^{A}{}_{T}$ $^{A}{}_{D}$ **R** rotation-angle

where three elements are replaced and R is entered exactly as shown.

! A R 45

where five ASCII characters are entered exactly as shown.

COMMAND ELEMENTS

Element Types

The last words of a 4663 Plotter variable element name indicates the element type required by the 4663, as shown below:

Last Words	Element Type
selector	selector
string	string
print-string	print string
pa	primary address
target-variable	target variable

Other endings imply a numeric element.

Examples:

- axis-selector is a selector element
- print-character-string is a string element
- radius is a numeric element

Omitted Numeric Elements

When a numeric element is omitted, a value of zero is assumed. Any exceptions are noted where they occur.

Argument Separators

A nonbold comma (,) between elements can be replaced by either a bold comma (**,**) or one or more SP characters. A nonbold semicolon (;) between variable elements can be replaced by either a bold semicolon (**;**) or a bold comma (**,**). Exceptions are noted where they apply.

Command Terminators

Command terminators are generally NOT shown in the individual command descriptions. Specific termination varies with the interface and the Command/Response Format used. Refer to Sections 4 and 5 for instructions on how to terminate commands.

NOTATIONS

Brackets [ ]

An element inside brackets is *optional*. Stacked elements within brackets indicate selection of one or none of the elements.

Example: [**0**] select **0**, **1**, or neither
[**1**]

Braces { }

When multiple elements are stacked within braces, one element *must* be selected.

Example: {**0**}
{**1**} select **0**, **1**, or **2**
{**2**}

Dots . . .

Three dots (ellipsis) indicate that a previous element *may* be repeated.

Example: SP. . . one or more SP characters

Indented for Continuation

If a command is continued on to the next printed line, the additional line is indented.

Example: AT AD **X** x-value,y-value[,x-value,
y-value]. . .

Section 1

INTRODUCTION

This section describes some of the basic terminology and concepts fundamental to the 4663 Plotter graphics. These terms and concepts are used throughout this manual and should be understood before proceeding.

Much of this overview applies to programmable commands, which are the focus of this manual. Some of these programmable operations (for example, the SET VIEWPORT command) are also available by using the front panel switches or Parameter Entry Card selections. Refer to the *4663 Interactive Digital Plotter Operator's Manual* for additional discussion of the front panel operations and for information concerning the physical parts of the Plotter.

This section includes descriptions of the following:

- The Plotter
- The Plotter Graphics System (General and Specific)
- General Plotting Information
- Page Scaling
- Viewport Scaling

Concepts for specific commands are included in the Command sections of this manual.

THE PLOTTER

The primary function of the Plotter (Figure 1-1) is to transfer graphic information onto media, such as paper or polyester film. This transfer is accomplished by commanding pens on the Plotter to draw lines and print alphanumeric characters.

The commands associated with drawing lines are referred to as *graphics commands*. These include the fundamental MOVE and DRAW commands:

- MOVE—causes the pen to move to a specified position (but not to draw a line).
- DRAW—causes the pen to draw a line to a specified position.

Additional graphics commands include setup commands for establishing graphics parameters, commands to draw circles and arcs (requires Option 31), and commands for specific operations such as axis generation.

Commands to print alphanumeric characters are referred to as *alpha commands* and are discussed extensively in Section 10 of this manual.

The 4663 digitizes points and sends those coordinates back to a host computer or terminal. Some graphics commands affect the digitizing function.

Clay Archer

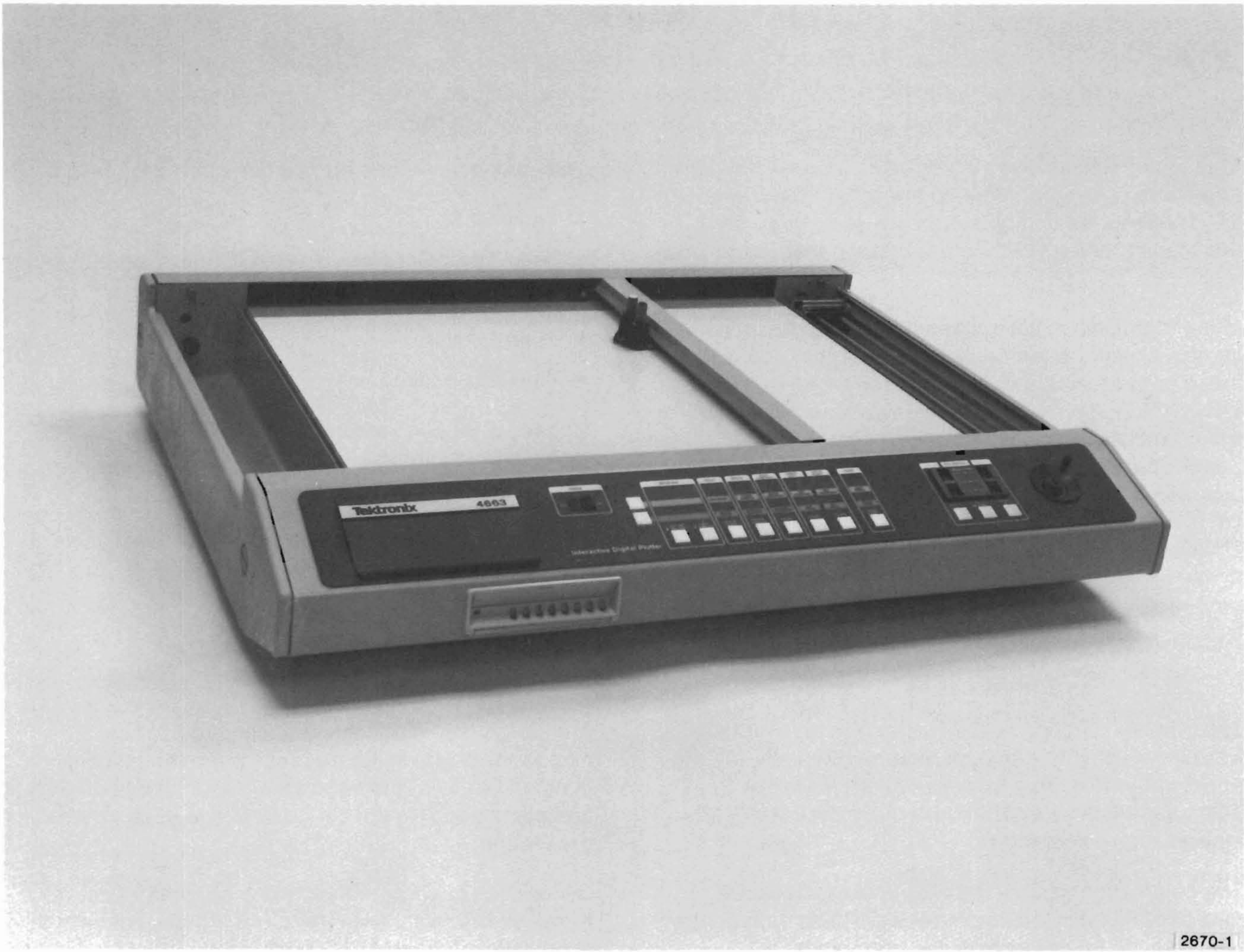


Figure 1-1. The 4663 Interactive Digital Plotter.

PLOTTER GRAPHICS SYSTEM—GENERAL

The Plotter's graphic system can be classified into two areas: the Plotter Domain and the User Domain. Some terms associated with these domains are defined here.

In the Plotter Domain:

- *Platen*—The large flat surface on which the media is placed.
- *Page*—A rectangular area on the Platen that is used for plotting.
- *Viewport*—A rectangular area of the Page onto which the plot is drawn.

In the User Domain:

- *World Space*—The space associated with the object or picture.
- *Window*—A two-dimensional rectangular portion of the World Space which is projected onto the Viewport.

The Plotter Domain and the User Domain each have coordinate systems and related units:

- *Device Coordinate System*—The device (Plotter) dependent Cartesian coordinate system for the Page.
- *Device Units*—Units for the Device Coordinate System.
- *World Coordinate System*—A Cartesian coordinate system for the World Space.
- *World Units*—Units for the World Coordinate System.

In typical applications, represented by Figure 1-2, a portion of the World Space determined by the Window is plotted. The Window is set using World Units. The location of the plot is determined by the location of the Viewport. The Viewport is set using Device Units. Then graphics commands (MOVEs and DRAWs) are issued in terms of World Units and cause the Plotter to produce a plot within the Viewport.

The typical application just described occurs when the Graphic Units selection is the default selection of World Units. One other Graphic Units selection, called Device Units, is available. With Device Units selected, graphics commands are issued directly in terms of Device Units. Coordinates for MOVE and DRAW commands are referenced to the Page and are not affected by Window and Viewport settings.

The Plotter allows three types of Device Units:

- Addressable Device Units (ADUs)
- Graphic Device Units (GDUs)
- Millimeters (mm)

ADUs are used with Serial devices, such as the TEKTRONIX 4010 Series terminals; GDUs are used with GPIB devices, such as the TEKTRONIX 4050 Series Graphic Systems. The mm selection provides an additional set of Device Units to relate actual physical size of millimeters.

Displaying graphics on the Plotter is very similar to displaying graphics on the screens of display devices. At power-up, the Plotter automatically selects the type of units that correspond to your installed interface. Thus, the 4663 allows you to use the same numbering system on the Plotter as you are using on your display device.

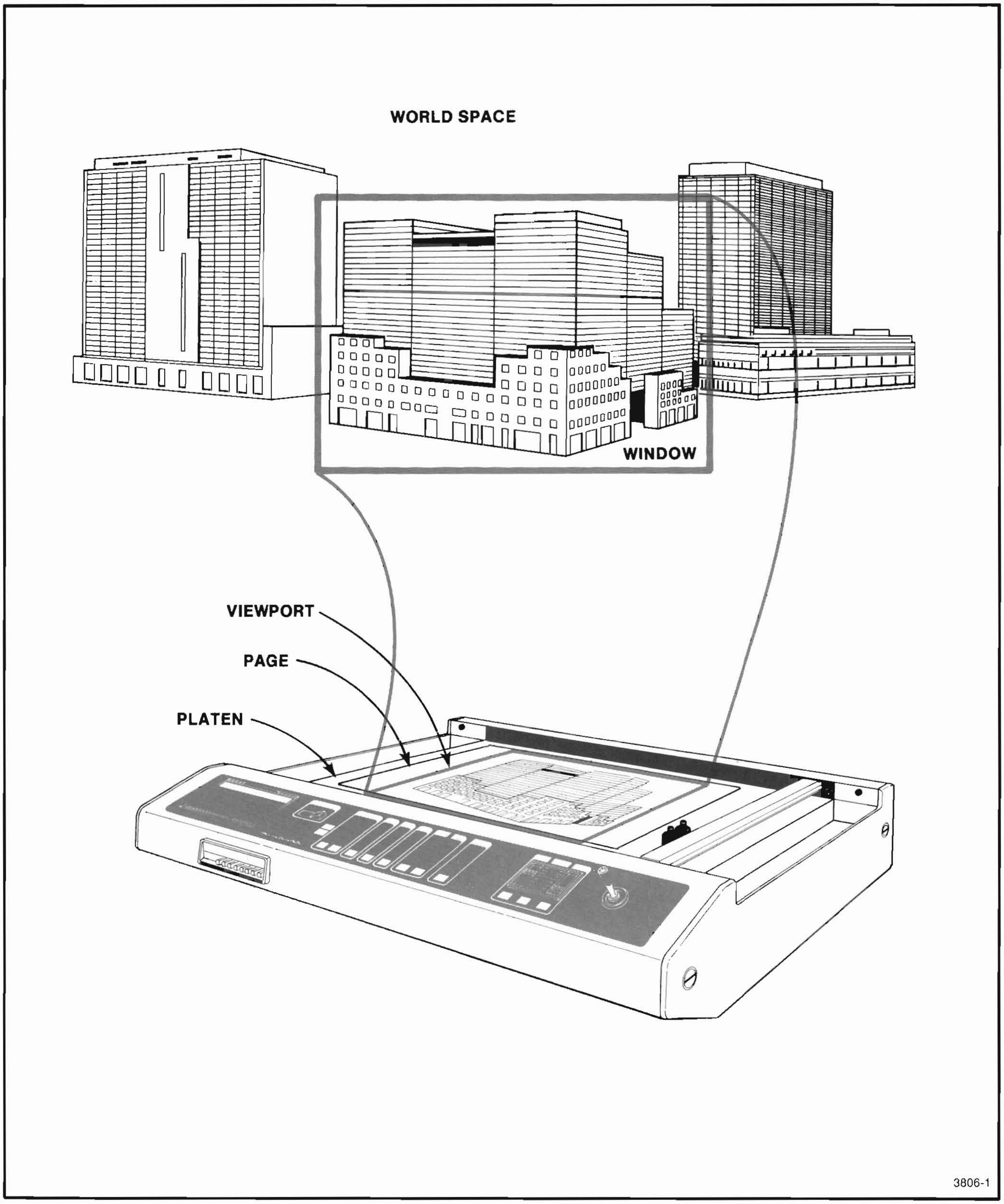


Figure 1-2. The Plotter's Graphics System.

PLOTTER GRAPHICS SYSTEM—SPECIFIC

The following description of the Plotter Graphics System is separated into two topics—*Plotter Domain* and *User Domain*. Additional general plotting information is provided later in this section.

PLOTTER DOMAIN

The Plotter Domain discussion covers the following parts of the Plotter: Pen Carriage, Platen, Platen Clipping Boundary, Page, Viewport, the Device Coordinate System, and Device Units.

Pen Carriage

The Pen Carriage consists of three parts—two pen holders and a crosshair cursor. Since any of these three parts may be physically positioned over the media at a point specified by plotting coordinates, this manual uses the generic term pen carriage unless it is necessary to specifically identify one of the pens or the crosshair cursor.

The front panel Pen Control switches and the SELECT PEN command select either of the pens (Pen 1 or Pen 2) or the crosshair cursor to be at the location specified by the plotting coordinates. For plotting, the selected pen moves to positions set by the plotting coordinates. When neither pen is in position, the crosshair cursor indicates the position specified by the plotting coordinates.

Platen and Platen Clipping Boundary

The platen is the large flat surface of the 4663 Plotter onto which the media is placed. The maximum area on the platen that is available for plotting is outlined by the Platen Clipping Boundary. Both pens have access to this area. See Figure 1-3.

Page

The Page is a rectangular area on the platen which can be used for plotting. The Page provides a plotting area (or viewing area) similar to the full screen on a terminal. Usually the Page size corresponds to the size of the media in use.

The operator may select one of 12 pre-defined Initial Page sizes which conform to standard media sizes. These 12 sizes are selected using the Initial Page Size and Inital Page Format lines on the Parameter Entry Card, as described in the *4663 Operator's Manual*. Normally, the media has margins extending beyond the Page boundaries. The Page boundary is shown in Figure 1-4.

Pages may also be established by the operator using front panel Set Page functions. One such user-defined Page may be saved by using the SAVE USER DEFINED selection on the Initial Page Size line of the Parameter Entry Card. Pages may not be set by the host, but the host can issue the DEVICE RESET (3) command to reset the Page to the Initial Page established by the current Parameter Entry Card selections. The selections which establish the Initial Page include the following lines: Media Form, Initial Page Size, Initial Page Format, Page Orientation, and Initial Aspect Ratio.

Pre-defined or user-defined Pages may extend outside the Platen Clipping Boundary. The portion of the plot within the Platen Clipping Boundary is drawn; the portion outside the boundary is not drawn.

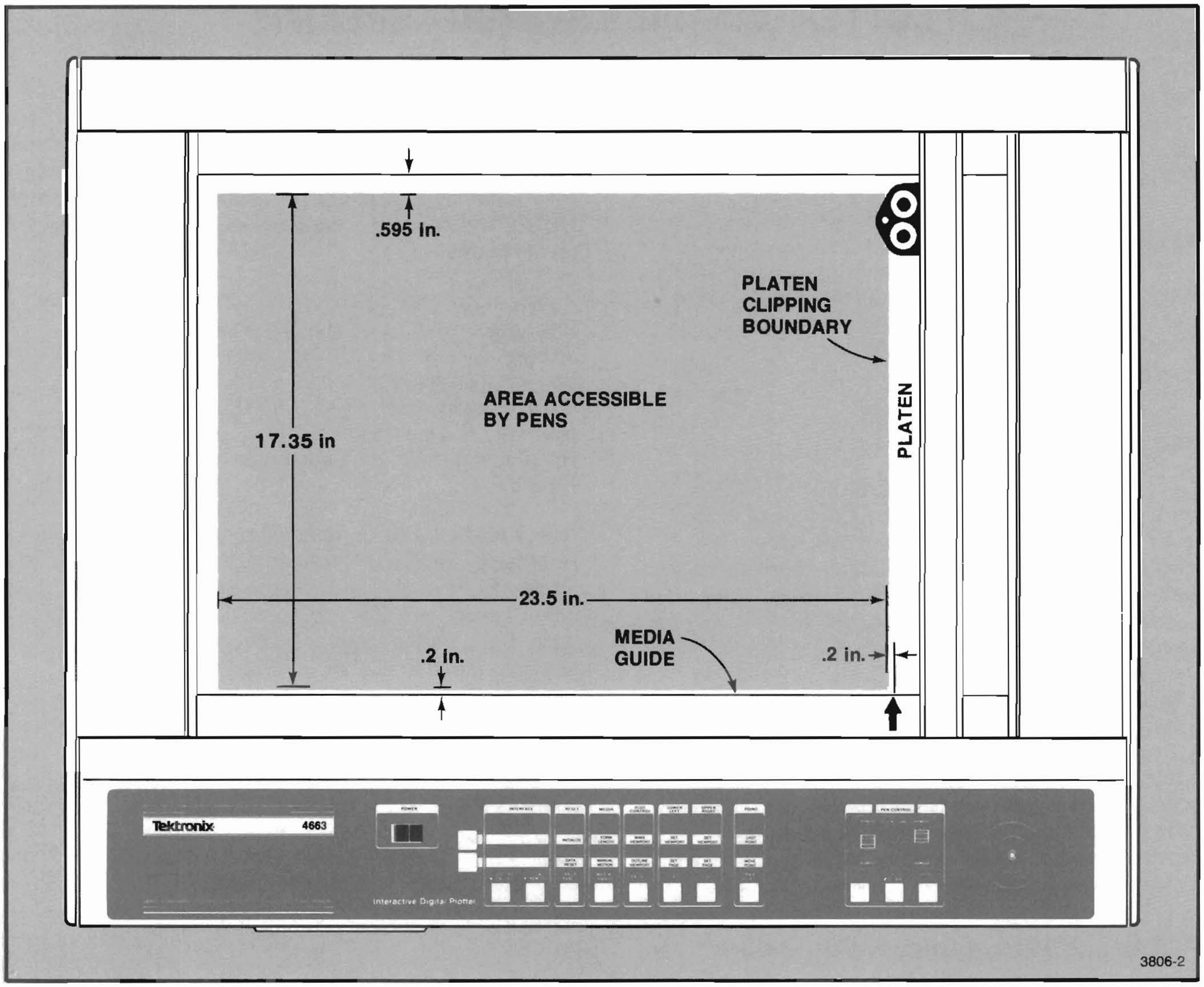


Figure 1-3. Platen and Platen Clipping Boundary.

The *4663 Operator's Manual* describes other capabilities associated with setting Pages from the front panel switches and Parameter Entry Card. One of these involves choosing VERTICAL or HORIZONTAL orientation on the Page Orientation line of the Parameter Entry card. With HORIZONTAL orientation, the horizontal axis (X-axis) is parallel to the media guide. With VERTICAL orientation, the Y-axis is parallel to the media guide.

Another capability involves Page reversals. The LOWER LEFT and UPPER RIGHT switches on the front panel are used to set the physical upper-right corner to be down and/or to the left of the physical lower-left corner. Procedures are included in the "Getting Started" section of the *4663 Operator's Manual* which describe these operations.

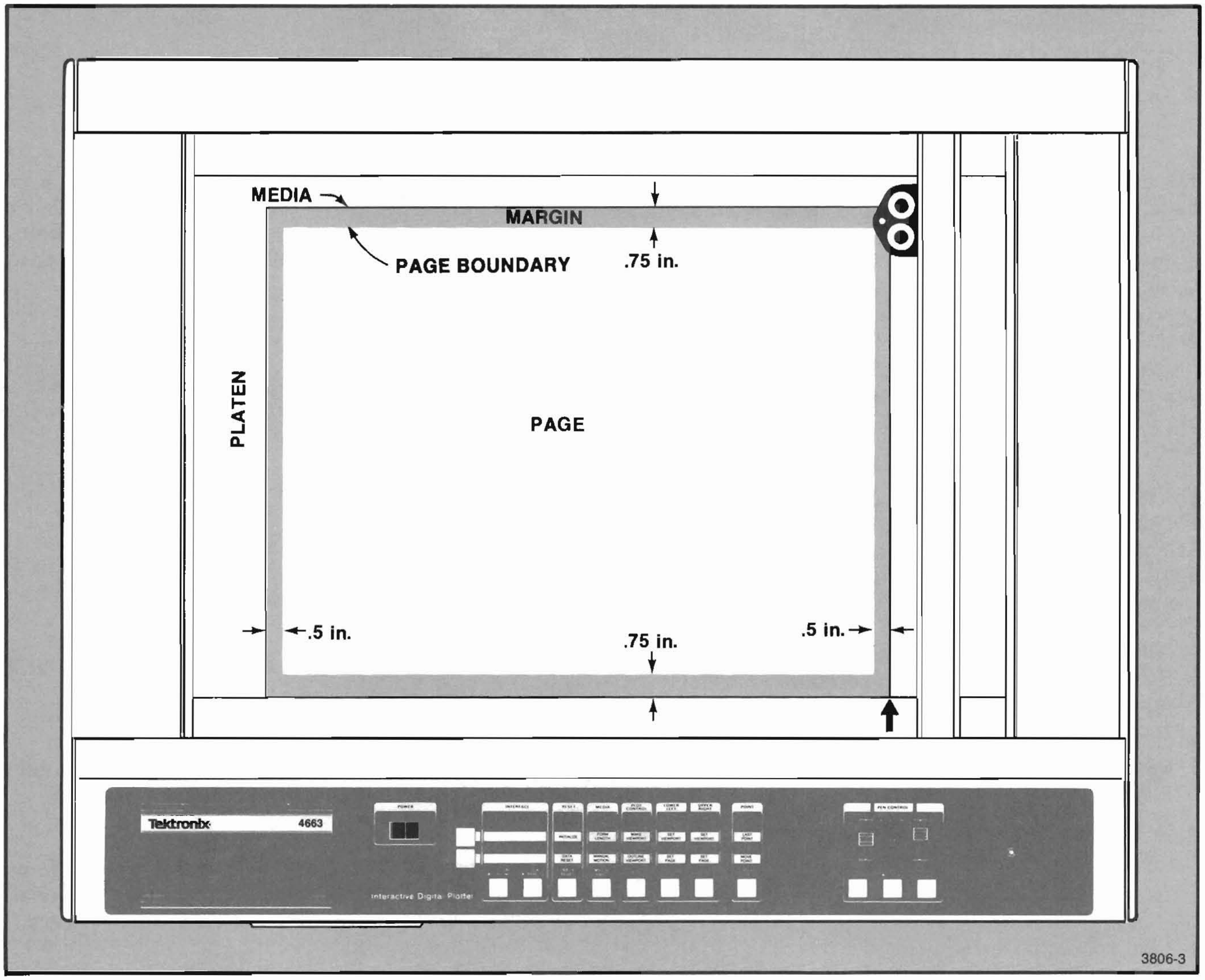


Figure 1-4. Page, Media, and Platen (Shown for C-Size Media, Drafting Boundaries, and Sheet Mode).

Device Coordinate System and Device Units

The Device Coordinate System is a Cartesian coordinate system for the Page. The origin (0,0) of the coordinate system is the lower-left corner of the designated Page. The physical lower-left corner may be at any corner because of Page orientation and Page reversal situations mentioned earlier.

Three Device Coordinate Systems exist for three types of Device Units:

- Addressable Device Units (ADUs)
- Graphic Device Units (GDUs)
- Millimeters (mm)

The default Device Unit selections are interface dependent. When using a Serial interface, ADUs are the default Device Unit; GDUs are the default Device Unit when the Plotter is attached to the controller through a GPIB interface. Default Device Units occur at power-up or after the DEVICE RESET, SELECT COMMAND/RESPONSE FORMAT, or INTERFACE PARAMETER RESET commands are issued. However, either ADUs, GDUs, or millimeters can be selected as the Device Unit by the SELECT DEVICE UNITS command.

The ranges for each of the three types of Device Units are based on the aspect ratio of the Initial Page and are established so that they are the same physical length in the horizontal and vertical directions. These ranges are as follows:

- ADUs—0 to 4096 on the larger (usually horizontal) Page axis and from 0 to a proportionally smaller number on the second axis.
- GDUs—0 to 100 on the shorter (usually vertical) Page axis and from 0 to a proportionally larger number on the second axis.
- Millimeters (mm)—0 to the actual physical length for each axis of the Initial Page.

These ranges are not changed when the Page is changed by front panel Set Page operations. However, Set Page operations can change the physical length of the Device Units and make them unequal for each axis. If this occurs, a subsequent DEVICE RESET (0 or 1) command or a SAVE USER DEFINED selection on the Parameter Entry Card establishes new Device Unit ranges according to the above rules for the revised Page.

Viewport

The Viewport is the rectangular area of the Page onto which the plot is drawn. Viewport boundaries are always on or within the Page boundaries. Coordinates for the Viewport are expressed in Device Units.

Viewports are specified or directly set in three ways:

- Using the Initial Aspect Ratio line on the Parameter Entry Card (this also re-establishes the Initial Page).
- Issuing the SET VIEWPORT command.
- Using the front panel Set Viewport function.

As with Page reversals, Viewports can be set to have the upper-right corner physically or numerically lower and/or to the left of the lower-left corner.

The Plotter's default or Initial Viewport is established in one of three ways: at power-up; when the Plotter receives the DEVICE RESET (0, 1, or 3) command; or when new selections are made on the Parameter Entry Card lines Media Form, Initial Page Size, Initial Page Format, Page Orientation, or Initial Aspect Ratio.

The Initial Viewport depends on the Initial Aspect Ratio selection on the Parameter Entry Card. The Initial Viewport and Initial Page lower-left corners are at the same location, and the Viewport fills as much of the Page as the Initial Aspect Ratio line setting allows. With FULL PAGE selected on the Initial Aspect Ratio line, the Initial Viewport and the Initial Page are identical.

Changes and distortions in the Page as a result of front panel Set Page operations cause proportional physical, but not numerical, changes in the Viewport.

USER DOMAIN

The User Domain includes the World Space, the World Coordinate System and World Units, the Window, and transformation-of-data capability.

World Space

The World Space is the space associated with an object or picture. World Space can contain graphs, maps, physical objects, mechanical drawings, or other pictures.

World Coordinate System and World Units

The World Coordinate System is a Cartesian coordinate system for World Space. World Units are used with the World Coordinate System. In general, World Units are established for the picture or object to be displayed and can be dimensional in any unit of measure, such as gallons, seconds, dollars, light years.

A default set of World Units is established at power-up or when the DEVICE RESET (0, 1, or 3) command is issued. The default set is either ADUs when using the Serial interface or GDUs when using the GPIB interface.

NOTE

The ADUs or GDUs for the World Coordinate System can be of different lengths than the ADUs or GDUs for the Device Coordinate System. This is discussed in the next subsection.

In addition, a default set of World Units is established for selections on any of the five Parameter Entry Card lines labeled: Media Form, Initial Page Size, Initial Page Format, Page Orientation, and Initial Aspect Ratio. The default set corresponds to the existing Device Units selection, which may include millimeters in addition to ADUs and GDUs.

Window

The Window is a rectangular portion of the World Space that is projected onto the Viewport. The Window is set using World Units. It does not change when the SET VIEWPORT command is issued or with front panel Set Viewport and Set Page functions.

The SET WINDOW command can be used to assign appropriate coordinates to the Window. For example, the SET WINDOW command might be used to match the Window coordinates to a graph with units of inches, miles, gallons, etc. The units and range for each axis would be different. For example, the range on one axis may be from 5 to 10 gallons and 85 to 105 seconds on the other.

Unlike Pages and Viewports, the Window coordinates cannot be defined so that the lower-left corner is numerically larger than the upper-right corner. However, selections on the Initial Axis Orientation line of the Parameter Entry Card other than the left-most selection (X-axis horizontal pointed right and Y-axis vertical pointed up) provide axis reversals and interchanges.

Initial Windows exist for each of the three Device Units selections of ADUs, GDUs, and millimeters. If the Device Units selection is changed, the Initial Window is set by the rules for the new Device Units selection.

NOTE

The range of Initial Window values is determined by the Initial Viewport dimensions as opposed to the Initial Page dimensions of the device. The Initial Aspect Ratio line on the Parameter Entry Card has selections that allow the Initial Viewport to be different than the Initial Page. The Viewport and Page are the same when FULL PAGE is selected on the Initial Aspect Ratio line. With other selections, the physical lengths of ADUs and GDUs for default World Units may be different than ADUs and GDUs for Device Units.

The numerical ranges for each Initial Window are established so that the World Units are the same physical lengths on the horizontal and vertical axes. The 0,0 location is projected on the Initial Viewport lower-left corner. The ranges are as follows:

- ADUs—0 to 4096 on the longer Initial Viewport axis and from 0 to a proportionally smaller number on the second axis.
- GDUs—0 to 100 on the shorter Initial Viewport axis and from 0 to a larger number on the second axis.
- Millimeters (mm)—0 to the actual physical lengths for each axis of the Initial Viewport.

These Initial Window ranges are established so that the Plotter appears similar to other display devices such as the TEKTRONIX 4010 Series for ADUs and the TEKTRONIX 4050 Series for GDUs. In fact, the Initial Window using the default World Units may match other devices closer than the Initial Page using the default Device Units because the Initial Window is based on the Initial Aspect Ratio selection on the Parameter Entry Card. For example, a 4X:3Y (actually 4096X: 3124Y) selection provides an Initial Window that matches the screen coordinates of a TEKTRONIX 4014 terminal, regardless of the Initial Page.

The Initial Window is important in providing units for specifying alphanumeric character size and spacing and dash pattern length, even though the actual Window can be different from the Initial Window.

Any selection other than the left-most selection on the Initial Axis Orientation line of the Parameter Entry Card changes the 0,0 location and/or the direction of the X- and Y-axes. However, the rules above still apply to determine the axis ranges for these cases.

Transformations

User data describing the picture or object can be modified by using *transformations*. The available transforms allow scaling, rotation, translation, skewing, and other special purpose operations. See Section 12, *Transform Commands*, for details.

Transforms are cleared on power-up, or through any other operation that redefines or resets the Window or Viewport.

GENERAL PLOTTING INFORMATION

GRAPHIC UNITS

Graphic Units, which may be either World Units or Device Units, are selected by the SELECT GRAPHIC UNITS command to establish the coordinate system for subsequent MOVE and DRAW commands. The SELECT GRAPHIC UNITS command is also used to independently select Graphic Units for digitizing (discussed shortly).

The default Graphic Units selection is World Units, which occurs at power-up or when the DEVICE RESET command is issued. When World Units are selected, the data is modified as a result of any transforms that have been entered. When Device Units are selected, the coordinates for MOVEs and DRAWs are referenced to the Page and are not affected by transforms or Window and Viewport settings.

COORDINATE TYPE

MOVE and DRAW commands can be specified using relative or absolute coordinates as determined by the SELECT COORDINATE TYPE command. *Absolute coordinates* are referenced to the current coordinate system. *Relative coordinates* are referenced to the current pen carriage position and are entered in terms of displacement in the X- and Y-directions from the current pen carriage position.

The default selection is absolute coordinates. This occurs on power-up, and after the DEVICE RESET command.

LINE TYPE

The SELECT LINE TYPE command can be used to select solid, fixed dash-pattern-length, variable dash-pattern-length, and point line types for drawing operations. Line type selections other than solid are available only when the Graphic Units selection is World Units. The default line type is solid line and occurs at power-up and when the DEVICE RESET command is issued.

Some operations, such as axis generation and the printing of alphanumeric characters, use solid lines regardless of the line type selection. More information about line types and related commands is contained in the Graphics Commands section.

CLIPPING

Clipping refers to the process by which the portion of a plot that falls outside of a clipping boundary is not plotted. This process does not distort the portion of the plot within the clipping boundary. When the pen is commanded to draw to a position outside the boundary, it draws only to the intercept point at the boundary. When commanded to draw from a position outside the boundary to a position within, the pen begins drawing at the intercept point of the new line and boundary.

There are two *active* clipping boundaries: the Platen Clipping Boundary and either the Page Clipping Boundary or the Viewport Clipping Boundary. The *Platen Clipping Boundary* outlines the maximum area accessible by the pen carriage. The *Page Clipping Boundary* outlines the edges of the Page. The *Viewport Clipping Boundary* outlines the Viewport edges. The SELECT CLIPPING CONTROL command is used to select either Page or Viewport Clipping Boundaries.

Normally, the Page and Viewport are inside the Platen Clipping Boundary. However, if an edge of an active Page or Viewport Clipping Boundary is outside, then clipping occurs at the corresponding edge of the Platen Clipping Boundary.

The clipping control of the SELECT CLIPPING CONTROL command selection is independent of the Graphic Units selection. For example, MOVEs and DRAWs can be specified using Device Units, but clipping occurs at the Viewport boundary. Similarly, clipping can occur at the Page boundaries even when World Units are specified.

The default clipping control selection is at the Viewport Clipping boundary and occurs at power-up and after the DEVICE RESET (0, 1, or 3) command is issued.

DIGITIZING

In addition to plotting, the 4663 Plotter can also be used for digitizing (also referred to as graphic input); that is, it can send coordinates to the host.

NOTE

Coordinates for digitizing are always absolute, regardless of the coordinate type selection of the SELECT COORDINATE TYPE command.

The SELECT GRAPHIC UNITS command provides a digitization selection. When Device Units are selected, coordinates are reported to the host in terms of the Device Coordinate System and using the current Device Units (ADUs, GDUs, or mm). When World Units are selected, the current Window, Viewport, and transforms all affect the response so that coordinates are reported in World Units. The Graphic Units selection for digitization can be different than the selection for commanded MOVEs and DRAWs. For example, the selection for MOVEs and DRAWs can be World Units while the digitization selection is Device Units.

Points digitized using Device Units are restricted to be within the Page boundaries. Coordinates lying outside are interpreted by the Plotter as being on the nearest Page boundary. An exception to this occurs when the Page Orientation line on the Parameter Entry Card is set to VERTICAL. For this special case, the actual coordinates are reported even if they are outside the Page boundaries.

The default Graphic Units for digitizing is World Units. This occurs on power-up and when the DEVICE RESET command is issued.

Several methods to digitize points are discussed in Section 13 and also in the READ STATUS command description.

DATA RESOLUTION

The maximum resolution between points on the Plotter is .001 in (0.0254 mm). Real numbers that imply greater resolution can be handled by the Plotter's internal software. However, the positioning of the pen and the reporting of manually located coordinate positions are limited to the maximum resolution.

ERROR REPORTING

The 4663 Plotter reports the occurrence of errors by displaying a flashing or steady Reset light and sounding the bell. Lights on the Parameter Entry Module can be interpreted as an error number to determine the source of the error. Appendix J tabulates the error numbers and error sources.

Procedures to read the error information provided by the Paramater Entry Module lights are given in the *4663 Operator's Manual* and in this manual under the description of the READ ERROR command.

PAGE SCALING

For some applications, the physical length of the ADU or GDU must be known. In general, this can be found by looking up or measuring the length of each axis and dividing by the number of Device Units. Appendix G includes tables listing the ranges for default Page sizes. An example is provided here for determining Page scaling.

A C-sized, drafting Page has 4096 ADUs along its longest axis of 21 in (533.40 mm). Therefore the Page scaling is:

$$\frac{21\text{ in}}{4096\text{ ADUs}} = .00513\text{ in/ADU or }195.05\text{ ADUs/in}$$

or

$$\frac{533.40\text{ mm}}{4096\text{ ADUs}} = .13022\text{ mm/ADU or }7.68\text{ ADUs/mm}$$

This same Page has 100 GDUs along its shortest axis of 15.5 in (393.70 mm). Its Page scaling is:

$$\frac{15.5\text{ in}}{100\text{ GDUs}} = .155\text{ in/GDU or }6.45\text{ GDUs/in}$$

or

$$\frac{393.70\text{ mm}}{100\text{ GDUs}} = 3.937\text{ mm/GDU or }.254\text{ GDUs/mm}$$

Other Page scaling can be calculated in a similar manner.

VIEWPORT SCALING

Viewport scaling is done in a manner similar to Page scaling. Default Viewport dimensions are also given in Appendix G.

For example, a C-sized, drafting Page with initial aspect ratio of 3X:4Y (3124X:4096Y) is considered. In ADUs, its longest axis is 15.5 in (393.70 mm), which corresponds to 4096 ADUs. The Viewport scaling is:

$$\frac{15.5\text{ in}}{4096\text{ ADUs}} = .00378\text{ in/ADU or }264.3\text{ ADUs/in}$$

or

$$\frac{393.70\text{ mm}}{4096\text{ ADUs}} = .0961\text{ mm/ADU or }10.04\text{ ADUs/mm}$$

For GDUs the shortest axis is 15.5 in times 3124/4096 which equals 11.82 in (300.27mm) or 100 GDUs. The Viewport scaling is:

$$\frac{11.82\text{ in}}{100\text{ GDUs}} = .1182\text{ in/GDU or }8.459\text{ GDUs/in}$$

or

$$\frac{300.27\text{ mm}}{100\text{ GDUs}} = 3.0027\text{ mm/GDU or }.3330\text{ GDUs/mm}$$

Section 2

FIRST-TIME OPERATION THROUGH THE SERIAL INTERFACE

NOTE

If using a GPIB interface, skip to Section 3, First-Time Operation Through the GPIB Interface.

INTRODUCTION

The procedures offered in this section provide the opportunity to experience many of the command sequences firsthand, thereby demonstrating the Plotter's actions in response to various commands. This familiarization procedure does not describe all commands or sequences of commands, but instead provides a general overview of Plotter operation.

It is assumed that the reader is already familiar with the information contained in the *Getting Started* section of the *4663 Operator's Manual.* That section includes media and pen loading instructions and gives an introduction to the Plotter's front panel controls and indicators.

The conventions used in this section, and throughout the manual, are described in detail in Section 6, *Introduction to Commands and Arguments*. For your convenience, a few of the more common conventions are discussed here:

- The *attention character* (AT), which is the first character in each command of this procedure, is the exclamation point (**!**). If necessary, you can select any of the three following characters to be the attention character: **EC**, **Λ**, or **SY**. To change the attention character, change the selection on the Attention Character line of the Parameter Entry Card and then use **EC**, **Λ** , or **SY** instead of "!". (**EC** and **SY** are shown as ESC and SYN, respectively, on the Parameter Entry Card.
- Characters shown in bold type must be transmitted exactly as shown.
- When communicating with the Plotter, the host should only transmit space characters indicated by **SP**. Spaces between command characters are included to improve readability of the text and should not be interpreted as space characters.
- **CR** stands for Carriage Return (or Return).

NOTE

Paper media and fiber-tip pens must be installed before performing these procedures. Refer to the 4663 Operator's Manual for media and pen loading instructions.

PROCEDURE

The following procedure assumes C-size media is installed.

I. Connect the Plotter to the terminal.

   a. Check that only the paper media is on the platen. All other materials should be removed.

   b. Turn off the Plotter (if not off already).

   c. Attach a terminal, such as a TEKTRONIX 4010 Series, 4006-1, to the Serial port on the rear panel of the Plotter marked TERMINAL.

   d. Turn on both the Plotter and the terminal, and initialize the terminal, if necessary. (After a few seconds, press the PAGE or ERASE key on the TEKTRONIX terminal to erase the screen.) All familiarization commands will be issued from the terminal keyboard.

   e. Set the terminal as follows:

      (1) Baud rates (transmit and receive) — 300 baud recommended for this procedure.

      (2) Echo — local.

      (3) If a TEKTRONIX 4024, 4025, or 4027 terminal is used, do not establish a Workspace; use Monitor mode. Also, select COMMAND LOCK OUT.

2. Set the Plotter's front panel switches.

   a. Set the following Plotter front panel switches (others should be ignored):

      (1) INTERFACE switches to ONLINE LOCAL (see Figure 2-1).

      (2) Select either PEN 1 OR PEN 2 with the appropriate SELECT switch in the Pen Control group.

3. Make Parameter Entry Card selections as follows:

NOTE

Push the Parameter Entry Card fully into the Plotter before making any selections.

NOTE

For this procedure, you need make selections only for the Parameter Entry Card lines listed here.

   a. Parameter Setup Select — SETUP 1. (If the INIT light on the Parameter Entry Module is ON, momentarily push the Parameter Entry Card fully in before resuming this procedure.)

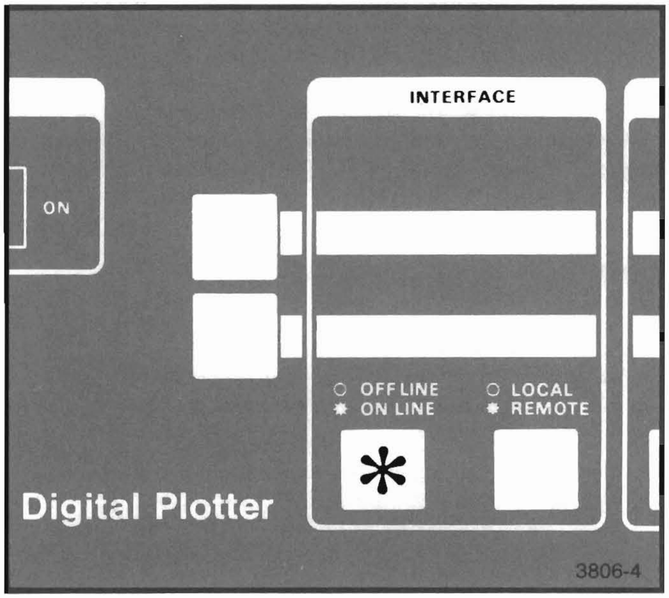


Figure 2-1. INTERFACE Switches Set to ONLINE/LOCAL.

b. Media Form — SHEET. (Set to ROLL if the Media Advance Option is installed and roll media is being used.)

c. Initial Page Size — C.

NOTE

You can select any of the other pre-defined sizes (B, A, A2, A3, or A4) on this line and you can select the other format (GRAPHING) on the next line. If you select a new combination, you may observe a slight difference between the plotted results and those shown in Figure 2-2.

d. Initial Page Format — DRAFTING.

e. Page Orientation — HORIZONTAL.

f. Initial Aspect Ratio — FULL PAGE.

g. Initial Axis Orientation — Select leftmost column.

h. Line Quality — PREVIEW.

i. Pen Parameter Access — PEN 1.

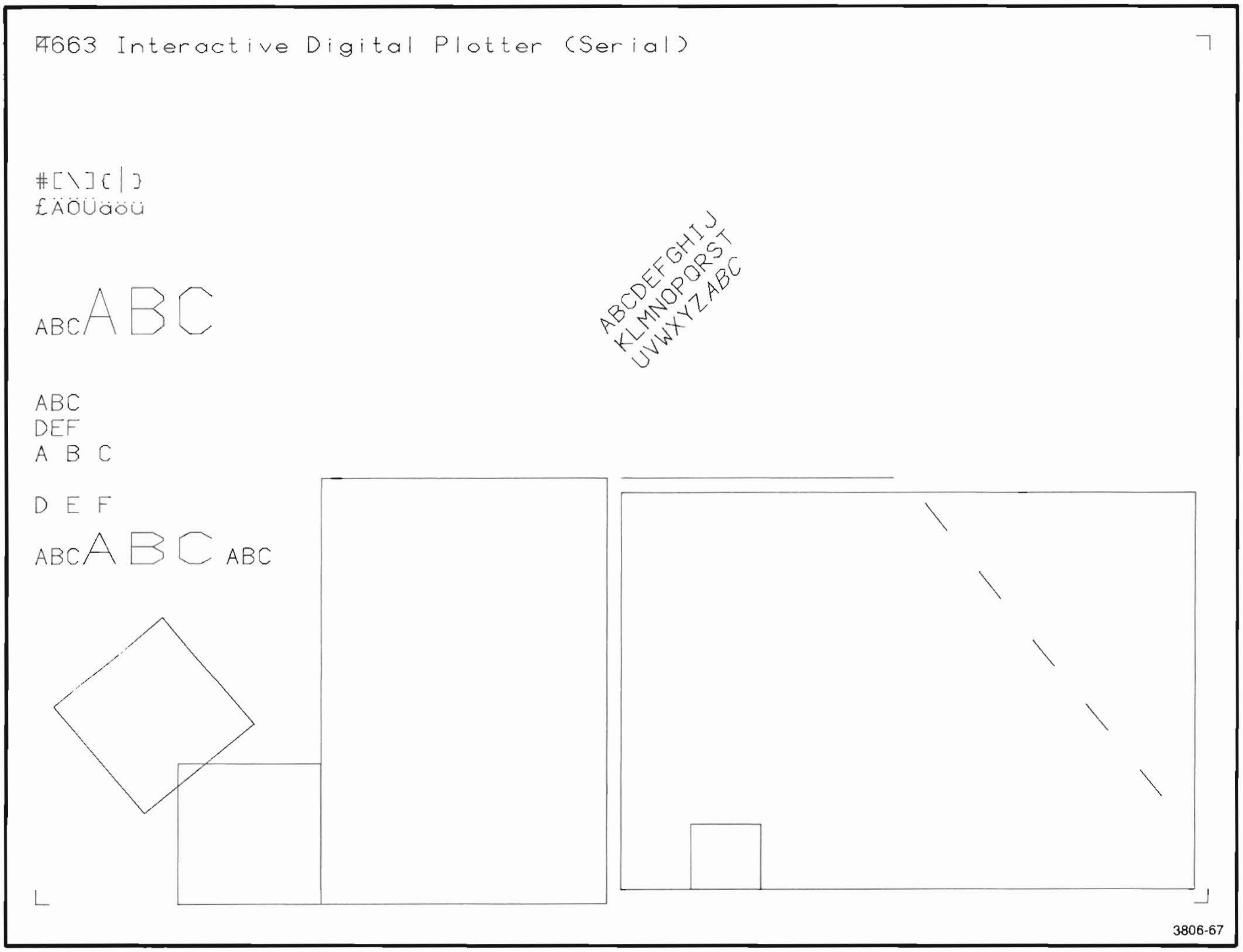


Figure 2-2. Plot Produced by Section 2 Procedures.

j. Pen Type — FIBER TIP.

NOTE

To set the Pen Type for Pen 2, select PEN 2 on the Pen Parameter Access line, and then make the FIBER TIP selection on the Pen Type line. The next two lines, Pen Pressure and Pen Velocity Limit, are set automatically.

k. Alpha Character Quality — NORMAL.

l. Interface Select — 1. (If the Interface Select line selection is changed, it is necessary to momentarily push the Parameter Entry Card fully in before resuming.)

m. Initial Command/Response Format — 2.

NOTE

The next three lines, GPIB Device Address, Interface Mode, and Interface Functions, are for use only with the GPIB interface and need not be set at this time.

n. Serial Device Address — A.

o. Receive Baud Rate - 300 baud recommended for this procedure. (Selected baud rate must match baud rate set on the terminal.)

p. Transmit Baud Rate — 300 baud recommended for this procedure. (Selected baud rate must match baud rate set on the terminal.)

q. Transmit Baud Rate Limit — FULL SPEED.

r. Character Format — 8 DATA BITS/CHAR and 1 STOP BITS. (Make two selections.)

s. Receive Parity/Transmit Parity — IGNORE/LOGIC 0.

t. Communications Control Mode — FULL DUPLEX.

NOTE

It is not necessary to make a selection on the DC1/DC3 Control line at this time.

u. Interface Functions — CR GENERATES LF. (Press either switch.)

v. Attention Character — !

w. Output Terminator — NONE.

x. Push the Parameter Entry Card back into the Plotter. No other selections are required at this time.

4. Execute the following samples of Plotter commands.

NOTE

In this manual, the ASCII Space character is always represented by $^{S}_{P}$. The use of spacing within command descriptions is for clarity of presentation only. Do not send a space character in place of spacing. The following example contains two space characters:

!A "P Three$^{S}_{P}$word$^{S}_{P}$example" $^{C}_{R}$

a. Issue the PLOTTER ON command:

!A E $^{C}_{R}$

This command causes the Plotter to execute subsequent commands.

b. Move the pen carriage to a point near the upper-left corner of the Page:

!A AH $^{C}_{R}$

c. Type the following text and note that the characters (upper and lowercase) are printed (some terminals are restricted to uppercase characters only):

4663$^{S}_{P}$Interactive$^{S}_{P}$Digital$^{S}_{P}$Plotter $^{S}_{P}$(Serial)$^{C}_{R}$

d. Transmit the following MOVE command to move the pen carriage to a point near the center of the Page:

!A X 2050,1500 $^{C}_{R}$

e. To draw a horizontal vector from this point, send a DRAW command:

!A Y 3000,1500 $^{C}_{R}$

f. To move two character spaces to the right and down one line space, type a CHARACTER MOVE command:

!A AM 2,—1 $^{C}_{R}$

5. Draw Dashed Line Vectors.

a. Transmit a SET DASH PATTERN command:

!A BD 2,3 $^{C}_{R}$

This determines the proportion of blank line to drawn line.

b. Establish a dash pattern length of 300 ADUs using the SET DASH PATTERN LENGTH command:

!A BS 300 $^{C}_{R}$

c. Transmit a SELECT LINE TYPE command to establish a fixed dash-pattern-length line type selection:

!A BL 1 CR

d. Issue a DRAW command to cause the Plotter to draw a dashed line from the current pen carriage position to the point determined by the command coordinates:

!A Y 4000,300 CR

e. To reset the Plotter to solid line type, transmit a SELECT LINE TYPE command:

!A BL 0 CR

6. Change the Font Selection.

a. Transmit a MOVE command to position the pen on the left edge and down from the top:

!A X 0,2500 CR

b. Type as many of the following characters, shown for Font 0, as your terminal permits and notice that they are printed as shown for Font 0 in Figure 2-3.

#[\]{|} CR

c. To select Font 2, transmit a SELECT STANDARD ALPHA FONT command:

!A T 2 CR

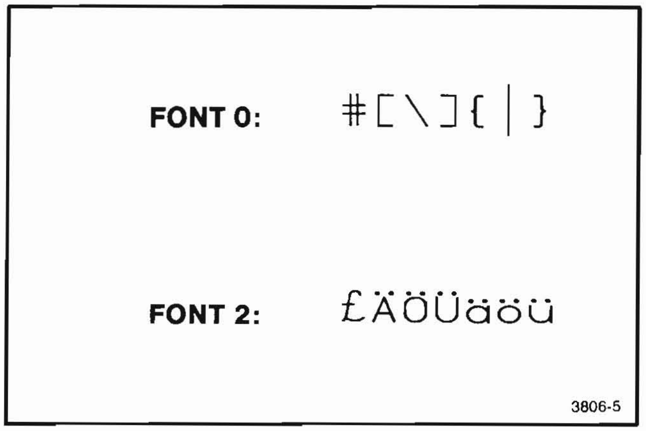


Figure 2-3. Fonts 0 and 2.

d. Type as many of the following characters, shown for Font 0, as your terminal permits and notice that printed characters appear as shown for Font 2 in Figure 2-3:

#[\]{|} CR

e. Set the alpha parameters back to default:

!A V CR

7. Change the printed character size.

a. Transmit a MOVE command to position the pen to the left edge and down from the top:

!A X 0,2000 CR

b. Type the following characters:

ABC

c. The size of the alphanumeric printing can be changed by transmitting an ALPHA SIZE command:

!A BZ 112,161 CR

d. Type the following characters:

ABC CR

The printed characters are about three times as large as they were for Step 7b.

e. Set the alpha parameters back to default:

!A V CR

8. Change the alpha ratio.

a. Type the following:

ABC CR
DEF CR

b. Change the alpha ratio:

!A BI 3,3.27 CR

c. Once again type the following characters:

ABC CR
DEF CR

The Plotter should print characters of the same size, but with a distance from the origin of one character to the origin of the next character of twice the default distance; the distance from the baseline of one line of printing to the baseline of the next line of printing is about twice the default distance.

d. Reset the alpha parameters to their default values:

!A V CR

9 Change the alpha scaling factors.

a. Type the following characters:

ABC

b. To alter the X and Y multiplication factors for Alpha characters, transmit a SET ALPHA SCALE command:

!A BH 3,2 $^{C}{}_{R}$

c Type the following:

ABC

Notice that the characters and character-to-character spacings are three times as wide and three times as high as they were before the SET ALPHA SCALE command in Step 9b.

d. To reset to the scaling factors, to the default condition, transmit a SET ALPHA SCALE command with magnification factors 1,1:

!A BH 1,1 $^{C}{}_{R}$

e. Type the following:

ABC $^{C}{}_{R}$

Notice that the characters are printed with default size and spacing.

10. Print characters on a diagonal line.

a. Move the pen carriage to a point above the center of the media:

!AX 2000,2000 $^{C}{}_{R}$

b. To set the printing angle (and the left margin), transmit a SET ALPHA ROTATION command:

!A J 45 $^{C}{}_{R}$

c. Type the following:

ABC

The characters are printed on a line rotated 45° from horizontal.

d. A right margin is established by transmitting a SET ALPHA MARGIN SEPARATION command:

!A BR 10 $^{C}{}_{R}$

e. Continue to type the alphabet (D, E, F, ...) and notice that after the 10th character has been typed on a line, the Plotter automatically executes a CARRIAGE RETURN ($^{C}{}_{R}$) and a LINE FEED ($^{L}{}_{F}$) to continue printing on the next line. Also, notice that both margins (left and right) are perpendicular to the printing direction (which in this case is at a 45° angle).

11. Change the slant of the printed characters.

a. Slanted characters can be printed by transmitting a SET ALPHA SLANT command:

!A BG 20 $^{C}{}_{R}$

b. Type the following:

ABC $^{C}{}_{R}$

Notice that these characters, although printed along the 45° angle printing direction are, in addition, slanted to the right 20°.

c. Reset the alpha parameters:

!A V $^{C}{}_{R}$

12. Sample Transform commands.

To illustrate some of the Transform commands, first draw a square box and then modify it with Transform commands.

a. Transmit a MOVE command to position the pen carriage at the lower-left corner of the box:

!A X 500,0 $^{C}{}_{R}$

b. To draw the box, transmit the following DRAW commands:

!A Y 1000,0 $^{C}{}_{R}$
!A Y 1000,500 $^{C}{}_{R}$
!A Y 500,500 $^{C}{}_{R}$
!A Y 500,0 $^{C}{}_{R}$

A box with 500 ADU sides will be drawn with its lower-left corner 500 ADUs to the right of the Viewport origin.

c To save the present (default) transform, transmit a SAVE CURRENT TRANSFORM command:

!A AX $^{C}{}_{R}$

d. To change the X-Y scaling factors for subsequent graphing, transmit a SET SCALE command:

!A AS 2,3 $^{C}{}_{R}$

e. Retransmit the commands in Steps 12a and 12b. Notice that this box is twice as wide and three times as high as the first box and is displaced to the immediate right of the first box. The lower-left corner of both boxes is 500,0, but 500 on the X-axis, after rescaling, is twice as far from the origin as it was when the first box was drawn.

f. To restore the default transform matrix and delete the scaled transform matrix from RAM storage, transmit a RESTORE PREVIOUS TRANSFORM command:

!A AY $^{C}{}_{R}$

13. Rotate the picture.

a. Transmit the SAVE CURRENT TRANSFORM command to save the present (default) transform matrix:

!A AX $^{C}{}_{R}$

b. To rotate the axes of the Viewport 40°, transmit a SET ROTATION command:

!A AR 40 $^{C}{}_{R}$

c. Then retransmit the commands in Steps 12a and 12b. Notice that the box is re-drawn rotated counterclockwise by 40°.

d. To restore the default transform matrix and delete the current transform matrix, transmit a RESTORE PREVIOUS TRANSFORM command:

!A AY $^{C}{}_{R}$

14. Establish a new Viewport.

a. To mark the Initial Viewport, transmit a MARK VIEWPORT command:

!A CM $^{C}{}_{R}$

b. To set the Viewport (without using the front panel controls or the Paramenter Entry Card), transmit a SET VIEWPORT command:

!A AV 2050,4050,50,1450 $^{C}{}_{R}$

c. To outline this new Viewport, transmit an OUTLINE VIEWPORT command:

!A CB $^{C}{}_{R}$

d. Redraw the box using the commands in Steps 12a and 12b. Notice that the box and the Viewport outline in Step 13c are reduced in size proportionately.

e. To restore the Initial Viewport, pull the Parameter Entry Card out to the Initial Page Size line and press the switch over C.

NOTE

Even though that switch's light will still be on, you must press the switch at this time. The Viewport size then becomes the same as the Initial Page size.

15. Disable one axis of the Joystick.

This command could be used, for example, to digitize the Y-coordinates of a graph at a known X-axis location.

a. To disable the X-axis of the Joystick, transmit a JOYSTICK DISABLE command, specifying the X-axis:

!A BJ 1 $^{C}{}_{R}$

b. Move the Joystick around and notice that it only causes the pen to move along the Y-axis.

c. Both axes of the Joystick are again enabled any time the host repositions the pen or another JOYSTICK DISABLE command is given, specifying that neither axis be disabled. Enter:

!A BJ 0 $^{C}{}_{R}$

16. Obtain digitized pen carriage coordinates.

    a. Pull the Parameter Entry Card out to the Initial Command/Response Format line and press the switch above "2".

    b. Position the pen near the center of the Viewport:

    ```
    !AX 2000,1500 CR
    ```

    c. To display the pen carriage coordinates (in ADUs) on the terminal screen, transmit a DIGITIZE command:

    ```
    !A M CR
    ```

    Notice that the first two of three numbers, written in scientific notation, indicate the X- and Y-coordinates, respectively, of the MOVE command in Step 16b.

    d. Use the Joystick to move the pen carriage. Then transmit another DIGITIZE command:

    ```
    !A M CR
    ```

    The new pen coordinates will be printed on the terminal screen.

This completes the familiarization procedure for the Serial interface commands.

Section 3

FIRST-TIME OPERATION THROUGH THE GPIB INTERFACE

NOTE

If you are using a Serial interface, refer to Section 2, First-Time Operation Through the Serial Interface.

INTRODUCTION

The procedures offered in this section provide the opportunity to experience many of the command sequences first-hand, thereby demonstrating the Plotter's actions in response to various commands. This familiarization procedure does not describe all commands or sequences of commands, but instead provides a general overview of Plotter operation.

It is assumed that the reader is familiar with the information contained in the *Getting Started* section of the *4663 Operator's Manual. Getting Started* includes media and pen loading instructions and gives hands-on instruction for the Plotter's front panel controls and indicators.

The conventions used in this section, and throughout the manual, are described in detail in Section 6, *Introduction to Commands and Arguments*. For your convenience, a few of the more common conventions are listed here:

- Characters shown in bold type must be transmitted exactly as shown.
- When communicating with the Plotter, the controller should only transmit space characters indicated by SP. Blank spaces between command characters are included to improve readability of the text.
- CR stands for Carriage Return (or Return).

NOTE

Paper media and fiber-tip pens must be installed while performing these procedures. The following procedure assumes C-size sheet media and fiber-tip pens are installed See the 4663 Operator's Manual for media and pen loading instructions.

PROCEDURE

1. Connect the Plotter to the GPIB controller.
   a. Check that only media is on the platen; all other materials should be kept off the platen at all times.
   b. Turn off the Plotter (if not off already).
   c. Attach an interconnecting cable between the GPIB controller (such as a TEKTRONIX 4050 Series) and the Plotter's GPIB port (marked IEEE 488-1978).
   d. Turn on the GPIB controller (hereafter called controller). After a few seconds, press the HOME/PAGE key on the TEKTRONIX 4050 Series controller to erase the screen.

2. Set the Plotter's front panel switches.
   a. Turn the POWER switch on and wait 5 to 10 seconds for the Plotter to initialize.
   b. Check that the INTERFACE switches are set to ONLINE/REMOTE (both lights on).
   c. Select either PEN 1 or PEN 2 with the appropriate (Pen Control) SELECT switch.

3. Make the Parameter Entry Card selections.

NOTE

Push the Parameter Entry Card fully in before making any selections.

NOTE

You need select only the Parameter Entry Card lines listed here.

   a. Parameter Setup Select — SETUP 1. (If the INIT light on the Parameter Entry Module is on, momentarily push the Parameter Entry Card fully in before resuming this procedure.)
   b. Media Form—SHEET. (Set to ROLL if the Media Advance Option is installed and roll media is installed.)
   c. Initial Page Size — C.

NOTE

You can select any of the other pre-defined sizes (B, A, A2, A3, or A4) on this line and you can select the other format (GRAPHING) on the next line. If you select a new combination, you may observe a slight difference between the plotted results and Figure 3-1.

   d. Initial Page Format — DRAFTING.
   e. Page Orientation — HORIZONTAL.
   f. Initial Aspect Ratio — FULL PAGE.
   g. Initial Axis Orientation — Select leftmost column.
   h. Line Quality — PREVIEW.
   i. Pen Parameter Access — PEN 1.
   j. Pen Type — FIBER TIP.

NOTE

To set the Pen Type for Pen 2, select PEN 2 on the Pen Parameter Access line, and then select FIBER TIP on the Pen Type line. The next two lines, Pen Pressure and Pen Velocity Limit, are set automatically.

   k. Alpha Character Quality — NORMAL.
   l. Interface Select — 2. (Select 1 if Option 4 is installed.) (If the Interface Select line selection is changed, it will be necessary to momentarily push the Parameter Entry Card fully in before resuming.)

m. Initial Command/Response Format — 1. (Non-Tektronix controllers may require a different selection. Refer to the Parameter Entry Card description in Section 5 for details.)

n. GPIB Device Address — 1.

o. Interface Mode — NORMAL.

p. Interface Functions — CR GENERATES LF. (Press either switch.)

NOTE

The remaining lines on the Parameter Entry Card may be ignored.

q. Push the Parameter Entry Card back into the Plotter. No other selections are required at this time.

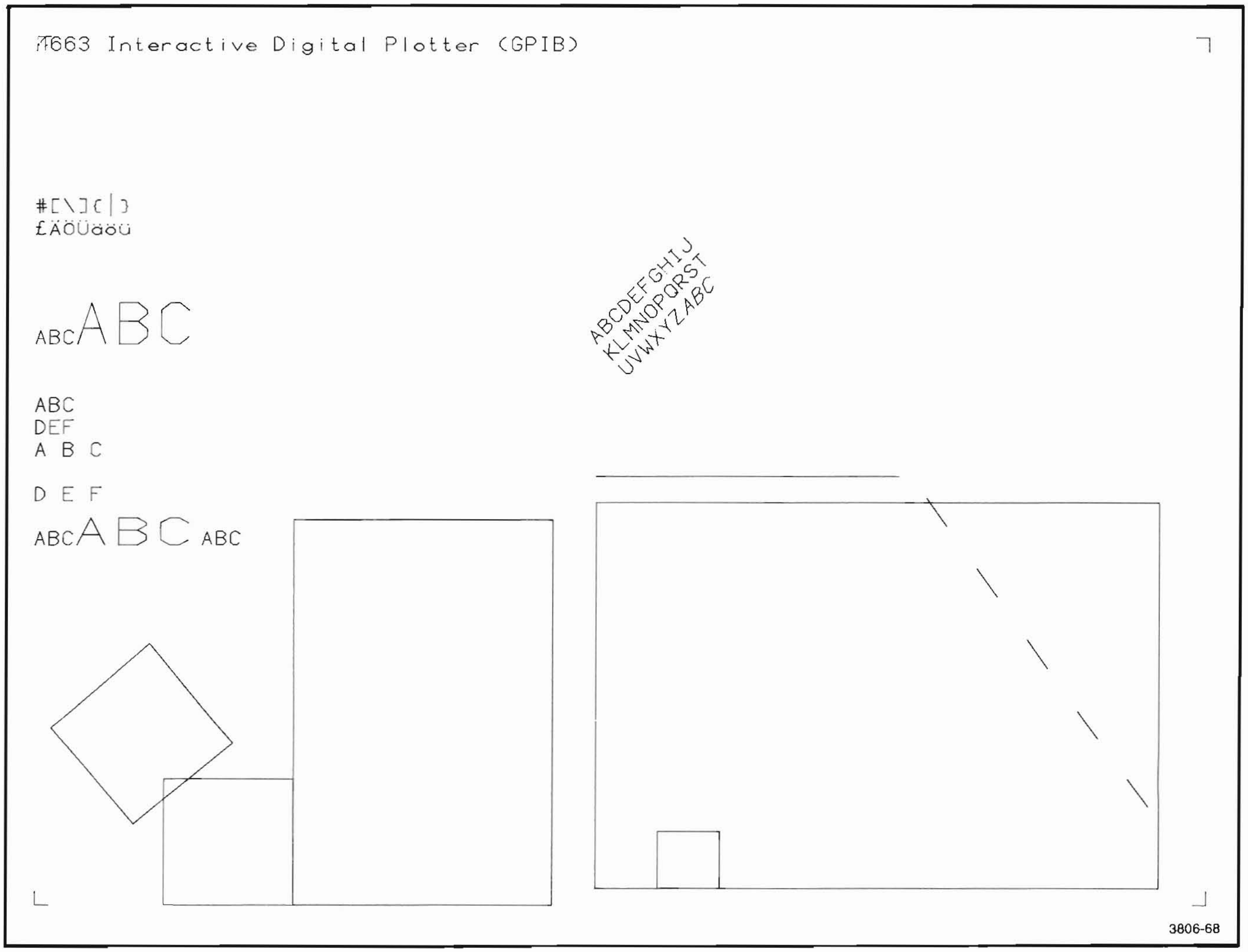


Figure 3-1. Plot Produced by Section 3 Procedures.

4. Execute the following samples of Plotter commands.

NOTE

In this manual, the ASCII Space character is always represented by ^{S_P}. The use of spacing within command descriptions is for clarity of presentation only. Do not send a Space character in place of spacing. The following example contains two Space characters:

PRINT@1,32: "P Three^{S_P}word^{S_P}example." ^{C_R}

   a. Move the pen carriage near the upper-left corner of the Page:

      PRINT@1,32: "AH" ^{C_R}

   b. Print: 4663 Interactive Digital Plotter

      PRINT@1,32: "P 4663^{S_P}Interactive^{S_P} Digital^{S_P} Plotter^{S_P}(GPIB)" ^{C_R}

      The alphanumeric text within the quotes (except for the first P) will be printed, followed by a carriage return and line feed.

NOTE

The P in the PRINT command is the Plotter command code for "print" and will not be printed.

   c. Transmit the following MOVE command to move the pen carriage to a point near the center of the Page:

      PRINT@1,32: "M"; 65;50 ^{C_R}

   d. To draw a horizontal vector from this point transmit a DRAW command:

      PRINT@1,32: "D"; 100;50 ^{C_R}

   e. To move two character spaces to the right and down one line space, type a CHARACTER MOVE command:

      PRINT@1,32: "AM"; 2;-1 ^{C_R}

5. Draw dashed line vectors.

   a. Transmit a SET DASH PATTERN command:

      PRINT@1,32: "BD"; 2;3 ^{C_R}

      This determines the proportion of blank line to drawn line.

   b. Establish a 10 GDU long dash pattern length using the SET DASH PATTERN LENGTH command:

      PRINT@1,32: "BS"; 10 ^{C_R}

   c. Transmit a SELECT LINE TYPE command to establish fixed dash- pattern-length line type:

      PRINT@1,32: "BL"; 1 ^{C_R}

   d. Issue a DRAW command to cause the Plotter to draw a dashed line from the current pen carriage position to the point determined by the command coordinates:

      PRINT@1,32: "D"; 130;10 ^{C_R}

   e. To reset the Plotter solid line type, transmit a SELECT LINE TYPE command:

      PRINT@1,32: "BL"; 0 ^{C_R}

6. Change the font selection.

   a. Transmit a MOVE command to move the pen carriage to the left edge of the Page and down from the top:

      PRINT@1,32: "M"; 0;80 ^{C_R}

   b. Transmit the following PRINT command; notice that the printed characters appear like those of the first line in Figure 3-2.

      PRINT@1,32: "P #[\ ] { |}" ^{C_R}

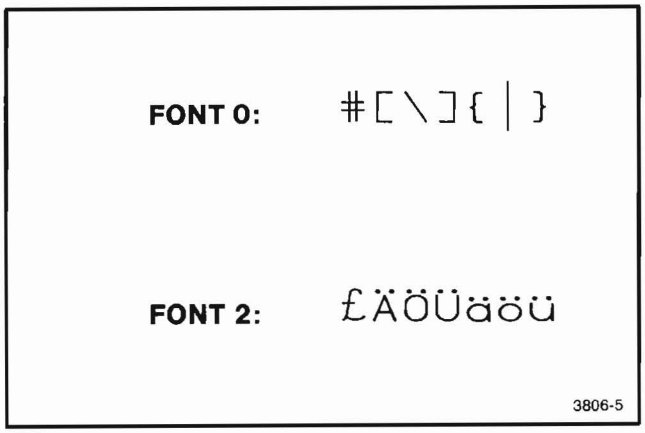


Figure 3-2. Fonts 0 and 2.

c. To select Font 2, transmit a SELECT STANDARD ALPHA FONT command:

PRINT@1,32: "F"; 2 CR

d. Transmit the following PRINT command; notice that the printed characters appear like those of the second line in Figure 3-1.

PRINT@1,32: "P #[\ ] { | }" CR

e. Reset the alpha parameters:

PRINT@1,32: "A" CR

7. Change the printed character size.

a. Transmit a MOVE command to move the pen to the left edge and down from the top:

PRINT@1,32: "M"; 0;65 CR

b. Print ABC:

PRINT@1,32: "P ABC" ; CR

c. Transmit an ALPHA SIZE command:

PRINT@1,32: "BZ"; 3.24;4.66 CR

d. Print ABC:

PRINT@1,32: "P ABC" CR

The printed characters will be approximately three times as large as they were for step 7b.

e. Set the alpha parameters back to default:

PRINT@1,32: "A" CR

8. Change the alpha ratio.

a. Transmit the following commands:

PRINT@1,32: "P ABC" CR

PRINT@1,32: "P DEF" CR

b. Change the alpha ratio:

PRINT@1,32: "BI"; 3;3.27 CR

c. Repeat Step 8a.

The Plotter prints characters of the same size, but with a distance from the origin of one character to the origin of the next character which is twice the default distance; the distance from the baseline of one line of printing to the baseline of the next line of printing is approximately twice the default distance.

d. Reset the alpha parameters to their default values:

PRINT@1,32: "A" CR

9. Change the alpha scaling factors.

a. To alter the X and Y multiplication factors for alpha characters, transmit a SET ALPHA SCALE command:

PRINT@1,32: "BH"; 3;2 CR

b. Transmit the following command:

PRINT@1,32: "P ABC" ; CR

Notice that the characters and character-to-character spacings are three times as wide and three times as high as they were before the SET ALPHA SCALE command in step 9b.

c. To reset the scaling factors to the default condition, transmit a SET ALPHA SCALE command with magnification factors 1,1:

PRINT@1,32: "BH"; 1;1 CR

d. Transmit the following Plotter PRINT command to verify that the characters are printed with default size and spacing:

PRINT@1,32: "P ABC" CR

10. Print characters on a diagonal line.

a. Move the pen carriage near the center of the media:

PRINT@1,32: "M"; 65;65 CR

b. To set the printing angle (and to set the left margin), transmit a SET ALPHA ROTATION command:

PRINT@1,32: "R"; 45 CR

c. Transmit the following 4663 PRINT command:

PRINT@1,32: "P ABC" ; CR

The characters are printed on a line rotated 45° from horizontal.

d. A right margin is established by transmitting a SET ALPHA MARGIN SEPARATION command:

PRINT@1,32: "BR"; 10 CR

e. Print the rest of the alphabet and notice that after ten characters have been typed on a line, the Plotter automatically executes a carriage return and a line feed and starts printing on the next line. Also, notice that both margins (left and right) are perpendicular to the printing direction (which in this case is at a 45° angle).

PRINT@1,32: "P DEFGHIJKLMNOPQRS-TUVWXYZ"; CR

11. Change the slant of the printed characters.

   a. Slanted characters can be printed by transmitting a SET ALPHA SLANT command:

      PRINT@1,32: "BG"; 20 CR

   b. Print three alphabetic characters:

      PRINT@1,32: "P ABC" CR

      Notice that these characters, although printed along the 45° angle printing axis are, in addition, slanted to the right 20°.

   c. Reset the alpha parameters:

      PRINT@1,32: "A" CR

12 Sample transform commands.

   To illustrate some of the transform commands, first draw a square box and then modify it with some of the transform commands.

   a. Transmit a MOVE command to position the pen carriage near the lower-left corner:

      PRINT@1,32: "M"; 15;0 CR

   b. To draw the box, transmit the following DRAW commands:

      PRINT@1,32: "D"; 30;0 CR
      PRINT@1,32: "D"; 30;15 CR
      PRINT@1,32: "D"; 15;15 CR
      PRINT@1,32: "D"; 15;0 CR

      A box with 15 GDU sides will be drawn with its lower-left corner 15 GDUs to the right of the Viewport origin.

   c. To save the present (default) transform, transmit a SAVE CURRENT TRANSFORM command:

      PRINT@1,32: "AX" CR

   d. To change the X-Y scaling factors for subsequent graphing, transmit a SET SCALE command:

      PRINT@1,32: "AS"; 2;3 CR

   e. Retransmit the commands in steps 12a and 12b. Notice that this box is twice as wide and three times as high as the first box and is displaced to the immediate right of the first box. The lower-left corner of both boxes is 15,0 but 15 on the X-axis, after rescaling, begins twice as far from the origin as when drawing the first box.

   f. To restore the default transform matrix and delete the scaled transform matrix, transmit a RESTORE PREVIOUS TRANSFORM command:

      PRINT@1,32: "AY" CR

13. Rotate the Viewport.

   a. Transmit the SAVE CURRENT TRANSFORM command to save the present (default) transform matrix:

      PRINT@1,32: "AX" CR

   b. To rotate the axes of the Viewport 40°, transmit a SET ROTATION command:

      PRINT@1,32: "AR"; 40 CR

   c. Then retransmit the commands in Steps 12a and 12b Notice that the box is redrawn and rotated counterclockwise by 40°.

   d. To restore the default transform matrix and delete the current transform matrix, transmit a RESTORE PREVIOUS TRANSFORM command:

      PRINT@1,32: "AY" CR

14. Establish a new Viewport.

   a. To mark the Initial Viewport, transmit a MARK VIEWPORT command:

      PRINT@1,32: "CM" CR

   b. Transmit a SET VIEWPORT command:

      PRINT@1,32: "AV"; 65;130,2;47 CR

   c. To outline this new Viewport, transmit an OUTLINE VIEWPORT command:

      PRINT@1,32: "CB" CR

   d. Redraw the box using the commands in steps 12a and 12b. Notice that the box and the Viewport outline in step 12c are reduced in size proportionately.

   e. To restore the Initial Viewport transform matrix, pull the Parameter Entry Card out to the Initial Page Size line and press the switch over "C".

NOTE

Even though that switch's light will still be on, you must press the switch at this time. The Viewport size then becomes the same as the Initial Page size.

15. Disable one axis of the Joystick.

    This command could be used, for example, when digitizing the Y-coordinates of a graph at a fixed X-axis location.

    a. To disable the X-axis of the Joystick, transmit a JOYSTICK DISABLE command and specify the X-axis:

       ```
       PRINT@1,32: "BJ"; 1 CR
       ```

    b. Move the Joystick around and notice that the pen only moves along the Y-axis.

    c. Both axes of the Joystick are again enabled anytime the host repositions the pen or another JOYSTICK DISABLE command is given, specifying that neither axis be disabled. Issue a JOYSTICK DISABLE command to enable the Joystick:

       ```
       PRINT@1,32: "BJ"; 0 CR
       ```

16. Obtain digitized pen carriage coordinates.

    a. Position the pen near the center of the Viewport:

       ```
       PRINT@1,32: "M"; 65;50 CR
       ```

       Remember, the Plotter is now operating with the Initial Viewport and the small Viewport created earlier has been deleted.

    b. To display the pen carriage coordinates (in GDUs) on the display screen, transmit the following sequence:

       ```
       PRINT@1,32: "G" CR
       INPUT@1,32: X,Y,T CR
       PRINT X,Y,T CR
       ```

       Notice that the first two numbers indicate the coordinates, respectively, for the MOVE command in step 16a.

    c. Use the Joystick to move the pen carriage. Repeat the previous command sequence in step 16b. The new pen coordinates will be be printed on the controller's display.

This completes the familiarization procedure for the GPIB interface commands.

Section 4

SERIAL INTERFACE

INTRODUCTION

This section describes two aspects of the Serial interface. These are:

- How to install the Plotter into a system while considering (1) connecting the Plotter to other serial devices, (2) making proper selections on the Paramenter Entry Card to permit device communication with the Plotter, and (3) determining the best method to control information flow between devices and the Plotter.
- How to program the Plotter by (1) understanding the various ways in which Plotter commands are formatted, and (2) seeing how any resulting Plotter responses are decoded.

CONNECTING THE PLOTTER TO OTHER DEVICES

The flexibility provided by the front panel Interface Switching Function and the Parameter Entry Card allows the Plotter to be connected with other devices in a variety of ways. The following describes four basic ways to interconnect the Plotter with a host computer, a terminal, or a storage device:

- The Plotter is connected as a peripheral to the host.
- The Plotter is connected in a *loop-thru configuration* to a terminal (where the host talks through the Plotter to the terminal).
- The Plotter is connected as a peripheral to a terminal.
- The Plotter operates in conjunction with an offline storage device (for example, a host does not exist in the system).

CONNECTED AS A HOST PERIPHERAL

In this configuration, the Plotter is viewed as an independent peripheral available to the host. A host peripheral configuration is most appropriate when the Plotter is a resource available to all host-system users.

The Plotter can be located near the host or located at a remote location (possibly with a terminal). Figure 4-1A illustrates the simplest version of this configuration where the Plotter is connected to the host through a dedicated input/output (I/O) port. When the Plotter is located near the host, modems are not required. Figure 4-1B shows an equivalent configuration where a single communication line to either the remote Plotter or terminal is shared using a multiplexer/modem combination. The software support for this configuration is the same as shown in Figure 4-1A.

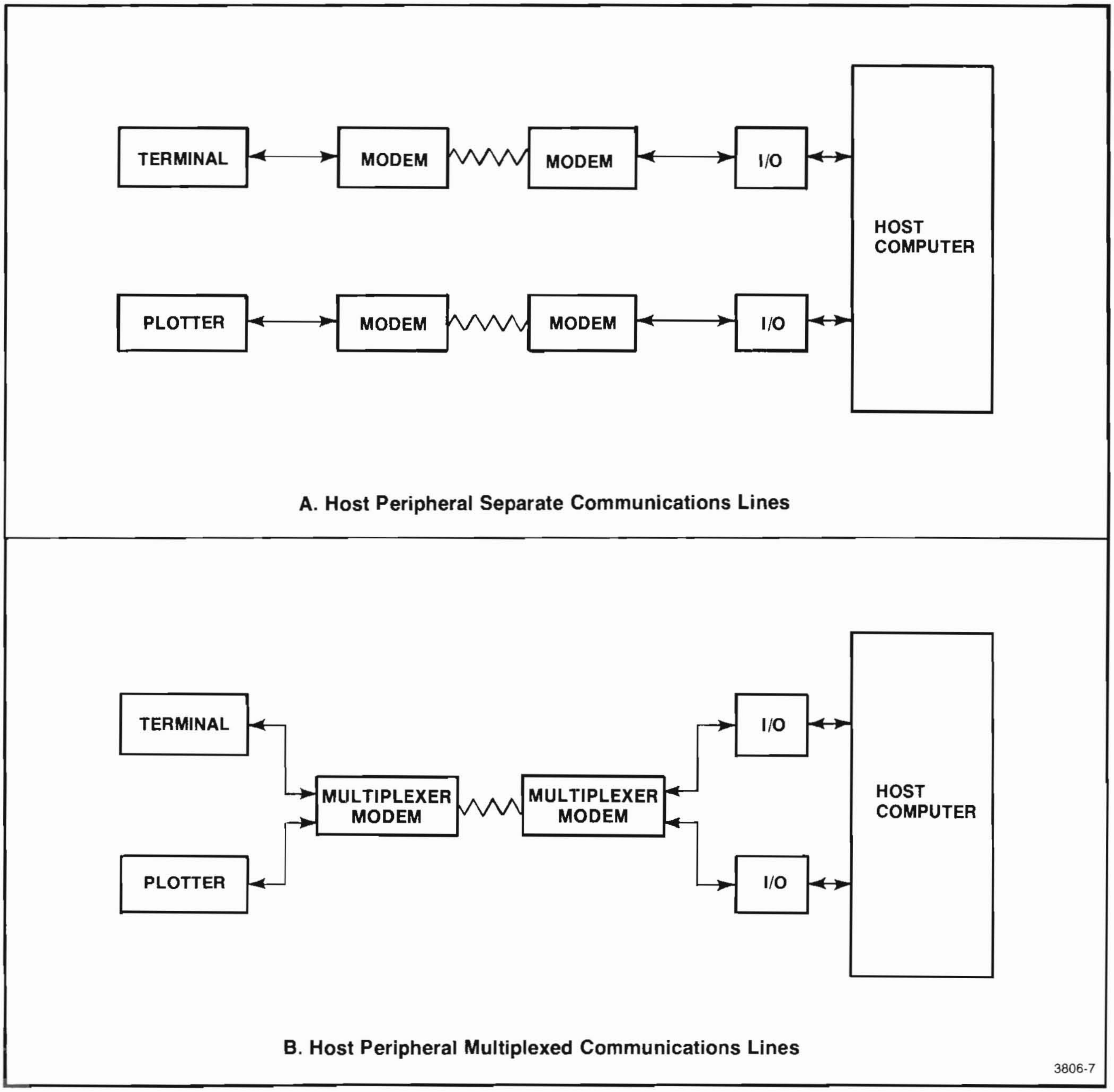


Figure 4-1. Host Peripheral Configurations.

CONNECTED IN A LOOP-THRU CONFIGURATION

For the configuration illustrated in Figure 4-2, the Plotter and terminal are connected in a loop-thru fashion in order to share a common communication line to the host. This configuration is appropriate when graphics to be drawn on the Plotter are generated by the host while under interactive control of the terminal.

Software support for this configuration is slightly more complex as the host must direct its communication through the single I/O port either to the terminal or to the Plotter. This is easily done with the Plotter's DEVICE ON and DEVICE OFF commands. When the host sends a DEVICE ON command to the Plotter (enabling the Plotter to recognize later commands), the Plotter interrupts command flow to the terminal.

A subsequent DEVICE OFF command disables the Plotter and allows the host to again access the terminal through the Plotter. Also, in this configuration, the Plotter can be used as a terminal peripheral if directed by the 4663 front panel Interface Switching function to communicate only with the terminal. The loop-thru configuration allows operation of the Plotter with virtually any type of terminal having a Serial interface.

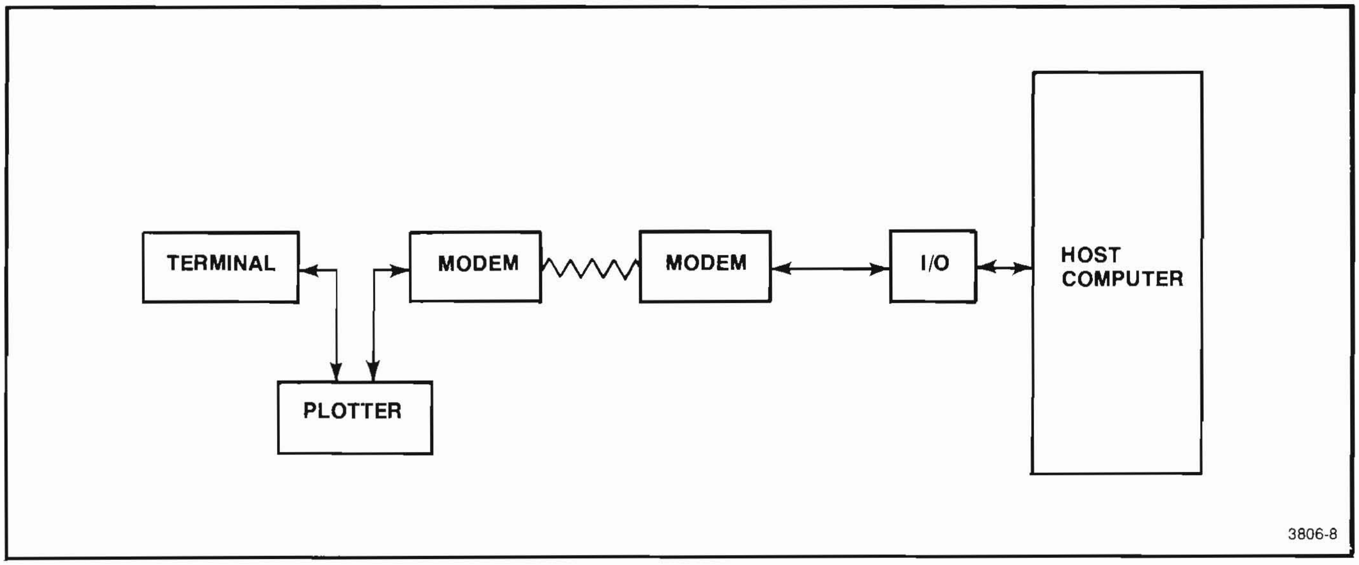


Figure 4-2. Host Peripheral Communications Line With Loop-Thru.

CONNECTED AS A TERMINAL PERIPHERAL

The configuration shown in Figure 4-3 gives the terminal priority over the host computer to use the Plotter as a peripheral. The terminal peripheral configuration is most appropriate when graphics to be plotted are generated either by the host or by the terminal. The terminal communicates with the Plotter directly, while the host must address the terminal to reach the Plotter.

Limiting host communication to only the terminal may simplify host support programs that deal with the terminal through the I/O port. This occurs especially when Plotter features used are the same as the available terminal features. If other unique Plotter features are used, more host software may be required to access the unique features through the terminal.

Other advantages of this configuration depend upon the terminal's capability. Refer to the specific Operator's Manual for the terminal. (Some terminals may be able to do background plotting from local data storage while continuously communicating with the host.)

Possible configurations with TEKTRONIX terminals are listed in Table 4-1 which notes any terminal interface options that must be present.

Table 4-1

TERMINAL INTERFACE OPTIONS REQUIRED WHEN CONFIGURING THE PLOTTER AS A TERMINAL PERIPHERAL

Tektronix Terminals	Interface Description	Option
4010, 4012, 4014, 4015	With Dual Interface Capability	Option 36
4016	With Dual Interface Capability	Option 35
4112, 4114	With 3-Port Peripheral Interface	Option 10
4025, 4027	With Peripheral Interface	Option 04[a]
4014, 4015	With Local Processor	Option 05[a]

[a] For these configurations, terminal communication with the the host is by a Serial Interface, while terminal communication with the Plotter is by a GPIB Interface. Refer to Section 5 of this manual for GPIB information.

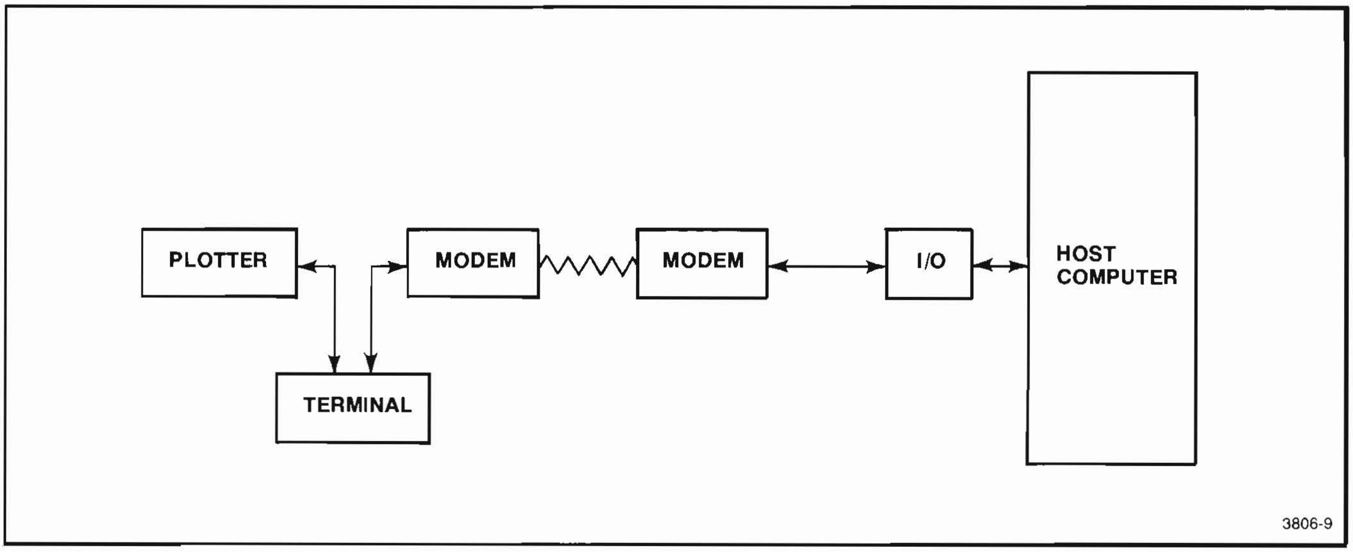


Figure 4-3. Terminal Peripheral Configuration.

OFFLINE PLOTTING

If data to be plotted is recorded on a storage media (such as magnetic tape or floppy disk), the Plotter can be configured as shown in Figure 4-4. This configuration permits data to be plotted without needing a terminal or a host computer as a controller. An example of a data storage device with a Serial interface is the TEKTRONIX 4923 Digital Tape Drive Unit with Option 01.

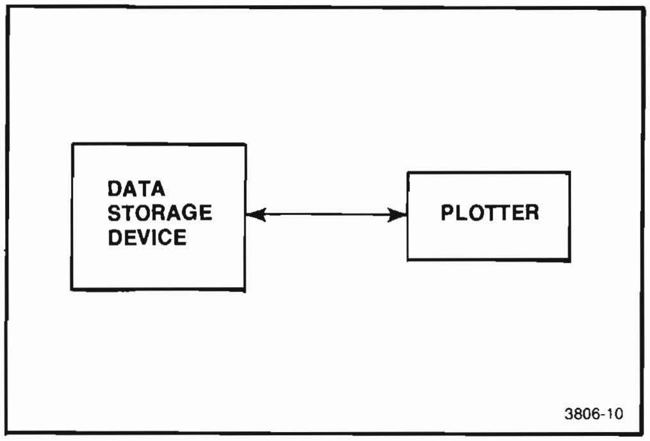


Figure 4-4. OFFLINE Plotting Configuration.

Two combination systems are shown in Figure 4-5. These systems use a TEKTRONIX 4010 Series terminal with either a standard 4923 Digital Tape Drive Unit or a 4923 Digital Tape Drive Unit with Option 01. These configurations allow plot data to be recorded while online with the host computer and then later plotted during offline time. The REMOTE and LOCAL switching functions on both the tape unit and the Plotter direct the routing of the plot data.

MULTIPLE PLOTTERS

Multiple Plotters can be installed in a loop-thru configuration in place of the single Plotter described in any of the above configurations. Each of the Plotters can be set to the same device address and receive identical data in parallel or each can use a different device address and receive data independently. Refer to *Communicating With Other Devices*, further in this section for details about multiple plotter configurations.

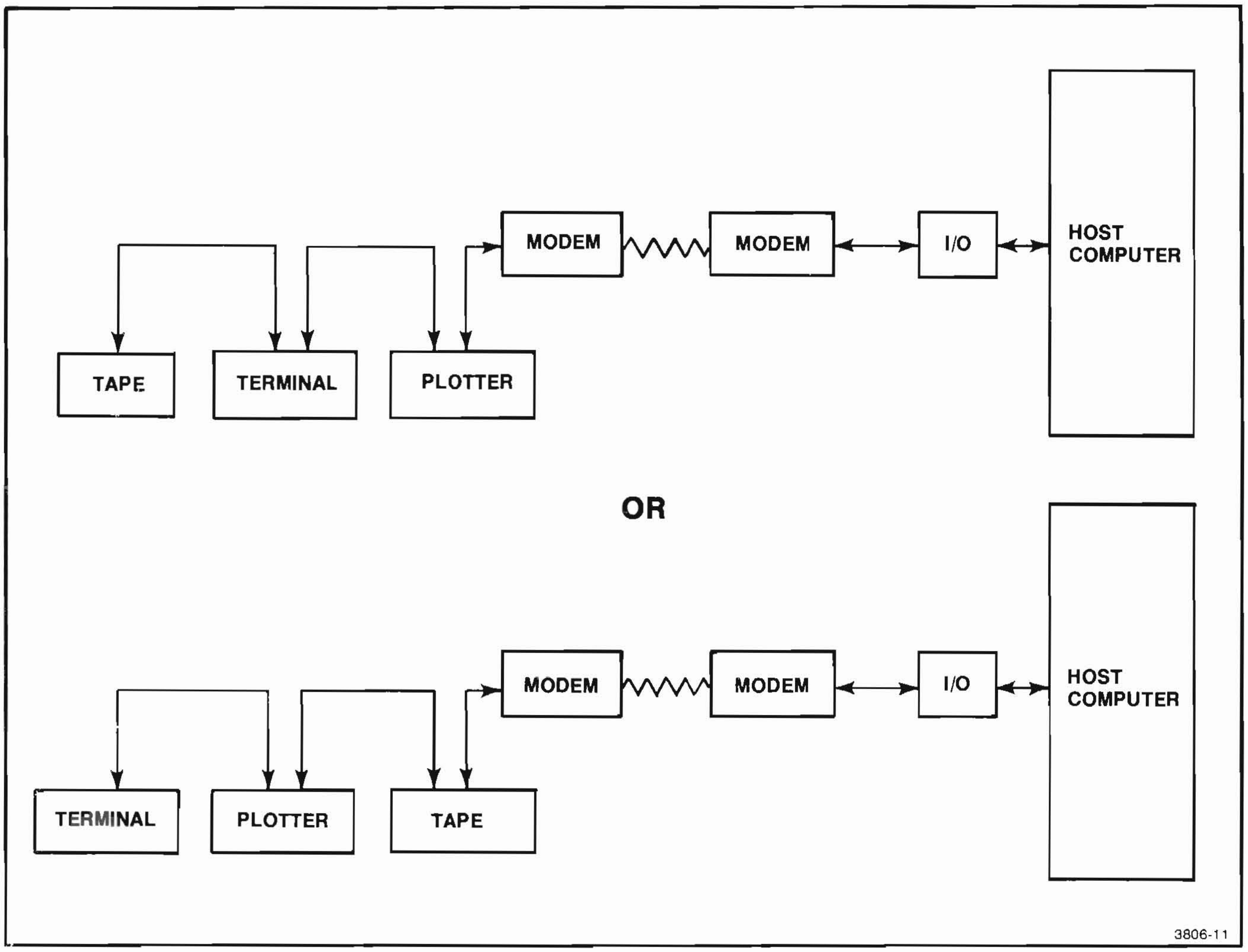


Figure 4-5. ONLINE Recording and OFFLINE Plotting.

SETTING UP THE PARAMETER ENTRY CARD

Selections made on the Parameter Entry Card lines permit the Plotter to operate in a variety of communications environments. Described here are the possible selections and information needed to choose selections for a particular communication environment.

The majority of the Paramenter Entry Card lines described here are from the Serial Interface portion of the Paramenter Entry Card. Refer to Figure 4-6. Two other lines, Interface Select and Initial Command/Response Format, are also included since they must be set specifically for each interface type installed in the Plotter.

Refer to the description of the Parameter Entry Card in the *4663 Operator's Manual* for details about making any of the selections described in this section.

INTERFACE SELECT LINE

The selection made on the Interface Select line determines which hardware interface is active. As shipped from the factory (with the standard RS-232-C Interface for Serial or with the Option 01 interface for Serial and GPIB), selecting "1" on the Interface Select line activates the Serial interface. Selecting "2" activates the GPIB interface. If the Plotter is shipped with Option 04 (GPIB Only Option), the "1" selection activates the GPIB interface and the "2" selection is inactive. Table 4-2 summarizes the possible selections.

NOTE

The interface activated by selection of "1" or "2" on the Interface Select Line could be changed by resetting straps on the Interface circuit board. Refer to the 4663 Service Manual for additional information.

Table 4-2

INTERFACE DESIGNATIONS

Interface(s) Installed	**Option Number**	**Interface Select Line**	
		1	**2**
Serial Only	Standard	SERIAL	Not used
Serial and GPIB	Option 01	SERIAL	GPIB
GPIB Only	Option 04	GPIB	Not used

INITIAL COMMAND/RESPONSE FORMAT LINE

The selection made on the Initial Command/Response Format line configures the Serial interface to accept commands and send responses in a specified format. Refer to *Introduction to Serial Commands and Responses*, in this section for more information about the various command and response formats.

If the Plotter is being installed to operate with existing host software, then the Initial Command/Response Format line selection must agree with the host software formatting requirements. If new host software is being developed to drive the Plotter, then the choice of a format line can be based upon the tradeoff between communications performance and host software complexity. The following paragraphs describe each of the Initial Command/Response Format line selections and indicate how an appropriate choice can be made.

NOTE

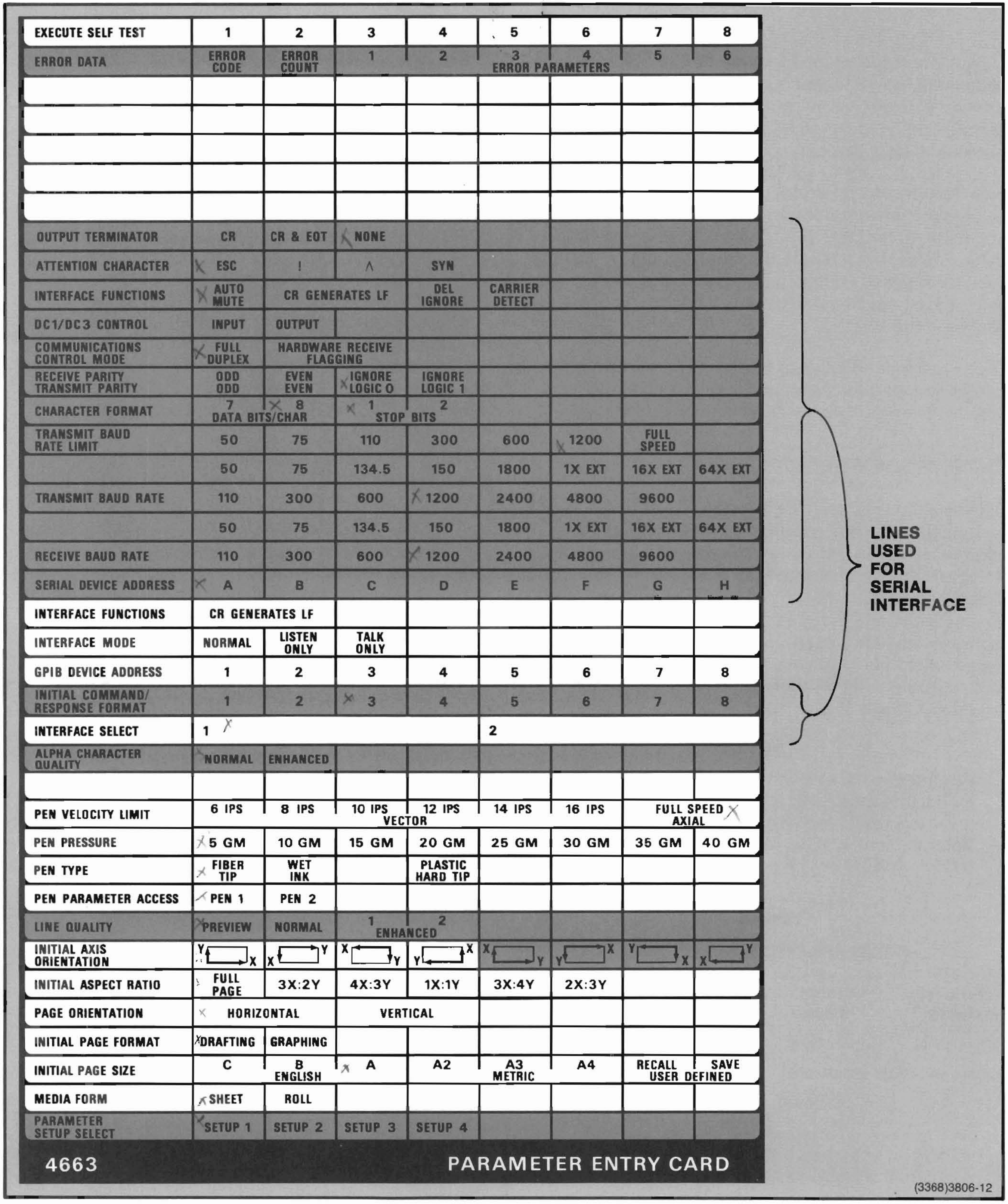
When certain events occur, selection of the Initial Command/Response Format configures the interface to accept the specified format. These events are: (1) at power-up, (2) when a new selection is made on the Initial/Command Response Format line, (3) when the Serial interface is first selected on the Interface Select line, and (4) when an INTERFACE PARAMETER RESET command is received.

The SELECT COMMAND/RESPONSE FORMAT command overrides the Initial Command/Response Format line selection but does not modify it. This allows the configuration defined by the Parameter Entry Card selection to be restored when one of the above listed conditions occurs.

NOTE

The following describes only the Initial Command/Response Formats as they apply to a Serial interface. The function of each format changes when the GPIB interface is selected. Refer to Section 5 for information about the GPIB Command/Response Formats.

SETTING UP PARAMETER ENTRY CARD



EXECUTE SELF TEST	1	2	3	4	5	6	7	8
ERROR DATA	ERROR CODE	ERROR COUNT	1 ERROR PARAMETERS	2	3	4	5	6
OUTPUT TERMINATOR	CR	CR & EOT	[✓ NONE]					
ATTENTION CHARACTER	[✓ ESC]	!	^	SYN				
INTERFACE FUNCTIONS	[✓ AUTO MUTE]	CR GENERATES LF		DEL IGNORE	CARRIER DETECT			
DC1/DC3 CONTROL	INPUT	OUTPUT						
COMMUNICATIONS CONTROL MODE	[✓ FULL DUPLEX]	HARDWARE RECEIVE FLAGGING						
RECEIVE PARITY TRANSMIT PARITY	ODD ODD	EVEN EVEN	[✓ IGNORE LOGIC 0]	IGNORE LOGIC 1				
CHARACTER FORMAT	7 DATA BITS/CHAR	[✓ 8]	[✓ 1] STOP BITS	2				
TRANSMIT BAUD RATE LIMIT	50	75	110	300	600	[✓ 1200]	FULL SPEED	
	50	75	134.5	150	1800	1X EXT	16X EXT	64X EXT
TRANSMIT BAUD RATE	110	300	600	[✓ 1200]	2400	4800	9600	
	50	75	134.5	150	1800	1X EXT	16X EXT	64X EXT
RECEIVE BAUD RATE	110	300	600	[✓ 1200]	2400	4800	9600	
SERIAL DEVICE ADDRESS	[✓ A]	B	C	D	E	F	G	H
INTERFACE FUNCTIONS	CR GENERATES LF							
INTERFACE MODE	NORMAL	LISTEN ONLY	TALK ONLY					
GPIB DEVICE ADDRESS	1	2	3	4	5	6	7	8
INITIAL COMMAND/ RESPONSE FORMAT	1	2	[✓ 3]	4	5	6	7	8
INTERFACE SELECT	[✓ 1]				2			
ALPHA CHARACTER QUALITY	[✓ NORMAL]	ENHANCED						
PEN VELOCITY LIMIT	6 IPS VECTOR	8 IPS	10 IPS	12 IPS	14 IPS	16 IPS	[✓ FULL SPEED AXIAL]	
PEN PRESSURE	[✓ 5 GM]	10 GM	15 GM	20 GM	25 GM	30 GM	35 GM	40 GM
PEN TYPE	[✓ FIBER TIP]	WET INK		PLASTIC HARD TIP				
PEN PARAMETER ACCESS	[✓ PEN 1]	PEN 2						
LINE QUALITY	[✓ PREVIEW]	NORMAL	1 ENHANCED	2				
INITIAL AXIS ORIENTATION	Y X	X Y	X Y	Y X	X Y	Y X	Y X	X Y
INITIAL ASPECT RATIO	[✓ FULL PAGE]	3X:2Y	4X:3Y	1X:1Y	3X:4Y	2X:3Y		
PAGE ORIENTATION	[✓ HORIZONTAL]		VERTICAL					
INITIAL PAGE FORMAT	[✓ DRAFTING]	GRAPHING						
INITIAL PAGE SIZE	C ENGLISH	B	[✓ A]	A2 METRIC	A3	A4	RECALL USER DEFINED	SAVE
MEDIA FORM	[✓ SHEET]	ROLL						
PARAMETER SETUP SELECT	[✓ SETUP 1]	SETUP 2	SETUP 3	SETUP 4				
4663					PARAMETER ENTRY CARD			

Figure 4-6. Parameter Entry Card Lines for the Serial Interface.

Command/Response Format 1

Format 1 features are:

- 4663 Command Set
- Packed Binary Response Encoding

Format 1 is the standard 4663 Command Response Format used with TEKTRONIX PLOT 10 Graphic Library 4663 support software. All 4663 commands can be used in Format 1. This format has the widest range of command and argument forms available.

For maximum transmission efficiency, any responses transmitted by the Plotter are encoded in a packed binary format that must be decoded by the host. This involves packing the binary bits of the output values into seven ASCII characters for transmission rather than converting the values to ASCII decimal numbers (which could take up to 33 characters). The packed-binary format limits the numerical range of response values (refer to *Output Responses* near the end of this section for more information).

Command/Response Format 2

Format 2 features are:

- 4663 Command Set
- ASCII Decimal Response Encoding

Format 2 is an optional 4663 Command/Response Format. All 4663 commands can be used as in Format 1, but the output responses are encoded in ASCII decimal format. This format is less efficient during command transmission, but does not require special decoding by most host computers and can encode any required numerical values.

Command/Response Format 3

Format 3 features are:

- 4662 Compatible Command Set
- 4662 Packed Binary Response Encoding

Format 3 is a 4662 compatible Command/Response Format used with TEKTRONIX PLOT 10 Graphic Library 4662 support software. This is the format to use when a 4663 Plotter is installed in a system that currently supports a TEKTRONIX 4662 Plotter. Refer to the *4662 Interactive Plotter Programmer's Reference Manual* for more information.

In this format, the 4663 accepts all commands written for the 4662 Plotter and any output responses are encoded in the 4662 packed-binary format. The packed-binary format is similar to the 4663 packed-binary format with the exception of TAG value assignments. Any 4663 Plotter commands generating an output response not defined for the 4662 Plotter (such as the READ VIEWPORT command) initiates an error when used with Format 3.

All other 4663 commands not generating output are accepted (permitting 4662 oriented systems to be upgraded). In Format 3, the ᶜʀ and semicolon characters are not recognized as special command delimiters.

NOTE

Format 1 can also be used in a 4662-oriented system as long as the only output responses requested from the 4663 are Block mode acknowledge responses.

Command/Response Format 4

Format 4 features are:

- 4662 Compatible Command Set
- Restricted 4662 Packed Binary Response Encoding

This format is identical to Format 3 with the exception that packed-binary coded output is restricted so it will not contain the ASCII character "@" (since some hosts use this character for special functions). In Format 4, any digitized output, generated from either a DIGITIZE command or from a front panel Point switch operation, always has the least significant bit of the X-coordinate set to a "1" (this avoids the "@" character). For 4662 compatible operation, the resolution of digitize responses, use only twelve bits. Thus, the four least significant bits of the coordinates, which normally are 0, should be ignored by the host.

SERIAL DEVICE ADDRESS LINE

The selection made on the Serial Device Address line configures the Serial interface to accept only Style I Plotter commands that contain the selected device address character (uppercase A through H). Any Style I Plotter command not containing this address is ignored. (Refer to *Introduction to Serial Commands and Responses* later in this section for a discussion of Style I commands.)

RECEIVE/TRANSMIT BAUD RATE LINES

The selections made on the Receive Baud Rate and Transmit Baud Rate lines should correspond to the baud rate of the modem, host I/O port, or other interconnected devices.

A wide range of baud rates are available to accomodate most communications devices, but if the baud rate selected exceeds 1200 baud, then some restrictions on communication methods may apply. Refer to *Communicating With the Host Computer* later in this section for a discussion of these restrictions, and how to select an optimum baud rate when the available communication devices permit a choice of baud rates.

The Receive Baud Rate and Transmit Baud Rate line selections are made independently to permit Plotter reception at one rate and transmission at a different rate.

If a modem is used that provides transmit and/or receive clocks (such as a synchronous modem used at higher baud rates), then the external clock selections (1X EXT, 16X EXT, and 64X EXT) must be set to match the modem requirements.

These selections choose an external clock to be supplied by the RS-232 connector and specify a clock divider ratio. If a 1X EXT or 64X EXT selection is required on a single baud-rate line, the same selection must be made on the other line. External clock selections made on the Transmit Baud Rate line or on the Receive Baud Rate line automatically sets both lines to the same rate. If a 16X EXT selection is required on a single baud-rate line, then one of the non-external baud rate selections may optionally be made on the other line. When an external clock is used, the baud rate is determined entirely by the external clock frequency and the clock dividing ratio. The maximum clock frequency to be used is 307 KHz. This gives the maximum baud rate obtainable when using each of the clock divider ratios as shown below:

X1	EXT	307 KBaud
X16	EXT	19.1875 KBaud
X64	EXT	4.7968 KBaud

Although interface hardware can receive characters up to rates of 307 KBaud, the Plotter cannot process sequential characters at a rate greater than 9600 Baud. Therefore, a delay must be added between successive characters when communicating at baud rates greater than 9600 baud to reduce the effective baud rate to 9600 baud or less.

NOTE

If the interface requires an X1 external clock signal, the signal must have a rising edge near the center of the bit cell.

TRANSMIT BAUD RATE LIMIT LINE

The Transmit Baud Rate Limit line can be used to accomodate hosts that are unable to accept a continuous string of characters at the current transmit baud rate. A selection on this line configures the interface to effectively reduce the current transmit baud rate to the specified value by pausing between transmitted characters (where each character is sent at the current transmit baud rate).

The required Transmit Baud Rate Limit needed to communicate with a given host varies widely depending upon the communications capabilities of the host and its current processing load. This rate may require a very conservative selection (such as a low limit value) to allow communication during all conditions. A typical limit would be approximately 300 to 600 baud (30 to 60 characters per second), which approximates the maximum input capability of a terminal operator.

For hosts without a low rate restriction, select FULL SPEED to permit response transmission at the specified transmit baud rate.

NOTE

The Transmit Baud Rate Limit line must be set to FULL SPEED if selections are made on the DC1/DC3 Control line of the Parameter Entry Card.

CHARACTER FORMAT LINE

The selections made on the Character Format line configures the interface to accept and send characters in the desired format. The DATA BITS/CHARACTER selection should be "7" if the parity of the received characters is to be checked (the Plotter communicates using 7-bit ASCII characters plus a parity bit). If the received character parity is to be ignored, then 8-bit characters should be specified. The parity check is enabled by the selection made on the Receive Parity/Transmit Parity line.

The STOP BITS selection specifies that each transmitted response character is to be formatted with 1 or 2 stop bits. Normally, "2" is selected for greatest reliability when transmitting at baud rates greater than 1200 baud. However, if communications equipment requires only one stop bit, "1" may be selected to improve transmission efficiency. The Plotter can receive characters with 1 or 2 stop bits irrespective of the STOP BIT selection.

NOTE

The 8 data bits per character and the 2 stop bits is an illegal combination of parameters when parity checking is enabled. Any selection on this line resulting in this combination causes parity checking to be automatically disabled by forcing the selection on the Receive Parity/Transmit Parity line to be IGNORE/LOGIC 1.

RECEIVE PARITY/TRANSMIT PARITY LINE

The Receive Parity/Transmit Parity line permits selection of the desired parity checks for received characters and specifies the parity bit (most significant bit) for transmitted characters. The choices include the following:

- ODD/ODD — Checks received characters for odd parity and sends transmitted characters with odd parity.
- EVEN/EVEN — Checks received characters for even parity and sends transmitted characters with even parity.
- IGNORE/LOGIC 0 — Disables any receive parity check and sets transmitted parity bits to LOGIC 0.
- IGNORE/LOGIC 1 — Disables any receive parity check and sets transmitted parity bits to LOGIC 1.

An IGNORE/LOGIC 0 or IGNORE/LOGIC 1 selection may be used if the communications devices do not require parity checks of responses that are transmitted by the Plotter. These choices accomodate host computers that require the parity bit for transmitted characters be set to a 1 or a 0, even when parity checking is not performed.

NOTE

Parity checking is illegal if 8 data bits per character and 2 stop bits are the current Character Format parameters. Any selection on this line that enables parity checking and results in this combination causes the STOP BIT selection on the Character Format line to automatically become 1 stop bit.

COMMUNICATIONS CONTROL MODE LINE

The selection made on the Communications Control Mode line establishes the communications protocol. (Refer to *Communicating With Other Devices* later in this section for more information on communications, including input/output rate control and the hardware flagging function.)

Both selections (FULL DUPLEX and HARDWARE RECEIVE FLAGGING) support full duplex communication (for example, simultaneous receive and transmit operations). The FULL DUPLEX selection is used when no active input rate control is desired. FULL DUPLEX is the normal selection unless HARDWARE RECEIVE FLAGGING has been chosen as the input rate control method. The FULL DUPLEX selection causes the RS-232-C Data Terminal Ready control line to be asserted whenever the Plotter's power is on (and any terminal connected in the loop-thru configuration is also asserting Data Terminal Ready). The HARDWARE RECEIVE FLAGGING selection enables the interface to provide active input rate control by using Data Terminal Ready to control transmission between other connected devices and the Plotter.

Hardware transmit flagging is always in effect, and the RS-232-C signal line Clear To Send must be asserted before transmission of a response character can begin. HARDWARE RECEIVE FLAGGING provides a convenient rate control method when supported by the other interconnected devices.

NOTE

When HARDWARE RECEIVE FLAGGING is chosen, selections should not be made on the DC1/DC3 Control line.

DC1/DC3 CONTROL LINE

The DC1/DC3 Control line enables the interface to use software flagging to provide active input and/or output rate control between other connected devices and the Plotter. (Refer to *Communicating With Other Devices* later in this section for more information on communications, including input/output rate control and the DC1/DC3 flagging function.)

The INPUT selection permits the Plotter to regulate the effective transmission rate of a connected device, while the OUTPUT selection permits a connected device to control the Plotter's effective transmission rate. In these modes, an ASCII DC3 (stop read) control character (CTRL S) and an ASCII DC1 (start read) control character (CTRL Q) are used to stop or start data transmissions between the Plotter or other connected devices to prevent overloading the receiving device. The DC1/DC3 Control functions provide a convenient control method when supported by the other connected devices.

NOTE

To use this function, FULL DUPLEX should be selected on the Communications Control Mode line and the Transmit Baud Rate Limit line selection must be FULL SPEED.

INTERFACE FUNCTIONS LINE

Selections made on the Interface Functions line must match the Plotter interface to a particular communications environment. These selections are AUTO MUTE, CR GENERATES LF, DEL IGNORE, and CARRIER DETECT. The following describes each selection and indicates the communications environment in which each could be used.

Auto Mute

Use the AUTO MUTE selection when the Plotter is installed in a loop-thru configuration. This selection prevents commands sent from either the host or other connected devices to the Plotter from being displayed on the terminal's screen. When AUTO MUTE is not selected, all data from other connected devices received by the Plotter also passes through the Plotter's interface to the terminal.

NOTE

If the Plotter is logically off, and AUTO MUTE is selected, a portion of the next PLOTTER ON command will be sent to the terminal before the mute function is activated.

At a receive baud rate of 9600, characters following the DEVICE OFF command may be garbled by the terminal if characters received by the Plotter do not contain two stop bits. Alternatively, there either should be a delay of about one character-time following the DEVICE OFF command, or the DEVICE OFF command should be followed by an ASCII ᴰᴛ character with even parity.

Refer to *Communicating With Other Devices* further in this section for a discussion of the AUTO MUTE function.

CR Generates LF

The CR GENERATES LF selection forces the Plotter to automatically generate a line feed following each ASCII carriage return (ᶜʀ) character that the host sends to the Plotter for printing. This selection is used when communicating with host computers that do not provide an automatic line feed after each ᶜʀ character.

Delete Ignore

Use the DEL IGNORE selection when communicating with host computers that use the ASCII $^{\mathrm{D}}{}_{\mathrm{T}}$ (DEL) character for special system functions. This selection causes any $^{\mathrm{D}}{}_{\mathrm{T}}$ characters to be ignored rather than used as a portion of a Style II command. An alternative two-character sequence $^{\mathrm{A}}{}_{\mathrm{T}}$ **?** may be sent from the host computer in place of $^{\mathrm{D}}{}_{\mathrm{T}}$ characters, which appear in Style II commands. Refer to *Introduction to Serial Commands and Responses* later in this section for more information about Style II commands.

Carrier Detect

When CARRIER DETECT is selected, the RS-232-C Receive Line Signal Detector (DATA CARRIER DETECT) signal line is monitored. If the signal is removed even momentarily, the front panel REMOTE light starts blinking to indicate a possible loss of data. Pressing the REMOTE switch on the Plotter cancels this status indication. Refer to *Communicating With Other Devices* further in this section for more details.

NOTE

The Plotter will not receive data if the RS-232-C Receive Line Signal Detector signal is held in an unasserted state.

ATTENTION CHARACTER LINE

The selection made on this line specifies the attention character used in the definition of Style I and Style II commands. The ASCII $^{\mathrm{E}}{}_{\mathrm{C}}$ (ESC) character is the usual selection and is compatible with the TEKTRONIX 4010 Series terminals, which also use Style II commands. However, the remaining three selections ("!", "∧", or SYN) may be useful in some communications environments where, for example, the host computer is not able to transmit an $^{\mathrm{E}}{}_{\mathrm{C}}$ character. Refer to *Introduction to Serial Commands and Responses* later in this section for more information about Style I and Style II commands.

OUTPUT TERMINATOR LINE

The selection made on this line specifies the Plotter's response termination characters to be used when the Plotter transmits messages back to the host computer.

The three selections available are CR, CR & EOT, or NONE. A host computer with line-oriented input normally requires a carriage return $^{\mathrm{C}}{}_{\mathrm{R}}$ terminator or possibly a $^{\mathrm{C}}{}_{\mathrm{R}}$ followed by an $^{\mathrm{E}}{}_{\mathrm{T}}$ (EOT) character to indicate the end of an input line. A host with character-oriented input may not require a response terminator and any selection (including NONE) would be acceptable.

COMMUNICATING WITH OTHER DEVICES

The following presentation describes how command and response transmissions are sent between the Plotter and other connected devices. This information is described under the following headings:

- Selecting a Communications Control Mode
- Interface Switching Function
- RS-232-C Lines for the Loop-Thru Configuration
- Device-to-Plotter Transmission
- Plotter-to-Device Transmission
- Communicating With Multiple Plotters

Two activities must be considered when transmitting commands and responses between the Plotter and the host computer (or other devices). The first is the detection and possible correction of transmission errors. This activity is optional and is discussed later in the manual.

The second activity is control of the rate at which commands and responses are passed between the Plotter and the host computer. As the Plotter is an electro mechanical device, the rate at which it processes received plot commands is normally slower than the rate at which the host computer or other devices can send commands. Thus, the Plotter must interactively control the rate at which commands are received to avoid loss of information. This process is refered to as *input rate control.*

Likewise, some host computer configurations require slow data input and cannot receive Plotter responses as rapidly as the Plotter can generate them. Thus, Plotter output must also be controlled to match the requirements of the host computer (output rate control). The following discussion explains how to select the optimum method to manage communication between the Plotter and connected devices.

SELECTING A COMMUNICATIONS CONTROL MODE

The following information provides a quick summary of advantages and limitations for each of the communication control modes which provide input rate control and, in some cases, error control of transmissions from an interconnected device to the Plotter. This summary should help select the communications control mode most appropriate for a given application or communication environment.

There are four individual modes; Continuous mode, Hardware Flagging mode, DC1/DC3 Flagging mode, and Block mode. Additional information for each of these modes is presented under *Device-to-Plotter Transmission*, later in this section.

Selecting control methods for Plotter transmissions to connected devices is done separately and described under *Plotter-to-Device Transmission* later in this section.

Continuous Mode

This is the Plotter's default mode upon power-up if Hardware Flagging or DC1/DC3 Flagging modes have not been enabled. Input Rate Control is provided by selecting an appropriate receive baud rate which gives a simple low performance control mode as indicated below. Advantages of Continuous mode are:

- No transmitting device support is required (may use on any system).

Limitations of Continuous mode are:

- Input baud rate must satisfy limitations. For small plots (less than 300 graphic commands) the baud rate used should be less than 2400. For larger plots, the baud rate should be less than 300.

- Provides parity error detection only with no correction of transmission errors.
- Lower input baud rates are required if programmable macro or downloadable characters are defined.

NOTE

The capability of this control mode can be extended if the transmitting device receives Plotter responses. Refer to Device to Plotter Transmission later in this section for more information.

Hardware Flagging Mode

This control mode permits communication at a high baud rate, but requires hardware flagging support in the transmitting device interface. This mode is enabled by the selection of HARDWARE RECEIVE FLAGGING on the Communication Control Mode line of the Parameter Entry Card. Advantages of Hardware Flagging mode are:

- Any available baud rate can be used for small or large plots.
- No transmitting device support unique to the Plotter is required.

Limitations of Hardware Flagging mode are:

- The Plotter and the transmitting device must be directly connected unless the communications devices (such as a modem) are able to pass the RS-232-C DTR (Data Terminal Ready) signal.
- The transmitting device must be able to suspend transmission within 185 characters without loss of characters when inhibited by the DTR signal.
- Provides parity error detection only with no correction of transmission errors.

DC1/DC3 Flagging Mode

This control mode permits communication at high baud rates, but requires DC1/DC3 flagging support in the transmitting device interface. This mode is enabled by the selection of INPUT on the DC1/DC3 Control line of the Parameter Entry Card. Advantages of DC1/DC3 Flagging mode are:

- Any baud rate can be used for small or large plots.
- No transmitting device support unique to the Plotter is required.
- Flagging can be done through a modem connection to the transmitting device.

Limitations of DC1/DC3 Flagging mode are:

- The transmitting device must be able to suspend transmission within 185 characters without loss of characters upon receipt of a $^{D}_{3}$ (stop) character and restart transmission upon receipt of a $^{D}_{1}$ (start) character.
- Provides parity error detection only with no correction of transmission errors.

Block Mode

This is a powerful but more complex control mode that provides the highest performance in all configurations. This mode is enabled when the Plotter receives a BLOCK START command. Refer to *Block Mode Communication* later in this section for a discussion of parameters used with Block mode. Advantages of Block mode are:

- Any baud rate can be used for small or large plots.
- Provides parity and checksum transmission error detection. Also provides error correction by retransmitting blocks of commands.

Limitations of Block mode are:

- This mode requires transmitting device software support which is unique for the Plotter. Refer to *Block Mode Communication* later in this section for software packages that have this support.

INTERFACE SWITCHING FUNCTION

The Interface Switching function provides a flexible, manual means of selecting the source and destination of Plotter data and responses. For example, one can modify the connections between devices without having to physically relocate cables. Refer to *RS-232-C Lines In Loop-Thru Configuration* for a discussion of the effect of the interface switching function upon the RS-232-C lines.

The Interface Switching function is most useful in the loop-thru configuration and is described in respect to this configuration. Two front panel switches (OFFLINE/ONLINE and LOCAL/REMOTE) on the Plotter provide four unique conditions for the interface switching function as shown in Figure 4-7 and described in the following text.

Online/Remote

In this configuration, the Plotter acts as a host peripheral when the terminal and Plotter communicate with the host (Figure 4-7A). Commands sent from the host to the terminal are ignored by the Plotter when it has not been turned on with a DEVICE ON command.

Commands sent from the host to the Plotter are not passed on to the terminal if the Plotter is turned on by a DEVICE ON command and if the AUTO MUTE selection is active on the Parameter Entry Card. In this configuration, both the terminal and the Plotter can transmit to the host.

NOTE

The Plotter and terminal should not transmit to the host at the same time as the transmitted characters could become garbled.

Online/Local

The Plotter and terminal communicate while the host computer is isolated (Figure 4-7B). This configuration switches the Plotter from a host peripheral to a terminal peripheral.

Offline/Remote

The terminal and host communicate while the Plotter remains isolated. (Figure 4-7C). This configuration is used to manually mute the Plotter.

Offline/Local

Here, no communication occurs between devices (Figure 4-7D). The configuration is used mainly for troubleshooting because the Plotter, host computer, and terminal are isolated from each other.

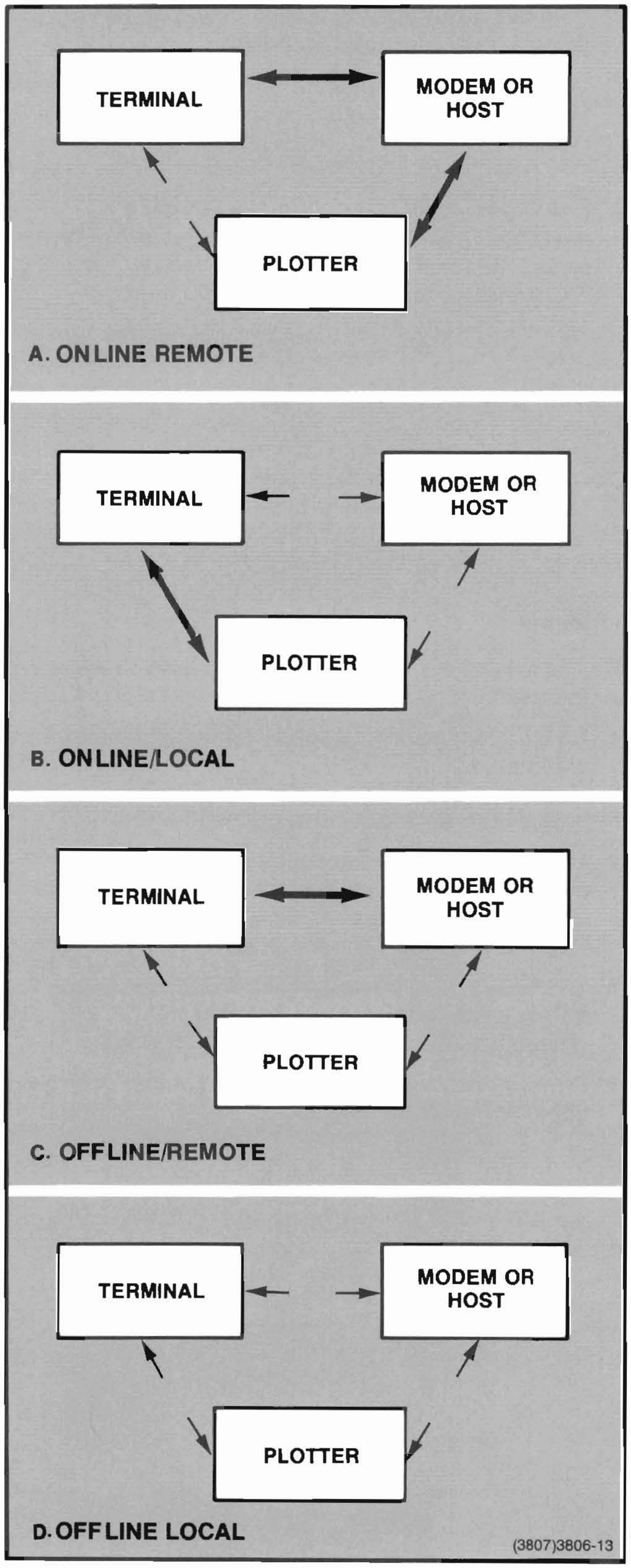


Figure 4-7. Interface Switching Function.

RS-232-C LINES IN LOOP-THRU CONFIGURATION

In the loop-thru configuration, the Interface Switching function actually switches three RS-232-C signal lines: Receive Data (BB), Transmit Data (BA), and Clear-To-Send (CB). Other signal lines are passed straight through the Plotter from the terminal to the modem. These lines are configured so the interface operates correctly with either a modem only, a terminal only, or both a modem and a terminal connected to the Plotter. Refer to the *4663 Interactive Digital Plotter Service Manual* for more details.

The RS-232-C Data Terminal Ready (CD) signal line from the Plotter is connected with the Data Terminal Ready line from the terminal. Both lines must be activated before a ready condition is signaled to the host.

NOTE

Because the Data Terminal Ready line is not switched, the hardware flagging function will only provide rate control for devices attached to the modem connector. Also, if the Plotter switching function is in in LOCAL, the non-switched RS-232 control lines still connect the computer and the terminal. Thus, a state change in these lines may affect the operation of the terminal or of the host even though data is not being exchanged.

For example, a connected modem that is not asserting Received Line Signal Detector (CF — Data Carrier Detect), will inhibit the Serial interface even if the switching function is in LOCAL.

When the Plotter's POWER switch is set to the off position, the Interface Switching function goes into an offline/remote state (allowing the terminal and the host to freely communicate). Refer to Appendix C for definitions of all RS-232-C lines.

DEVICE-TO-PLOTTER TRANSMISSION

The following topics, *Input Buffering, Input Rate Controls*, and *Block Mode Communication*, help describe the transmission of commands from the host computer (or data storage device) to the Plotter. The amount of data (in the form of commands) continuously transmitted to the Plotter at a given baud rate is described under *Input Buffering*. The various transmission rate control modes used with the Input Buffering function (to keep commands from being lost) are described under *Input Rate Controls*. The information under *Block Mode Communication*, describes the various parameters and host computer functions involved in block data transmission. This information can help in choosing the best parameter values for optimum Block mode performance.

Input Buffering

When commands are received from the host computer, they are placed in a temporary buffer in the Plotter's memory. Temporary command storage is required when commands are received faster than the Plotter can execute them. When the Plotter is ready for another command, the stored commands are withdrawn from the input storage buffer and executed. The following paragraphs describe:

- How the Input Buffering function operates.
- How much memory is required to store a Plotter command.
- How much memory is available under various conditions for command storage.

Refer to *Input Rate Controls* for information on controlling the transmission rate to satisfy input buffering constraints described in the following paragraphs.

Buffer Operation. Figure 4-8 illustrates a simplified model of the Plotter input buffers. All commands are received as a series of ASCII characters and are entered into the Input Character Buffer (Buffer A) at the rate the characters are read from the Plotter's communication hardware. The stored characters are then retrieved from Buffer A and converted into a corresponding internal command form. Any Interface commands are executed at once (refer to the Serial Interface Commands section for a listing of these commands).

All other commands are stored in the Input Command Buffer (Buffer B). When the Plotter is ready for another command, these stored commands are retrieved from Buffer B and executed. The Plotter processes Buffer A characters into commands while it is (1) waiting for the current pen carriage motion of the Plotter to be completed, and (2) while it is not servicing the interface hardware. Thus, the rate at which characters are removed from Buffer A depends upon the plot in process and the rate at which characters are being received by the Plotter.

Commands can be sent to the Plotter in *bursts* (which do not exceed the capacity of the Input Character Buffer) until Buffer B is full. Buffer A is of fixed size and can contain up to 248 characters, with each character requiring one byte of storage. For high baud rates (greater than 2400 baud), the command characters usually arrive from the host faster than they can be formed into internal commands. Thus, for these baud rates, the maximum size of a continuous burst of characters sent (including any padding characters, such as ^{N}U (NUL), ^{S}Y (SYN), and ^{D}T (DEL)) must be less than 248 characters. If the number of padding characters added by the host output is hard to determine, use a maximum of 200 to 225 characters.

For moderate baud rates (600 to 2400), transmitted characters are normally formed into commands faster than the Plotter receives them. Thus, Buffer A overflow is not a consideration and commands can be received until the capacity of Buffer B is exceeded. For low baud rates (50 to 300), commands are normally executed faster than they are received, and the buffering function is not active. The input rate controls, discussed later in this section, allow command transmission to the Plotter to be controlled by the input buffering constraints.

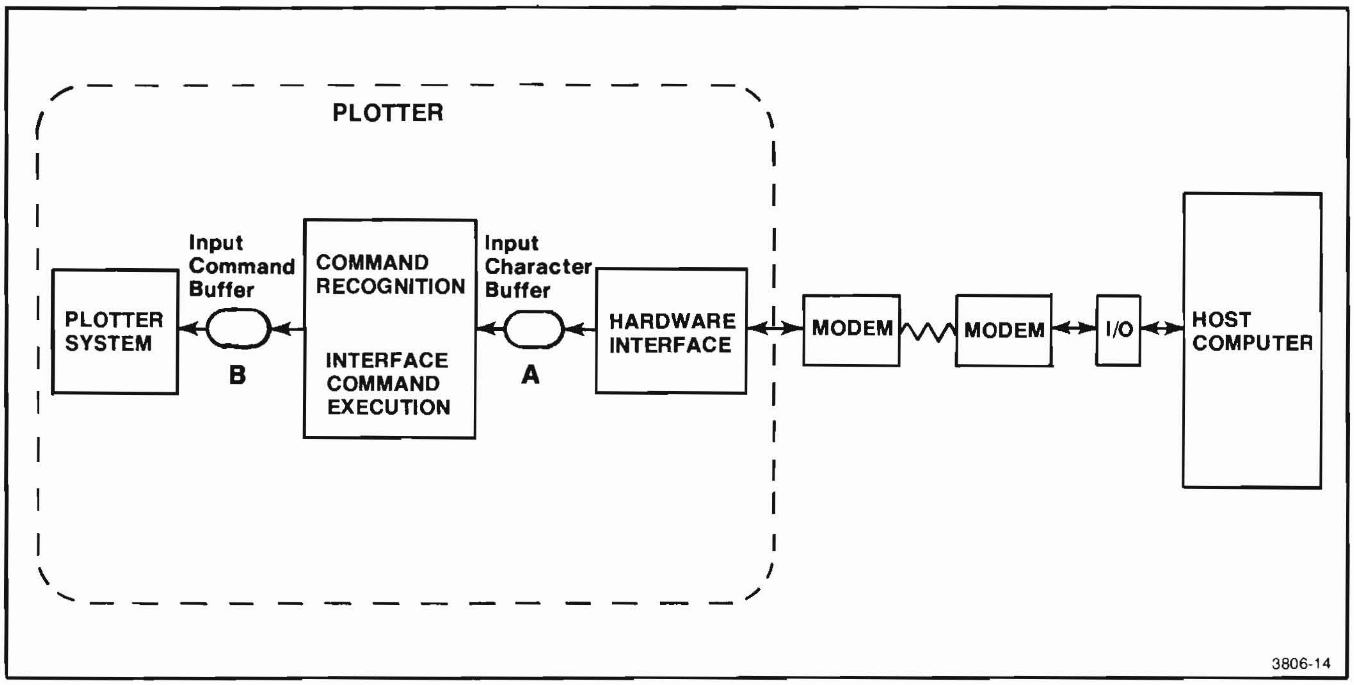


Figure 4-8. Input Buffering Model.

Storage Requirements for Commands. All commands are received by the Plotter in various forms of Style I and Style II commands. These command formats are described under *Introduction To Serial Commands and Responses*, later in this section and should be referred to if you are not familiar with Style I and Style II formats.

The internal form into which Style I Plotter commands are converted requires the following storage for the Input Command Buffer:

- Two bytes of storage for each command code, plus,
- One byte of storage for each character argument,
- And five bytes of storage for each numeric argument.

For commands with string arguments, a separate command is formed for each string character. For commands with a command code followed by multiple groups of arguments, a separate command is formed for each argument group.

The internal form into which Style II Plotter commands are converted requires the following Input Command Buffer storage:

- Three bytes of storage for each PRINT or Attention Action command.
- Twelve bytes of storage for each MOVE, DRAW, CIRCLE, or POINT command.
- Twenty four bytes of storage for each ARC command.

The internal form for each command is the same regardless of the command format used to transmit the command to the Plotter. This means for some cases that more bytes of memory are needed for a command stored in the Input Command Buffer than it took characters to transmit the command to the Plotter. The ratio of these respective number of characters is defined as the *internal expansion factor.* The following examples show some different internal expansion factors:

- A typical Style I PRINT command, such as ^{E}c **A P "X",** takes six bytes to express and three bytes to store, giving an internal expansion factor of 0.5.
- A Style II PRINT command takes one character to express and three bytes to store, giving an internal expansion factor of 3.0.
- A typical Style I DRAW command, such as ^{E}c **A Y 100,10** takes nine bytes to express and twelve bytes to store, giving a 1.33 internal expansion factor.
- A typical Style II DRAW command (for twelve-bit graphics) takes five bytes to express and twelve bytes to store, giving an internal expansion factor of 2.4.
- A typical Style II DRAW command (for ten-bit graphics) takes four bytes to express and twelve bytes to store, giving an external expansion factor of 3.0.
- The worst case is an optimized Style II DRAW command that may, for a short vector, reduce to one character (again requiring twelve bytes to store), giving an internal expansion factor of 12.

Further reference to the *internal expansion factor* occurs under *Block Mode Communications*, later in this section.

Memory Limits for the Input Command Buffer. The Plotter memory available for buffer operation is used in two ways: First, some memory always establishes a minimum size for the Input Character Buffer and the Input Command Buffer. Second, the remaining memory is used on a "first come, first served" basis for (1) the expansion of the Input Command Buffer to store more input commands, (2) the temprorary buffering of responses formed by "output generating commands", and (3) providing longer term storage for the following types of data:

- Graphic transform levels saved by a SAVE CURRENT TRANSFORM command.
- Programmable macros defined by Macro commands. (Option 31).
- Downloadable characters defined by Downloadable Character commands (Option 32).

Refer to individual command descriptions to determine the amount of memory space required for data storage. If memory space is unavailable, the Plotter attempts to temporarily inhibit the transmitting device from sending more commands using the current rate control method. If further commands are received before memory is available, the commands will be lost and an error reported on the Plotter. When stored data is subsequently deleted, the allocated memory becomes available again for other uses.

NOTE

When a buffer overflow error occurs, several commands are usually lost during recovery from the error.

While a programmable macro or downloadable character is being defined, or if most of the available memory has been used to store programmable macros or downloadable characters, the Input Command Buffer is forced to assume its minimum size. The number of commands stored while in this minimum size depends upon the command type. For example, about 200 PRINT CHARACTER commands or about 50 MOVE and DRAW commands could be stored. The number of external characters required to define these commands depends upon the command style being used. If no transforms, programmable macros, or downloadable characters are stored, and no output responses are stored, then the Input Command Buffer can expand to a maximum size. In a standard Plotter with 8K bytes of memory, about 340 MOVE and DRAW commands or 1360 PRINT CHARACTER commands could be stored.

NOTE

The Plotter's Input Character and Input Command Buffers are cleared if the interface is initialized by the front panel Initialize function.

Input Rate Controls

Input rate controls allow the Plotter to control the rate at which it receives commands from the host, terminal, or a data storage device. This control prevents data loss when the Plotter is unable to keep up with incoming commands and when the capacity of the input buffers is exceeded. Four input rate control modes are used with the Plotter; Continuous mode, Hardware Receive Flagging mode, DC1/DC3 Flagging mode, and Block mode. The following describes the operation of the various rate control modes in respect to the input buffering function. A brief summary of these modes can be found under *Selecting A Communications Control Mode*, located earlier in this section.

Continuous Mode. Continuous mode provides only passive rate control and is the default communications mode for the Plotter during the following conditions:

- After Plotter power-up or a front panel Initialize function if a flagging mode has not been selected on the Parameter Entry Card.
- After a DEVICE ON command has been received.

Rate control is accomplished by selecting a host-to-Plotter input baud rate that transfers commands to the Plotter only at the rate the commands can be processed or stored. For short plots (less than 350 graphic commands), baud rates up to 2400 can be used. For large plots, baud rates of 300 or less are normally required. Commands arriving faster than they can be processed are stored in the Plotter as long as memory is available. Received commands exceeding memory storage capabilities are lost and an error is reported.

Active host support is not required for Continuous mode and no unique control operations are performed. Thus, Continuous mode can be used with data storage devices not having active control capabilities. Continuous mode is useful for plotting commands that are generated for a terminal and contain no Block mode rate control commands for the Plotter (such as 4010 Series Style II commands).

If the plot (1) stores several levels of graphic transforms, (2) stores programmable macros, or (3) stores downloadable characters, then the receive baud rate should be reduced to keep the input buffer size to a mimimum while macro or downloadable character definitions are taking place.

Parity Error detection or loss of carrier monitoring are the only transmission error functions available and, with this mode, error correction is unavailable.

If the transmitting device is able to receive Plotter responses, Continuous mode operation may be extended. For example, if an output generating command (such as READ STATUS) is sent to the Plotter after a block of commands, the reception of the Read Status back from the Plotter informs the transmitting device that the input buffer is empty and another block of commands can be sent. This allows operating at higher baud rates and sending plots of arbitrary length.

Hardware Receive Flagging Mode. This mode is used only to control devices attached to the modem connector. Hardware Receive Flagging mode permits any available baud rate for host-to-Plotter transmission to be used. This can be done without concern that a given baud rate is too high for a particular plot operation.

Hardware Receive Flagging mode requires that the host's input/output port be able to suspend transmission without loss of data when inhibited by the RS-232-C signal line Data Terminal Ready (CD). This This restricts the Plotter to a directly connected host unless a modem that is capable of passing the Data Terminal Ready signal to the host is used. Some data storage peripherals can operate in this mode for offline plotting. This mode is also used for plotting commands that are generated for a terminal and contain no Block mode rate control commands for the Plotter (such as 4010 Serial Style I commands).

The Hardware Receive Flagging mode is enabled by selecting HARDWARE RECEIVE FLAGGING on the Communications Control Mode line of the Parameter Entry Card.

When operating in Hardware Receive Flagging mode, commands are received at the selected baud rate and stored as in Continuous mode. If commands are being received faster than they can be processed, the input buffer will fill up. When the Input Character Buffer is about 185 characters from being full, the RS-232-C control line Data Terminal Ready is deactivated to notify the host to stop transmitting. The host computer is then allowed to slow to a stop by completing the transmission of the current line or output buffer. However, transmission should stop within the 185 character limit.

The Plotter then continues to perform commands until the Input Character Buffer is almost empty. The Data Terminal Ready line is then reactivated to notify the host that data transmission may resume.

NOTE

At high baud rates, the Input Character Buffer may fill up, causing flagging to occur even though the Input Command Buffer is not full. (Refer back to Buffer Operation, in this section.) This action occurs because commands cannot be processed into the Input Command Buffer fast enough even if there is room to store the commands. Thus, more flagging action can occur at high baud rates.

DC1/DC3 Flagging Mode. DC1/DC3 Flagging mode operation is identical to the Hardware Receive Flagging mode with the exception that an ASCII $^{\mathrm{D}}_{3}$ (DC3) control character (stop) is sent to the host to suspend data transmission, and an ASCII $^{\mathrm{D}}_{1}$ (DC1) control character (start) is sent to the host to resume transmission. The $^{\mathrm{D}}_{1}$ and $^{\mathrm{D}}_{3}$ control character functions replace the Data Terminal Ready line operation described for Hardware Receive Flagging. In this mode, the host input/output port must be able to suspend data transmission without data loss when it receives the $^{\mathrm{D}}_{3}$, and restart again when the $^{\mathrm{D}}_{1}$ character is received.

Because the flagging is done with characters rather than with control lines, this mode can control devices attached to either the terminal or a modem connector. Some data storage peripherals can operate in this mode. This mode is useful for plotting commands that were generated for a terminal and contains no Block mode rate control commands for the Plotter (such as 4010 Series Style II commands).

DC1/DC3 Flagging mode is enabled by selecting INPUT on the DC1/DC3 Control line of the Parameter Entry Card.

NOTE

If a communications error causes a DC1/DC3 character to be garbled in transmission, the handshake involved with the host computer causes computer communications to stop. If this occurs, (1) manually interrupt transmission of commands from the the host computer, (2) reinitialize the Plotter using the front panel Initialize function, and (3) restart command transmission.

Block Mode. Block mode provides rate control, error detection, and error correction. Block mode is used in any communications environment as no special hardware or communications support is required. However, several host computer functions that need software support are required. These functions and the methods for selecting values for their related parameters are discussed in the *Block Mode Communications*.

In Block mode communication, the host computer divides the commands (to be sent to the Plotter) into blocks of characters and sends these blocks, one at a time, to the Plotter. After each block is transmitted, the host pauses for a response from the Plotter. When the Plotter receives a block, the input buffer is examined for space to store another block. If there is sufficient memory, a positive acknowledge character (an ASCII A) is sent to the host to enable transmission of the next block. (Refer to *Plotter-to-Device Transmission* for a discussion on controlling the output of the acknowledge character.)

If insufficient space exists in the input buffer, the transmission of the acknowledge character is deferred until sufficient Plotter commands are executed to make the required space available. This action matches the effective input baud rate to the rate at which the Plotter processing takes place regardless of the selected input baud rate.

Each block begins with a BLOCK START command and is terminated by a BLOCK END command. The BLOCK END command includes a checksum computed from the contents of the block. When the Plotter receives a block, this checksum is recomputed and evaluated. If the recomputed checksum does not match the checksum in the BLOCK END command, (indicating a transmission problem), the block is discarded and, if sufficient memory is available to hold the block, a negative acknowledgement character (an ASCII I) is transmitted to the host. The host then retransmitts the block and the checksum is evaluated again. This process provides a means of error detection and error correction (by sending the data again). If the error is a persistent one, the host will usually abort the transmission after a few attempts.

NOTE

If a communications error causes a BLOCK START command, a BLOCK END command, or a block acknowledge character to be garbled in transmission, the handshake involved with the host computer will cause computer communications to stop. If this occurs, (1) manually interrupt the command transmission from the host, (2) reinitialize the Plotter with the front panel Initialize function, and (3) restart command tranmission.

Block Mode Communication

This communications mode requires the host computer to perform four related functions: grouping the Plotter commands into blocks, computing a checksum for the block, transmitting the block, and processing the resultant block acknowledge character.

The following paragraphs describe how to determine what Block mode blocksize to use under various conditions. Also described are procedures for assembling Block mode blocks, computing the block checksum, and then using Plotter commands to control Block mode communications. If Tektronix host software support is not used, these procedures must be implemented in a user-written support package containing Block mode host software. In addition, there is a brief description of the available Tektronix host support that can be used with the Plotter.

Maximum Blocksize. Information here describes how to determine the optimum size of the blocks to be used for Block mode communications. A summary of blocksizes at the end of this topic provides conservative blocksize values that can be used when optimum values are not desired.

You must specify the maximum blocksize to the Plotter using a SET BLOCK SIZE command before initiating Block mode. After a block is received from the host and before an acknowledge character is returned (to allow the next block to be transmitted), the current maximum blocksize value is checked to see if another block of this size can be stored.

If a blocksize is not specified, a value of "0" is assumed, which disables the check for available input buffer space (and, therefore, the rate control function). The number of command characters contained in a block between the BLOCK START and BLOCK END commands should be not greater than this specified value. The ASCII characters — NU (NUL), SY (SYN), and DT (DEL if the Parameter Entry Card's DEL IGNORE selection is active), or the command delimiters ";" and CR (when the Command/Response Format is 1 or 2) — need not be included in the count. The actual size of the block transmitted may be any value less than or equal to this size.

If possible, a blocksize should be specified that is small enough to allow some communications overlap. Communications overlap means that one Block mode block is being received while the commands in the previous block are being executed. This procedure provides a continuous flow of commands to the Plotter. However, communication overlap requires that the blocksize be less than one-half the input buffer size (to ensure space is available in time to receive the next block before the commands in the current block are executed).

The maximum blocksize that may be specified when communicating with a standard Plotter (with 8K bytes of memory) is approximately 1,020 characters assuming an average internal expansion factor of 4. (For example, it usually takes four times as many input buffer bytes to store the internal form of the commands as it takes characters in a Block mode block to transmit the command. Refer back to *Input Buffering* earlier in this section for a discussion of the internal expansion factor.)

NOTE

> *Use of large blocksizes can disable any communications overlap. Also, in difficult communications environments where characters are frequently garbled and blocks must be retransmitted, smaller blocksizes should be used to keep the pen in motion.*

When the maximum blocksize value is used to determine if enough memory exists for the next block, an average expansion factor of 4 is assumed. This assumption is sufficient for most plotting. However, there are applications where this assumption is not accurate enough and compensation must be made. One example that allows optional compensation is when the plot commands consist almost entirely of PRINT CHARACTER commands. Here the average expansion factor is actually 3 and allows the host computer to optionally send blocks that are one-third larger than the specified maximum blocksize. However, the maximum blocksize is still about 1020 characters.

Another extreme that requires compensation occurs when optimized Style II commands are sent that include many short vectors, (as in a graphing or contouring application). Here, the internal expansion factor could be as great as 12, which means the actual blocksize transmitted should be no more than one-third of the maximum blocksize specified in the SET BLOCK SIZE command. This reduces the maximum blocksize to about 340 characters.

For many applications, the blocksize value does not have to be precisely specified. If the block expands less than expected, a temporary loss of buffering capacity is the only result. If the block expands more than expected and overflows the available buffer space, the block is discarded. Then the actual size of the block (rather than the specified maximum size) is used to determine when sufficient buffer space is available. Once the space becomes available, the Plotter sends a negative acknowledge character to initiate retransmission of the block. However, if programmable macros or downloadable characters are to be defined, close attention must be paid to the maximum blocksize specification.

During definition of a programmable macro or a downloadable character, the input buffer is forced to a minimum size to make the most memory available for macro or character storage. This minimum buffer size accomodates a maximum blocksize of 150 characters assuming an internal expansion factor of 4. There would be no communications overlap unless the blocksize is less than 75 characters. As noted earlier, if the internal expansion factor is 12 (for optimized 4014 short vector graphics), minimum buffer size operation reduces the maximum blocksize to 50 caracters. Blocks that expand more than specified (or expected) can normally be accomodated if there is additional memory that has not been filled with macro or character storage. Blockmode communications may fail if a large portion of the available memory is used to store macros or downloaded characters and the blocksize and expansion factor constraints listed above are not observed.

When SAVE CURRENT TRANSFORM commands are sent to the Plotter, care must be taken that sufficient memory exists to store each transform (a transform requires about 30 bytes for storage). One or two transform levels are always available under all conditions (even if the memory is full of programmable macros or downloadable characters). If several levels are to be saved, then pausing or sending a READ STATUS command after the SAVE CURRENT TRANSFORM command would ensure that the input buffer is of minimum size and that maximum memory is available to store the transform levels.

For operation at high baud rates (greater than 2400 baud), the buffering constraints dictate that the maximum burst of character size transmitted to the Plotter (including the BLOCK START and BLOCK END commands and the total number of characters within the block where ᴺᵤ (NUL), ˢʏ (SYN), and ᴰʟ (DEL) padding characters are included) must be less than 155. (Refer to *Input Buffering* earlier in this section) Therefore, when using high baud rates (greater than 2400 baud), the recommended maximum Block mode blocksize should be 155 characters minus the expected number of padding characters.

NOTE

> *Either Hardware Receive Flagging mode or DC1/DC3 Flagging mode can be used with Block mode communications to avoid Input Character Buffer overflows. This permits using larger blocksizes even at high baud rates.*

The requirements for maximum blocksize are summarized as follows (with more restrictive operation using a smaller blocksize):

- Standard maximum blocksize is less than 1020 characters.
- Use a maximum blocksize of 155 characters if:
  (1) the majority of memory is filled with programmable macros or downloadable characters, or if
  (2) the baud rate is greater than 2400 baud.
- If communications overlap is desired, reduce blocksizes listed above by one-half.
- If Plotter commands are mostly Style II PRINT commands, increase standard blocksize by one-third.
- If Plotter commands are mostly optimized Style II MOVE or DRAW commands that specify short motions, reduce standard blocksize by two-thirds.

Block Contents. Blocks may include any Plotter command except DEVICE ON, DEVICE OFF, or DEVICE RESET. However, interface commands should not be included in a block. These commands would be executed as soon as they were recognized by the interface, but they cannot take effect in time to control the transmission of the acknowledge character for that block. Therefore, the interface commands used to establish communication parameter values should be transmitted in Continuous mode before initiating Block mode.

Include only complete Style I commands in a command block. (For example, BLOCK START or BLOCK END commands cannot appear within other Style I commands). Style II commands may include Style I commands and thus be split between two blocks. Refer to *Introduction to Serial Commands And Responses* for more discussion of Style I and Style II commands.

Block Mode Checksum. The checksum contained in the BLOCK END command is computed by adding each 7-bit character of the block into a 12-bit accumulator with an end-around-carry of any overflow. The block characters included in the checksum extend from and include the "(" in the BLOCK START command to the ")" in the BLOCK END command, as shown in Figure 4-9.

All block characters contained between the BLOCK START and BLOCK END commands are included in the checksum, except for the following:

- The $^{N}_{U}$ (NUL) character.
- The $^{S}_{Y}$ (SYN) character unless the $^{S}_{Y}$ character has been selected on the ATTENTION CHARACTER line of the Parameter Entry Card.
- The $^{D}_{L}$ (DEL) character if the DEL IGNORE has been selected on the Interface Functions Line on the Parameter Entry Card. This allows systems that use $^{D}_{L}$ characters for padding to be accomodated. An $^{A}_{T}$ **?** character sequence should be substituted for the $^{D}_{L}$ character in Style II commands if this selection is made.
- Any $^{D}_{1}$ or $^{D}_{3}$ characters.

The 12-bit accumulator is initially set to "0". If the addition of a 7-bit character causes the accumulated value to exceed 4095 (the maximum 12-bit number), 4095 (−4096+ 1) is subtracted from the accumulated value to accomplish an end-around-carry from the 12th bit. When all characters have been added, the checksum value is converted to an ASCII decimal number and transmitted as a BLOCK END command argument. When the block is received by the Plotter, the checksum is recomputed using the same procedure. After the ")" character has been added, the checksum value should be the same as the checksum value transmitted from the host. If a value of "0" is specified for the checksum, the checksum error detection function is disabled and a positive acknowledge is automatically generated.

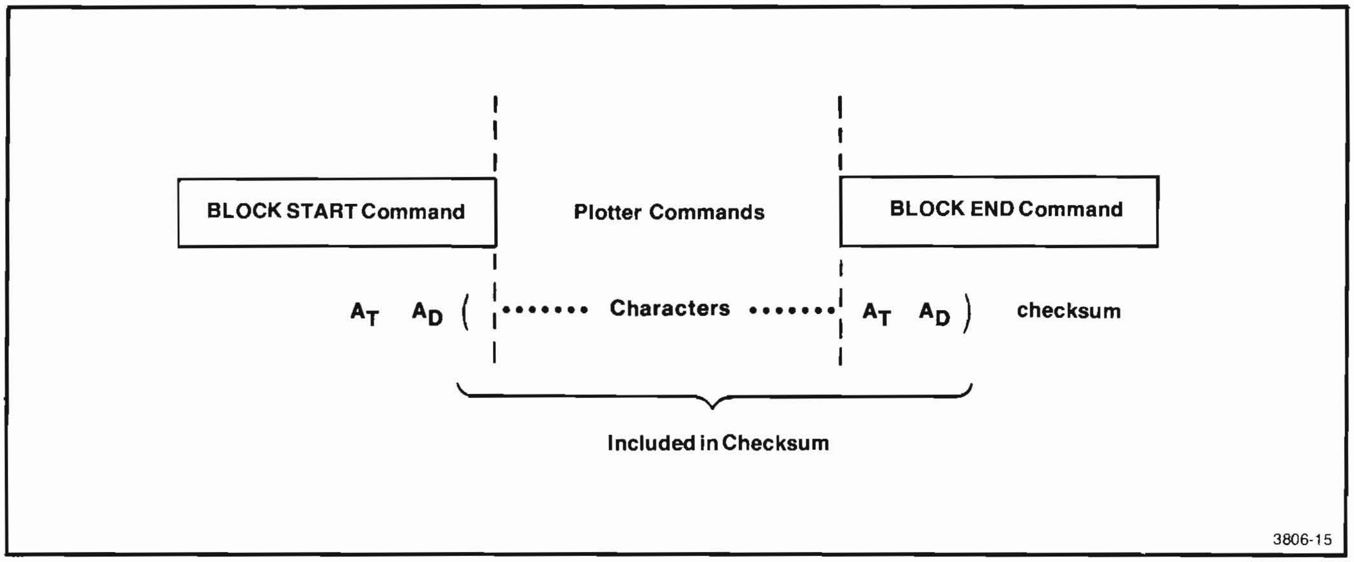


Figure 4-9. Block Mode Checksum.

Turning Block Mode On and Off. Described here are procedures for enabling and disabling Block mode communications. Figure 4-10 is a state transition diagram that illustrates the relationship between Block mode and Continuous mode. Each circle in the diagram represents an interface state where certain processing steps (described later) are performed on the received commands. Each arc connecting the circles represents a change of state when the specified command is received.

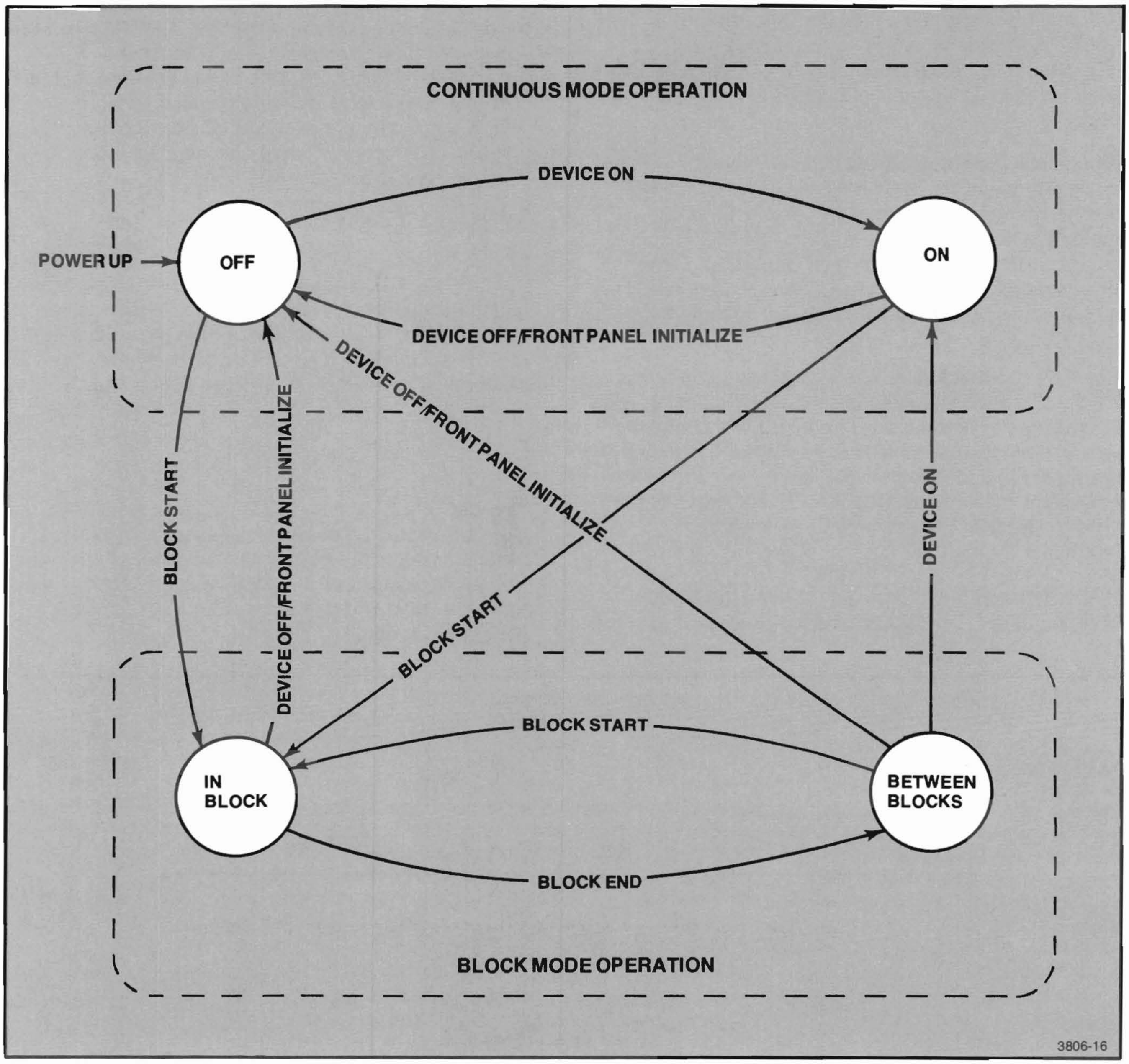

CONTINUOUS MODE OPERATION
DEVICE ON
POWER UP
OFF
ON
DEVICE OFF/FRONT PANEL INITIALIZE
DEVICE OFF/FRONT PANEL INITIALIZE
DEVICE OFF/FRONT PANEL INITIALIZE
BLOCK START
DEVICE ON
BLOCK START
BLOCK START
IN BLOCK
BETWEEN BLOCKS
BLOCK END
BLOCK MODE OPERATION
3806-16

Figure 4-10. Communications Mode State Diagram.

NOTE

> *Hardware Receive Flagging mode and DC1/DC3 Flagging mode are not included in this diagram. This is because the two flagging modes operate independent of Block or Continuous mode.*

Continuous mode automatically occurs at Plotter power-up or when a front panel Initialize function is selected. Continuous mode is represented by the Plotter On and Plotter Off states. In the Plotter On state, commands are recognized and executed as they are received.

If a BLOCK START command is received, Block mode communication is initiated (the In Block and Between Blocks states). Communication continues in Block mode until either a DEVICE ON or DEVICE OFF command is received or the interface is reinitialized. While in the In Block state, the interface recognizes the commands, stores them in the input buffer (without executing them), and accumulates the checksum.

After the BLOCK END command is received, the checksum is verified and the positive or negative acknowledge character is generated. If the acknowledge was negative, the stored commands are discarded. If the acknowledge was positive, the commands are executed. While in the Between Blocks state, all characters and commands except the Bypass Cancel character, the prompt string (see *Plotter-To-Host Communications*), and the DEVICE ON, DEVICE OFF, or BLOCK START commands, are ignored. This state allows the interface to ignore any unexpected characters transmitted by the host between Block mode blocks.

Existing Plotter Support Software. Several host support software packages are available from Tektronix and can be installed on most host computers to manage Block mode communications as well as provide other Plotter and terminal support functions. These are components of the TEKTRONIX PLOT 10 Graphics Library. These packages provide capabilities from simple graphing to state-of-the-art graphics. A set of utility routines interface these packages to the Plotter. For further information about these software products, contact the local Tektronix sales office.

PLOTTER-TO-DEVICE TRANSMISSION

The Plotter transmits several types of information to the host computer in response to Plotter commands. The following paragraphs discuss the transmission of output information, including:

- How output blocks are generated.
- How the host can identify the output blocks that are received.
- How output blocks are stored and grouped into messages for transmission.
- How initiation of the output transmission is controlled for compatibility with the host computer.
- How the output transmission rate is controlled.
- How the Plotter bypass function disposes of output message characters echoed by the host.

Output Blocks

The Plotter generates two classes of outputs: communications control information and data generated in response to Plotter commands. Each of these output classes, the forms in which they are transmitted, and the priority of their transmission are described in the following paragraphs.

One type of communications control information is the DC1/DC3 characters required for DC1/DC3 flagging rate control. This information is transmitted as single characters rather than as output blocks. Another type of communications control information is the Block mode acknowledge characters. These characters are transmitted as special output blocks consisting of only the acknowledge character.

The other class of Plotter output (responses generated by Plotter commands) is transmitted as one or more output blocks. Each output block contains two numeric response values plus a *TAG value*. For commands whose response consists of one value (such as READ FORM LENGTH), or two values (such as DIGITIZE) only one output block is generated. For commands that return more than two response values, two or more output blocks are generated. If a response requires only one numeric value, the other value is set to "0".

The *TAG value* is the third numeric value generated along with response values. For Command/Response Formats 1 and 2, the TAG value may indicate an additional characteristic of the two response values, as well as a LAST block, a NOT LAST block, or CONTROL designation for the block. The meaning of any additional characteristics are defined for each command response. For example, the coordinate values in a response block generated by a DIGITIZE command may be designated as the end point coordinates of a Move or a Draw.

The TAG value designations for different types of blocks are:

- CONTROL for Block mode acknowledge output blocks (the only control block defined).
- LAST BLOCK for single block responses or for the last block of a multiple block response.
- NOT LAST for all other response blocks.

For Command/Response Formats 3 and 4, the TAG value uniquely identifies the response block in terms of the initiating command. Refer to *Output Responses* near the end of this section for more information on TAG values for all Command/Response Formats.

The size of these response blocks can vary from 7 to 33 characters, depending upon the current Command/Response Format and the numeric values of the responses. Refer to the command description sections for a discussion of response value formats set up for each response block and the accompanying TAG values.

Groups of output blocks sent to the host are referred to as output messages. Each output message consists of three parts, which must be transmitted in the following order:

1. A signature character (if one has been defined via a SET SIGNATURE CHARACTER command).
2. The response blocks ready to be transmitted followed by a control block (if any).
3. One or two output termination characters depending upon the selection made on the Output Terminator line of the Parameter Entry Card.

The actual coding of the responses and the actual characters transmitted in the output message depend upon the current selection of the Command/Response Format.

Output Block Identification

When the host receives an output response block, you must identify the response and associate it with the output generating command that created the response. If a single output generating command has been sent, identification is easy. If multiple output response blocks are received, refer to the following guidelines to help identify the output generating commands.

- For Command/Response Formats 1 and 2, response blocks are *unlabeled* (the response block itself does not indicate the command that generated it, for example, a READ FORMLENGTH command or a DIGITIZE command). Thus the order in which responses are received is critical.
- Output responses are always generated in the order that the generating commands were received (with the exception of Block mode acknowledge responses, which are generated before responses to commands within a block are generated). In addition, all blocks of a multiple block response are generated before any other command responses are started.
- A TAG value of LAST is always present in the last block of a response.
- For Command/Response Formats 3 and 4, the response blocks are labeled and can be easily identified by their TAG values.
- The use of a unique signature character permits the host to identify the source of response blocks when multiple plotters are being used.

Output Buffering

As the Plotter's outputs are generated, each output is stored until all transmission constraints are satisfied. These contraints are discussed in more detail later in this section, Only one flagging control character and one Block mode acknowledge character can be ready for transmission at any one time. For that reason, they are stored as individual pieces of data. However, multiple response blocks can be ready and stored in the Plotter's output buffer until transmission is initiated.

NOTE

The memory utilized to store output responses reduces the amount of memory available to store commands in the input buffer, programmable macros, downloadable characters, and transforms. Therefore, buffering of output responses should be kept to a minimum.

Normally the host waits for the response of one output generating command to be returned from the Plotter before sending another output generating command (although several outputs can be generated and returned to the host together). Enough memory is reserved for output storage so that a single response block can always be buffered even if all the remaining memory has been allocated to other functions. Up to 127 response blocks may be stored if memory is available.

The amount of buffer space required to store responses is the sum of the number of bytes in each response block plus one byte per response block.

Because memory is allocated to buffering functions in 62-byte increments, the amount of memory required to store this number of bytes is determined by rounding the number of bytes up to the nearest multiple of 62. The output buffer is cleared whenever a front panel Initialize function is performed.

Formation of Output Messages

The way in which types of buffered output are returned to the host are determined by two considerations: the priority of the output and any possible limitations of the host computer in receiving the response.

The buffered output blocks are grouped into appropriate output messages to satisfy these considerations. This output can consist of DC1/DC3 control characters, control blocks, or response blocks.

DC1/DC3 flagging control characters have the highest priority for transmission from the Plotter. Because the host computer continually scans for the DC1/DC3 characters, the Plotter can transmit them at any time regardless of any current output activity. When transmission is not in process, the flagging character will be transmitted immediately, ignoring any output initiation constraints. When transmission is currently in process, the flagging character will be the next character transmitted, even if this places the character within an output block. The signature character or the output termination characters are not added to the flagging characters when they are transmitted.

The next higher priority is any pending control block (Block mode acknowledge character). A pending control block is always contained in the next output message sent when the output initiation constraints are satisfied.

NOTE

The Block mode acknowledge character generated when a BLOCK END command has been received by the Plotter is always transmitted before any responses that are generated by commands contained within the block.

Some line-oriented host computers store only incoming data and inhibit processing until an output (End-of-Line) terminator is received. As the block containing the Block mode acknowledge character is a control block, the information needs to be processed quickly and is always followed by the currently defined Output Terminator characters selected on the Parameter Entry card. These characters define the end of the output message.

The lowest priority is the transmission of pending response blocks. When the higher priority messages have been transmitted, any pending response blocks are formed into messages and transmitted when the output initiation constraints have been satisfied.

Many host computers also have limitations on the length of the line received. Thus, when transmitting output responses, the output termination characters are inserted after the appropriate number of response blocks so that a single output message will contain no more than 62 characters (including the output termination characters). This is a maximum line length value acceptable to most host computers.

Output Initialize

In many cases, due to hardware or software limitations, the host computer can only receive responses from the Plotter when it is in a ready-to-read state. This case can occur even if the Plotter is connected to the host via a full duplex communication line. Responses transmitted to the host computer at other times could be lost. Two output initialize mechanisms are available in the Plotter to control initializing response transmissions to ensure that the host is ready to read. These are the Turnaround Delay function and the Prompt function. Either one or both functions may be required to match the Plotter to the requirements of a particular host. Once these mechanisms are satisfied, transmission is initiated and continues until a complete output meassage (ending with the current output termination characters) has been transmitted.

Turnaround Delay Function. This function accommodates host computers that are not able to read for a short interval after each time they transmit. The Turnaround Delay function is enabled when the SET TURNAROUND DELAY command changes the Turnaround Delay value to a number greater than 0. This function forces software half duplex operation so the host computer must stop transmitting for a time equal to the Turnaround Delay before the Plotter can initiate transmission. Each time any characters (including padding characters) are received from the host, the time period is restarted. This time period allows the host software to switch from Send mode to Receive mode.

Some host computers cannot read for a short period after they receive an output (End-of-Line) terminator. The Turnaround Delay function also accommodates this restriction by inhibiting the start of transmissions of a following output message (for the Turnaround Delay period) after the previous output message has been transmitted, even if all other transmissions have been satisfied.

Prompt Function. The second mechanism controlling output initiation is the Prompt function. Here, prompts accommodate host computers that exhibit a varying time period between the end of a host transmission and when the host is ready to receive information. This function is enabled when the SET PROMPT STRING command changes the prompt string to a non-null value. The Prompt function is used when communicating with host computers that transmit one or more prompt characters to indicate they are ready to receive.

The prompt strings can be up to four characters long and contains any ASCII characters. These characters must be received successively to be recognized (for example they must not be separated by other characters such as the $^{N}_{U}$, $^{S}_{Y}$, or $^{D}_{L}$ padding characters). The Plotter scans all characters received from the host for a prompt string when one or more output generating commands have been received and not all generated output has been transmitted. When a prompt string is recognized, the string characters are deleted from the Plotter input (that is, they are not processed as command characters).

The prompt string is not recognized within a block during Block mode. The Plotter recognizes a prompt string whenever prompt scanning is enabled. However, only one prompt string at a time is recognized because prompts cannot be queued up. Only after initiating transmission of the corresponding output messages will the next prompt be recognized.

Remember that BLOCK END commands (which allow a block acknowledge to be generated) and OPERATOR DIGITIZE ENABLE commands (which allow the operator to generate digitize response blocks) are included in the output generating commands category.

Command/Response Formats 3 and 4 allow the OPERATOR DIGITIZE function to be permanently enabled. This permits digitize response blocks to be generated without a corresponding output generating command to enable the prompt scan. For these formats, the PROMPT LIGHT ON command can be used to enable the prompt scan if operator digitized output is expected but not yet generated. The prompt scan is enabled only by an external PROMPT LIGHT ON command and disabled by a PROMPT LIGHT OFF command. Other Plotter functions may turn off the Prompt light, but they will not disable a prompt scan.

Combined Output Initialize. A combination of the Turnaround Delay and the Prompt mechanisms allow constraints of most host computers to be accommodated. If a Turnaround Delay or Prompt string is not defined, then the host computer is assumed to have full duplex capability and transmission is started for each output block as soon as it is generated. Thus, each block constitutes an output message and is terminated by the current output termination characters.

If a Turnaround Delay or a Prompt string is defined, then software half duplex operation is assumed and the output blocks are buffered until the transmission constraints are satisfied and transmission can be started. For example, the host has transmitted the Prompt string and has stopped transmitting for the Turnaround Delay period.

As an example, the Plotter is communicating with a host under software half-duplex constraints and has received a Block mode block containing a READ STATUS command and a DIGITIZE command. The following sequence of events can occur:

1. The block is processed and its checksum verified.
2. If OK, a positive Block mode acknowledge control block is generated.
3. When the Prompt string has been received from the host computer and the Turnaround Delay has expired, the control block is transmitted as an output message followed by any current output termination characters.
4. At the same time, the commands within the block are processed, and the two command response blocks are generated.
5. When the next Prompt string is received from the host and the Turnaround Delay has again expired, the two response blocks are transmitted as an output messaqe.
   - If the host computer responded quickly with the second prompt before the second response-generating command was processed, the output message might contain only the first response block, and thus require another prompt from the host computer to cause both responses to be transmitted.
   - If the host computer responds slowly and the first prompt is received (after the output generating commands are processed and the command response blocks are generated), the block acknowledge is still transmitted as the first message (because a control block must be followed by an output terminator). The next prompt then causes the two response blocks to be transmitted.

Output Rate Control

Normally, the transmit baud rate is set to the maximum value usable by the communications hardware so Plotter responses are quickly sent to the host. However, if the transmit baud rate is dictated by hardware communication capabilities, and if the host cannot accept a complete response message at the dictated baud rate (due to software limitations), then the Transmit Baud Rate Limit function on the Parameter Entry Card can be used to reduce the effective transmit baud rate. (Refer to *Setting Up The Parameter Entry Card For Serial Operation* earlier in this section for more information on this function.) Other ways to control the effective transmit baud rate are through the DC1/DC3 Control line of the Parameter Entry Card or by using Hardware Output Flagging. These methods can be used only when the host accepts a limited number of characters without a pause during input. The following paragraphs discuss these control methods.

DC1/DC3 Control. If the host computer can utilize DC1/DC3 flagging control, then OUTPUT can be selected on the DC1/DC3 Control line of the Parameter Entry Card to allow host control of the Plotter output transmission.

When the host needs to inhibit the Plotter output, it sends an ASCII ␓ (stop) character to the Plotter. (This character can be merged anywhere into other output currently being transmitted to the Plotter with the exceptions of a DEVICE ON or DEVICE OFF command.) When the ␓ character is received by the Plotter, transmission of further output characters is inhibited after any character currently in process has been completed. When the host is again ready to receive, it sends an ASCII ␑ (start) character to the Plotter to continue transmission of any other output characters.

The output is temporarily enabled to allow the ␓ character to be sent to the host if the following three conditions occur:

- Input and output DC1/DC3 character flagging is enabled simultaneously.
- The Plotter needs to send a ␓ to inhibit output from the host.
- The host has inhibited output from the Plotter at the same time.

Hardware Output Flagging. If the host's interface hardware can control the RS-232-C Clear To Send signal line, the host can also temporarily inhibit the transmission of output characters from the Plotter by unasserting this signal line. Asserting the signal line again will continue transmission. This function is always enabled, requiring the RS-232-C Clear To Send signal line to be asserted before any Plotter output can be transmitted.

The Bypass Function

Many host computers used with terminals send back (echo) the received characters for display on the terminal. But if output responses from the Plotter are echoed back, the characters are processed as commands. To avoid this situation, the Bypass function directs the Plotter to ignore the echoed characters. This function is enabled when the SET BYPASS CANCEL CHARACTER command changes the bypass cancel character to a non-null character.

If a response is to be sent to the host computer, bypassing begins when transmission is started and continues until the bypass cancel character is received from the host. For example, if the current output termination character is a CR, and the host adds an LF character following each CR, the bypass cancel character is set to LF. Then the entire output message, sent from the Plotter to the host, would be echoed by the host and ignored by the Plotter. The final character echoed would be a CR and the following LF character added by the host computer would end bypassing and allow further commands to be received by the Plotter.

NOTE

Characters echoed while bypass is in effect do not reset the Turnaround Delay function.

COMMUNICATING WITH MULTIPLE PLOTTERS

During host communication with multiple Plotters in a loop-thru configuration, the following communication requirements and restrictions apply:

- Only the Plotter closest to the terminal should have AUTO MUTE selected on the Parameter Entry Card.
- Continuous mode and Hardware Receive Flagging modes can be used for rate control and allow all Plotters to have the same device address and to receive commands in parallel.
- DC1/DC3 Flagging mode cannot be used for rate control in multiple Plotter configurations.
- Block mode can be used if each Plotter has a unique address. Each Plotter must have a unique prompt string and be individually prompted for the block acknowledge response.

INTRODUCTION TO SERIAL COMMANDS AND RESPONSES

This section describes how to form commands that control the Plotter and how to interpret responses sent by the Plotter. To initiate an action, commands are sent to the Plotter from a host, a terminal, or a data storage device. This action, for example, may be drawing a line or generating a response to be sent back to the host computer.

The Plotter provides four Serial command and response formats to accommodate most connected devices or host computers.

The actual form of the commands and responses you use is determined by the current Command/Response Format selection. Refer to *Setting Up The Parameter Entry Card* earlier in this section for a discussion on how to select an appropriate Initial Command/Response Format.

Commands recognized by the Serial interface are of two general types: *Style I commands* and *Style II commands*.

- Style I commands are unique Plotter commands that use ASCII decimal data coding. These commands are easily generated and more readable.
- Style II commands use binary data coding and are efficiently transmitted. Style II commands, however, are more difficult to generate and cannot be read easily. Style II commands are used with the TEKTRONIX 4010 Series Terminals.

The following describes Style I and Style II commands separately and then shows how both command styles can be combined to control the Plotter.

NOTE

As characters received by the Serial interface are processed into commands, any ${}^{N}{}_{U}$, ${}^{S}{}_{Y}$, ${}^{D}{}_{L}$ (if DEL IGNORE is selected on the Parameter Entry Card), ${}^{D}{}_{1}$, ${}^{D}{}_{3}$, or characters of a prompt string (when Prompt scanning is enabled) are processed separately. These characters can appear anywhere within command definitions with the exceptions of the DEVICE ON or DEVICE OFF commands.

STYLE I COMMANDS (PLOTTER COMMANDS)

Style I commands are easy to generate either by the host computer or by manually entering them at a terminal. The general form of Style I commands is:

<attention character> <address character> <command code> <arguments>

Each part of the Style I command is discussed in the following paragraphs along with any separators and terminators required to correctly punctuate the command. Also presented is Plotter response to incomplete or unrecognized commands.

Attention Character

Each Style I command begins with an attention character (${}^{A}{}_{T}$) as selected on the Attention Character line of the Parameter Entry Card. The chosen character should be unique and not be used for other purposes except within delimited print string arguments or as part of Style II Attention Action commands. Use of an attention character permits the command to be merged into a text or graphics message being sent to the Plotter as Style II commands.

Address Character

The address character is an uppercase ASCII character that must agree with the address character chosen on the Parameter Entry Card. The address character allows commands to be directed to a single Plotter even when multiple Plotters are connected to the communication line. If each of the multiple Plotters have the same address, then all the Plotters can receive the commands in parallel.

Command Code

The command code consists of one or two ASCII characters (either uppercase or lowercase) that uniquely identifies the command. The command code for each command is defined in the command description sections of the manual. The command code can be followed by a SP character instead of a command delimiter.

Arguments

Any arguments follow the command code. Specific arguments for each command are described in the command description sections of the manual. There are two types of command arguments: numeric or string. Individual arguments are partitioned from each other by argument separators and the end of an argument set is marked by a command terminator. Refer to *Argument Separators and Command Terminators* later in this section.

Numeric Arguments. Numeric arguments can be expressed in any of three standard forms: integer, floating point, or scientific notation. The following examples show each form:

- −15 (Integer)
- +15.8 (Floating Point)
- 0.158E+01 (Scientific Notation)

The following rules apply when entering numeric arguments:

- When a numeric argument is positive, the "+" sign is optional.
- Leading zeros are ignored.
- In the scientific notation form: the mantissa can be expressed in either integer or in floating point form.
  - The E can be entered either in uppercase or lowercase and the exponent can be expressed in 1 or 2 digits.
  - A "+" sign is not required if the exponent is positive.
  - A space can be substituted for a "+" sign in either the mantissa or the exponent. No other spaces should appear within the argument.
- A numeric argument is terminated by any character that is not part of a specified numeric form; for example, a character that is not one of the digits 0 through 9, and the characters + — e E and a period.

These numeric argument forms accomodate the number formats specified in ANSI Standard X3.42. The units, resolution, and allowable range of numeric arguments are unique to each command and are noted in the command description sections. A selector argument is a special type of numeric argument that specifies a prenumbered choice, which is defined for some commands.

String Arguments. A *string argument* represents a sequence of one or more ASCII characters in one of two forms: as a delimited string or as an undelimited string. Three simple examples of the *delimited string form* are shown below, each representing the ASCII character "A".

- **"A"**
- **'A'**
- **/65**

A delimited string can be formed by preceding and following a group of one or more ASCII characters by the same delimiter (either ’ or “); alternatively, a delimited string can consist of a / character (slant) followed by the numeric argument that is equal to the ADE (ASCII decimal equivalent) value assigned to a given ASCII character. The ADE values for each of the ASCII characters are shown in Appendix B.

The delimited string representation can be used to specify any ASCII characters with the following exceptions:

- The string delimiter character.
- Characters that cannot be transmitted by the host computer interface.
- Characters having communications functions. These are $^{N}{}_{U}$, $^{S}{}_{Y}$, $^{D}{}_{L}$ (if DEL IGNORE is selected), $^{D}{}_{1}$, or $^{D}{}_{3}$.
- The characters of a prompt string (when prompt scanning is enabled).

The ADE representation can be used for any ASCII character. You can mix the delimited string and ADE representations to form a single, more complex delimited string. Additional examples of delimited strings follow:

Delimited String	Characters Represented
“ABC”	ABC
‘AB’/13/10‘CD’	AB$^{C}{}_{R}{}^{L}{}_{F}$CD
‘He$^{S}{}_{P}$said:$^{S}{}_{P}$“I$^{S}{}_{P}$can’“”“’t“’	He $^{S}{}_{P}$ said:$^{S}{}_{P}$ “I $^{S}{}_{P}$ can’t”

An *undelimited string* is a sequence of ASCII characters, conforming to the following restrictions:

- The first character cannot be one of the following characters:
  - The / (slant)
  - The ’ (apostrophe)
  - The “ (quote mark).
- The string cannot include any of the following characters:
  - $^{S}{}_{P}$
  - , (comma)
  - $^{D}{}_{1}$
  - $^{D}{}_{3}$
  - $^{N}{}_{U}$
  - $^{S}{}_{Y}$
  - $^{D}{}_{L}$ (if DEL IGNORE selected)
  - $^{C}{}_{R}$
  - ; (semicolon)
  - $^{A}{}_{T}$

There are two exceptions to these rules that apply only to Command/Response Formats 3 and 4:

- An undelimited string can include a semicolon
- The undelimited string is limited to a single character. (The string is automatically terminated after one character.)

NOTE

The TEKTRONIX 4662 Plotter does not accept delimited strings. Thus, when generating commands intended for either the 4662 or 4663 Plotter, undelimited strings must be used. When generating commands for the 4663 Plotter only, the more general delimited string form can be used.

Argument Separators

Argument separators are special sequences of characters that define the end of one argument and the beginning of the next. A space or a comma character are the most common argument separators. However, an argument separator can also be expressed in one of two more general forms:

- A sequence of one or more spaces, or
- A sequence of any number of spaces (or none) followed by a comma, followed by another sequence of any number of spaces (or none).

NOTE

The argument separator implies that another argument follows (if it is allowed). Therefore, be sure that the last argument of a command is not followed by a space or comma character (which would be interpreted as an argument separator).

Command Terminators

A *command terminator* is a character that marks the end of the command's last argument and indicates that no more arguments follow. A command is not executed by the Plotter until the command has been terminated. There are two general types of command terminators defined: discarded terminators, and reprocessed terminators.

Discarded command terminators explicitly delimit the command and, after they are recognized, are discarded without further interpretation. The discarded command terminators are the semicolon character and the CR character.

Reprocessed command terminators are characters which terminate a previous command but are also used as part of the next command. An example is when the attention character for the next command terminates the previous command. Take care to properly terminate the last argument of a command.

NOTE

The characters NU, SY, DL (if DEL IGNORE is selected on the Parameter Entry Card), D1, D3, the characters in a prompt string, or the bypass cancel characters (when bypass is active) will not act as command terminators.

For Command/Response Formats 3 and 4, a semicolon character or a CR character are treated as reprocessed command terminators (possibly as part of a Style II command). Formats 3 or 4 have no discarded command terminators defined. There are several nonprinting characters, such as SH, which do not affect Plotter output and can be used as reprocessed command terminators (if they can be transmitted by the host computer). Refer to Table I-1 in Appendix I for an overview of characters that do not affect Plotter output. A CLOCK END command can be terminated by any printing or nonprinting character (as long as the character terminates a numeric argument).

A command is not executed by the Plotter until it has been terminated (see following NOTE). Thus, the final command in a group of commands should be explicitly followed by a terminator.

NOTE

For Command/Response Formats 3 and 4, commands which have no arguments require no command terminators. For Formats 1 and 2, a command terminator is optional for commands without arguments.

Multiple Argument Groups

A complete set of arguments for a command is referred to as an argument group. For some commands, multiple groups of arguments can follow the command code when multiple occurrences of the command are desired. An example is a DRAW command that may be followed by multiple pairs of X,Y coordinates. The final argument group is followed by a command terminator. The specific commands that allow multiple argument groups to be specified are called out in the individual command description sections.

NOTE

Commands allowing multiple argument groups should be carefully terminated. For example, if you use a space or comma character following the command, the Plotter assumes more arguments follow, which causes undesired results.

Null Arguments

You can specify a null argument in place of an expected numeric or string argument. A null numeric argument is given a zero value and a null string argument is given a value of a single ASCII character NU. Null argument values are established by simply not specifying the argument and including only the argument separator (or terminator).

NOTE

The argument separators must be unique; for example, two consecutive commas will specify a null argument, but two consecutive spaces will not. A space following a command code is interpreted as a command code delimiter and not as an argument separator defining a null argument. Thus, the argument separator must be a comma if a leading null argument is to be specified.

Optional Arguments

Some Plotter commands have optional arguments. These optional arguments can be eliminated by entering a command terminator before the optional argument position instead of a command separator. The effects of optional argument selection for such commands are discussed in the individual command description sections.

Command Interpretation

The following outlines the Plotter's responses to invalid Style I commands.

- If the attention character is garbled or omitted, the beginning of a Style I command is not recognized and the remaining command characters are processed as Style II command characters.
- If the address character is valid (uppercase A through H), but is not the current Plotter address character, the command is ignored. Commands are ignored by discarding command elements (such as the command code and arguments) until the command is terminated.
- If the address character is not valid, the AT character sequence is checked for a Style II Attention Action command. If a sequence is not recognized, the characters are discarded and the remaining characters are processed as Style II command characters.
- If the address character is the current Plotter address, but the command code is not recognized, an unrecognized command error is generated. Again, arguments are discarded until the command is terminated.
- Errors in argument specifications (which cause a delimited string argument to be recognized when a numeric argument was expected) cause an argument type error to be generated and the command to be discarded.
- A numeric argument is interpreted as an undelimited string argument if a string argument was expected.
- An undelimited string argument (with the exception of numbers) is interpreted as a command terminator if a numeric argument was expected.
- If a command is entered in error, a new command can be started anywhere if started with a new attention character. The erroneous portion entered is discarded or executed, depending upon the characters already entered and the errors previously described.

STYLE II COMMANDS (4010 SERIES TERMINAL COMMANDS)

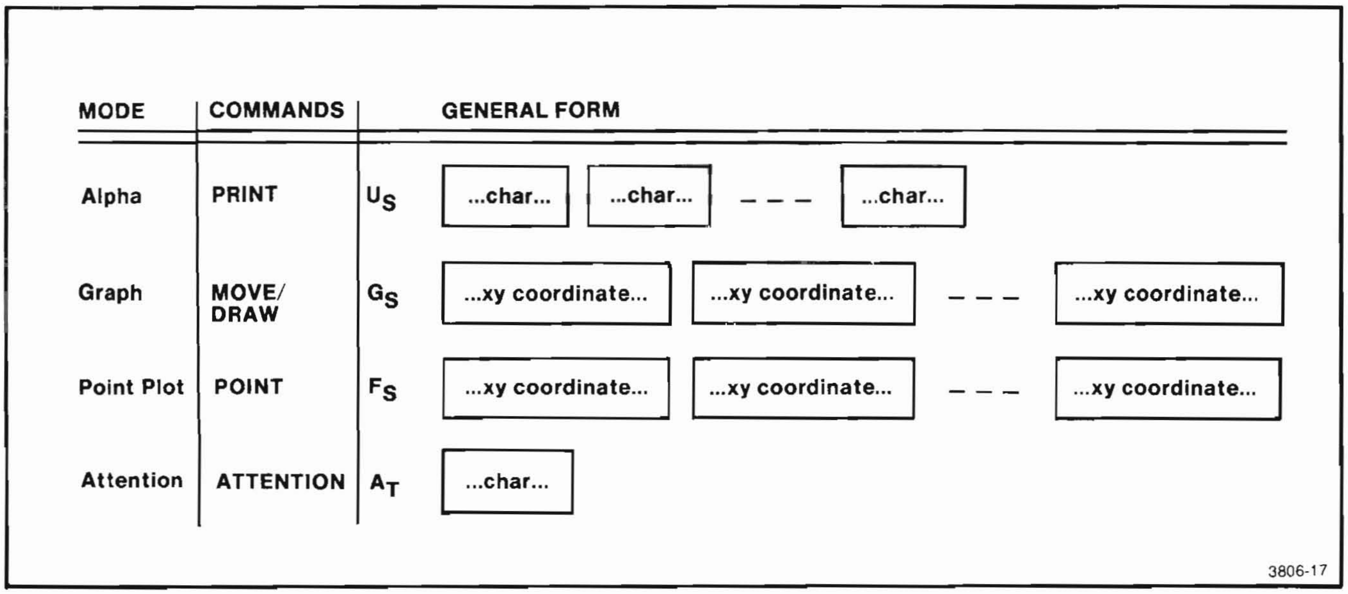
The Style II commands recognized by the Plotter are very similar to the commands that control TEKTRONIX 4010 Series Terminals. For most standard functions, the Plotter responds the same as a TEKTRONIX 4014 Terminal equipped with Option 34 (Extended Graphics Module). Refer to Appendix I for more information.

Style II commands permit plots developed at a 4010-Series Terminal to be sent to the Plotter for a high quality plot. Style II commands are highly coded for transmission efficiency and are more difficult to generate manually than Style I commands. However, these commands are normally generated by host computer support software.

The four basic Style II commands are shown in Figure 4-11. Each command controls some basic Plotter action, such as printing characters, moving the pen, drawing lines, arcs, and circles with the pen, and marking points at specific coordinates. The Attention Action commands control special functions or establish parameters for print or plot actions. Each of these commands (with their functions and parameters) are discussed in the following text.

These commands are expressed to the Plotter by first sending a mode control character (such as, U_S, G_S, F_S, or A_T) to place the Plotter interface into the proper mode. The relationship between these modes is defined by the mode transition diagram shown in Figure 4-12.

The mode control characters are similar to command codes. The modes "remember" that the corresponding commands are being processed. Thus, the mode control characters do not need repeating for each successive command unless the mode is changed. While the Plotter interface is in the desired mode, successive characters received are interpreted as arguments for the corresponding commands. The required coding for these arguments is described for each of the modes.



MODE	COMMANDS	GENERAL FORM				
Alpha	PRINT	U_S	...char...	...char...	— — —	...char...
Graph	MOVE/ DRAW	G_S	...xy coordinate...	...xy coordinate...	— — —	...xy coordinate...
Point Plot	POINT	F_S	...xy coordinate...	...xy coordinate...	— — —	...xy coordinate...
Attention	ATTENTION	A_T	...char...			

Figure 4-11. Basic Style II Commands.

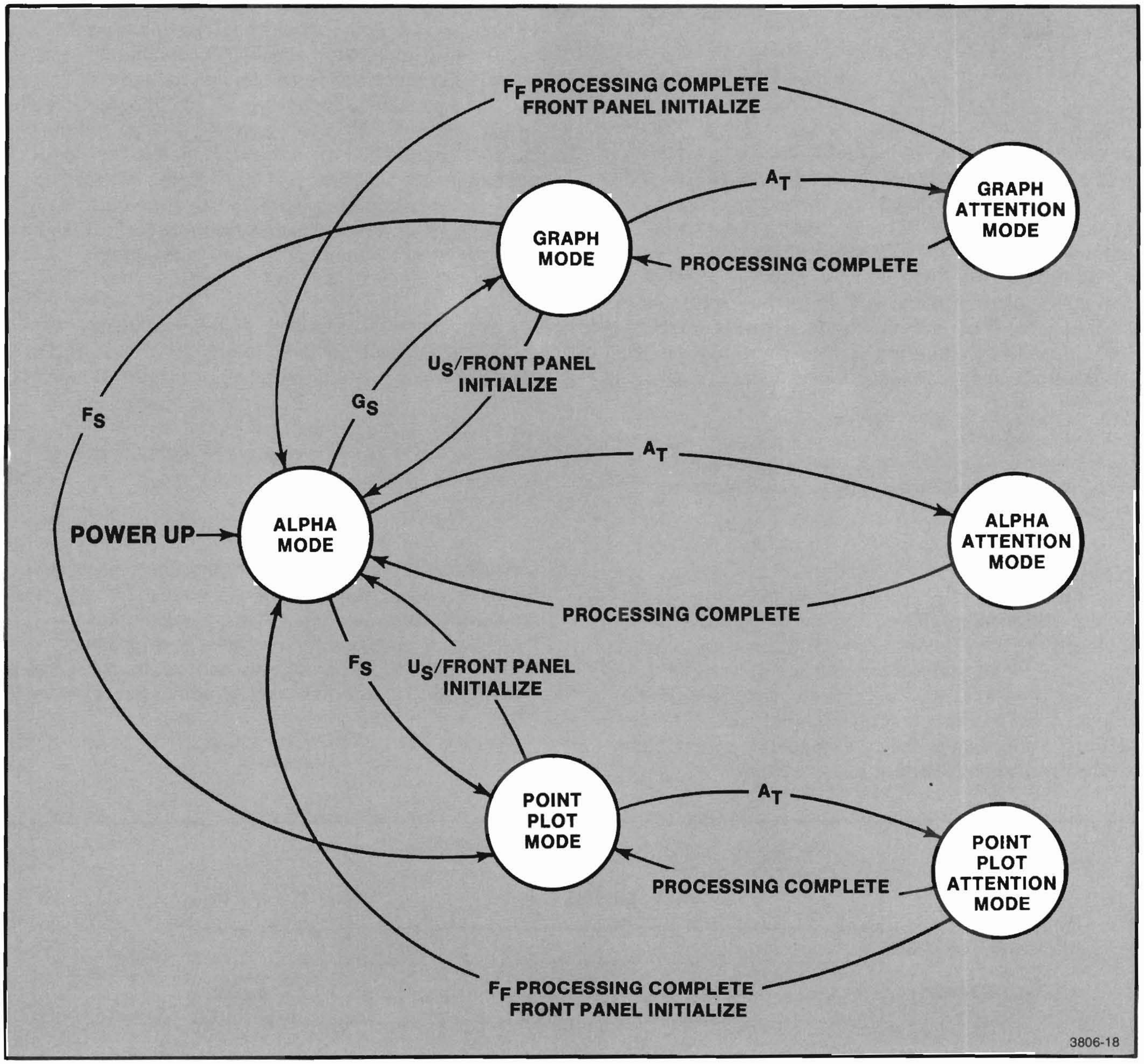


Figure 4-12. Style II Command Modes.

Alpha Mode

To print characters using Style II commands, send a $^{U}_{S}$ character to the Plotter to force Alpha mode. The Plotter automatically resets to Alpha mode at power-up or when a front panel Initialize function is performed. In Alpha mode, subsequent characters sent to the Plotter are normally printed beginning at the current pen position. Refer to the Alpha Commands section of this manual for a detailed description of the print actions for each ASCII character. (The characters $^{F}_{F}$, $^{S}_{O}$, and $^{S}_{I}$ and the mode control characters cannot be printed in Alpha mode.) Also, several Attention Action commands affect Alpha mode printing actions. Refer to *Attention Action Mode* later in this section for more information.

NOTE

A solid line type is always used when alpha characters are drawn regardless of the current graphic line type.

Graph Mode

To draw lines using Style II commands, send a $^{G}_{S}$ character to the Plotter to force Graph mode. In Graph mode, the Plotter can perform Moves, Draws, Arcs, and Circles. Move and Draw actions and the methods that specify their arguments (consisting of encoded pairs of X and Y coordinates) are described next. Arcs and Circles are described later in this section.

The first X,Y coordinate pair received following the $^{G}_{S}$ character is interpreted as a MOVE command coordinate, which causes a pen up motion to the specified position. Successive X,Y coordinate pairs are interpreted as DRAW command coordinates, which cause a straight line to be drawn from the current position to the specified positions. When a Draw, Arc, or Circle is specified, it is drawn using the current graphic line type (solid, dashed, dotted, etc.). This line type can be set with either an Attention Action command or a SET LINE TYPE command.

A Move can follow a Draw by preceeding the MOVE command's coordinate with another $^{G}_{S}$ character. The Move following the $^{G}_{S}$ can be suppressed by either: (1) following the $^{G}_{S}$ with the current coordinates (which generates a Move to the current position), or (2) following the $^{G}_{S}$ character with a $^{B}_{L}$ (bell) character (which changes the following Move to a Draw and rings the bell).

Coordinate Encoding. The X,Y coordinate pairs sent to the Plotter as MOVE or DRAW command coordinates are described as two 16-bit binary coordinates in Figure 4-13. Thus, each coordinate has 12 bits of integer value and 4 bits of fractional value. This gives a 0 to 4096.9375 coordinate range where the units are in ADU's.

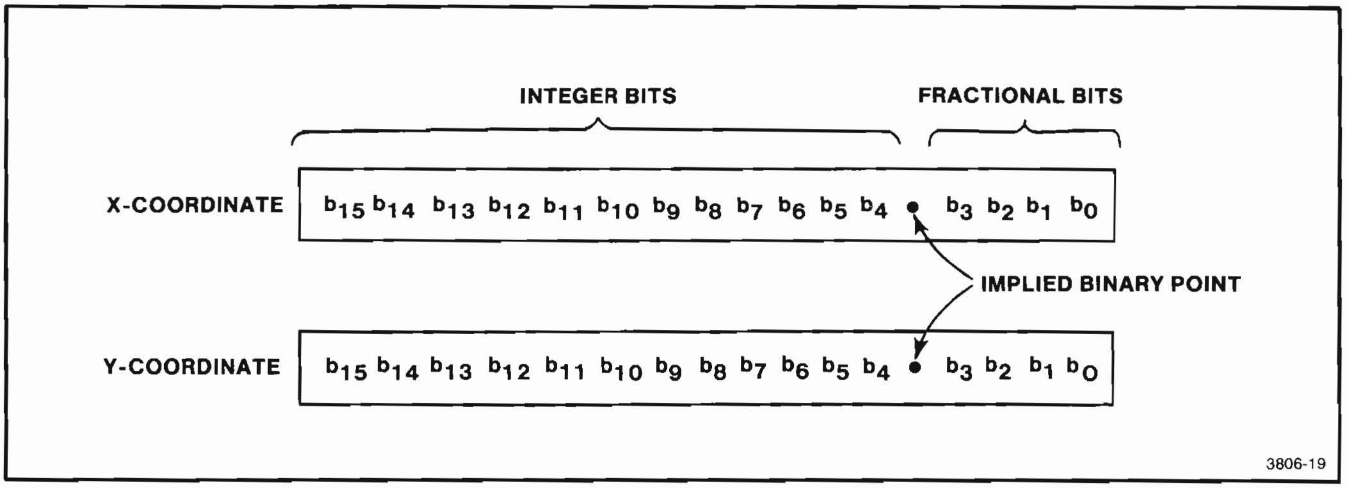

INTEGER BITS
FRACTIONAL BITS
X-COORDINATE
b15 b14 b13 b12 b11 b10 b9 b8 b7 b6 b5 b4 • b3 b2 b1 b0
IMPLIED BINARY POINT
Y-COORDINATE
b15 b14 b13 b12 b11 b10 b9 b8 b7 b6 b5 b4 • b3 b2 b1 b0
3806-19

Figure 4-13. 16-Bit Binary Coordinates.

NOTE

The X,Y coordinates specified in Style II commands are always interpreted as World Units with ADUs as the current Device Units regardless of the graphic units selected in either the SELECT GRAPHIC UNITS or SELECT DEVICE UNITS command. When using Style II commands, the current graphic units should be World Units, and the current Device Units should be ADU's.

Figure 4-14 defines five groups of binary bits within each 16-bit binary description (for example, HIX, LOX, EB, EEB, and EEEB for the X-coordinate). The bit patterns that appear in these groups are mapped into portions of the ASCII character set as shown in Figure 4-15. The five least significant bits of the 7-bit ASCII characters are matched with the corresponding bit patterns.

The other two bits of the 7-bit ASCII characters identify one of the seven coding groups. The appropriate ASCII characters are then used to express the encoded coordinates. The HIX and HIY groups are assigned to the same portion of the ASCII character set and identified by the order in which the Plotter receives them. The same holds true for the extra bytes (EB, EEB, and EEEB) that are assigned to the same portion of the LOY byte.

NOTE

The correct identification of the encoded coordinate characters depends upon the order in which the characters are received. If a sequence of coordinate characters is interrupted while being received (before the LOX character is received), a following sequence must begin with another GS character to mark the beginning of a sequence.

NOTE

This coding scheme requires that the host computer be able to transmit all of the characters in the last six columns of Figure 4-15, plus the current attention character.

Some host computers use the DL character for system functions. In this case, an AT **?** Attention Action command can be substituted for the DL character during encoding of the X,Y coordinate pair. Refer to *Attention Mode* further in this section for more information.

A detailed definition of the bytes required to encode the X,Y coordinate pair, including the order in which they are transmitted to the Plotter, is defined in Table 4-3.

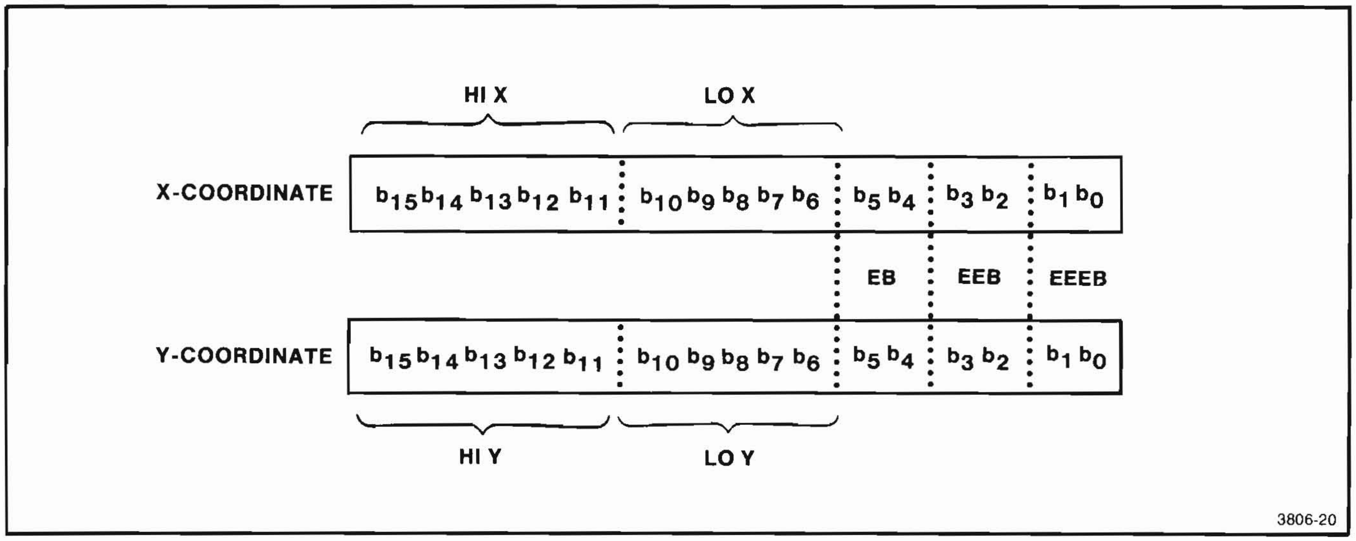


Figure 4-14. Definition of Graphic Coding Groups.

B7 B6 B5	Ø Ø Ø	Ø Ø 1	Ø 1 Ø	Ø 1 1	1 Ø Ø	1 Ø 1	1 1 Ø	1 1 1
BITS	CONTROL		HI X HI Y		LO X		LO Y EB EEB EEEB	
B4 B3 B2 B1								
Ø Ø Ø Ø	0 NU 0 NUL 0	20 DL 10 DLE 16	40 Sp 20 32	60 0 30 48	100 @ 40 64	120 P 50 80	140 ` 60 96	160 p 70 112
Ø Ø Ø 1	1 SH 1 SOH 1	21 D1 11 DC1 17	41 ! 21 33	61 1 31 49	101 A 41 65	121 Q 51 81	141 a 61 97	161 q 71 113
Ø Ø 1 Ø	2 SX 2 STX 2	22 D2 12 DC2 18	42 " 22 34	62 2 32 50	102 B 42 66	122 R 52 82	142 b 62 98	162 r 72 114
Ø Ø 1 1	3 EX 3 ETX 3	23 D3 13 DC3 19	43 # 23 35	63 3 33 51	103 C 43 67	123 S 53 83	143 c 63 99	163 s 73 115
Ø 1 Ø Ø	4 ET 4 EOT 4	24 D4 14 DC4 20	44 $ 24 36	64 4 34 52	104 D 44 68	124 T 54 84	144 d 64 100	164 t 74 116
Ø 1 Ø 1	5 EQ 5 ENQ 5	25 NK 15 NAK 21	45 % 25 37	65 5 35 53	105 E 45 69	125 U 55 85	145 e 65 101	165 u 75 117
Ø 1 1 Ø	6 AK 6 ACK 6	26 SY 16 SYN 22	46 & 26 38	66 6 36 54	106 F 46 70	126 V 56 86	146 f 66 102	166 v 76 118
Ø 1 1 1	7 BL 7 BEL 7	27 EB 17 ETB 23	47 ' 27 39	67 7 37 55	107 G 47 71	127 W 57 87	147 g 67 103	167 w 77 119
1 Ø Ø Ø	10 BS 8 8	30 CN 18 CAN 24	50 (28 40	70 8 38 56	110 H 48 72	130 X 58 88	150 h 68 104	170 x 78 120
1 Ø Ø 1	11 HT 9 9	31 EM 19 25	51) 29 41	71 9 39 57	111 I 49 73	131 Y 59 89	151 i 69 105	171 y 79 121
1 Ø 1 Ø	12 LF A 10	32 SB 1A SUB 26	52 * 2A 42	72 : 3A 58	112 J 4A 74	132 Z 5A 90	152 j 6A 106	172 z 7A 122
1 Ø 1 1	13 VT B 11	33 EC 1B ESC 27	53 + 2B 43	73 ; 3B 59	113 K 4B 75	133 [5B 91	153 k 6B 107	173 { 7B 123
1 1 Ø Ø	14 FF C 12	34 FS 1C 28	54 , 2C 44	74 < 3C 60	114 L 4C 76	134 \ 5C 92	154 l 6C 108	174 |* 7C 124
1 1 Ø 1	15 CR D 13	35 GS 1D 29	55 – 2D 45	75 = 3D 61	115 M 4D 77	135] 5D 93	155 m 6D 109	175 } 7D 125
1 1 1 Ø	16 SO E 14	36 RS 1E 30	56 . 2E 46	76 > 3E 62	116 N 4E 78	136 ^ 5E 4	156 n 6E 110	176 ~ 7E 126
1 1 1 1	17 SI F 15	37 US 1F 31	57 / 2F 47	77 UNL ? 3F 63	117 O 4F 79	137 UNT — 5F 95	157 o 6F 111	177 DT DEL RUBOUT 7F 127

*¦ on some keyboards or systems

KEY

octal	25	
	NK	graphic representation
hex	15 NAK 21	decimal
	mnemonic	

(2670) 3806-21

Figure 4-15. ASCII Code Chart Showing Assignment of Graphic Coding Groups.

Table 4-3
PACKED BINARY COORDINATE CODING DEFINITION

		7-Bit ASCII Character						
Byte Number	**Byte Name**	**Fixed Bits**		**Data Bits**				
		B_7	B_6	B_5	B_4	B_3	B_2	B_1
1	High Order Y (HIY)	0	1	YI5	Y14	Y13	Y12	Y11
				5 most significant bits of Y				
2	Extra 3 Byte (EEEB)	1	1	*	Y1	Y0	X1	X0
				Unused	2 lowest bits of Y		2 lowest bits of X	
3	Extra 2 Byte (EEB)	1		*	Y3	Y2	X3	X2
				Unused	2 lower bits of Y		2 lower bits of X	
4	Extra 1 Byte (EB)	1	1	*	Y5	Y4	X5	X4
				Unused	2 low bits of Y		2 low bits of X	
5	Low Order Y (LOY)	1	1	Y10	Y9	Y8	Y7	Y6
				5 intermediate bits of Y				
6	High Order X (HIX)	0	1	X15	X14	X13	X12	X11
				5 most significant bits of X				
7	Low Order X (LOX)	1	0	X10	X9	X8	X7	X6
				5 intermediate bits of X				

*Designates a "don't care" bit position.

This coding results in the following graphic character sequence, which specifies a 16-bit XY coordinate pair:

Plotter ← HIY EEEB EEB EB LOY HIX LOX ← HOST

These characters must appear in the defined order to be correctly interpreted. Appendix F contains two types of tables that you can use to determine the coded bytes required to define a given X,Y coordinate pair.

Coordinate Resolution. When the Plotter operates at less than maximum resolution, all 16 bits do not need to be transmitted to the Plotter, which results in greater transmission efficiency. Different resolutions of 10, 12, 14, and 16 bits can be tranmitted for the X,Y coordinates by simply eliminating the appropriate number of extra bytes. These optional bytes permit coordinate resolutions as shown in Table 4-4.

Table 4-4

COORDINATE RESOLUTION

Coordinate Bits	Resolution (ADUs)	Extra Bytes Required
10	4	none
12	1	EB
14	.25	EEB, EB
16	.0625	EEEB, EEB, EB

Optimized Coordinate Encoding. Further transmission efficiency can be achieved by optimizing coordinate transmission. When the Plotter receives coordinate specifications, the graphic bytes update two 16-bit graphic memory coordinates (which always reflect the current graphic position). The current values of the graphic memory coordinates are then used to form the internal arguments for the MOVE or DRAW commands (and the ARC, CIRCLE and POINT commands described in following sections). Thus, it is necessary to transmit only the portion of these coordinates that changes from the last Move or Draw. Table 4-5 defines the minimum number of bytes that must be used to transmit the coordinate bits.

Table 4-5

INSTRUCTIONS FOR OPTIMIZED BINARY COORDINATE CODING

Bytes Changed	Bytes Sent						
	HIY	EEEB	EEB	EB	LOY	HIX	LOX
HIY	X						X
EEEB		X	X	X	X		X
EEB			X	X	X		X
EB				X	X		X
LOY					X		X
HIX					X	X	X
LOX							X

NOTE

If the specification of an X,Y coordinate is interrupted and then restarted, the graphic memory is modified even if no LOX character is received to initiate the corresponding Move or Draw action. Therefore, to ensure that the graphic memory contains the desired values, the restarted graphic sequence should be fully specified and not optimized.

NOTE

The graphic memory in the Plotter is only reset to 0 during power-up or when a front panel Initialize function is performed. Therefore, a plot should normally begin with a full 16-bit coordinate specification that sets all coordinate bits to known values even if less resolution is used for the remainder of the plot.

Arcs and Circles. You can plot arcs and circles while in Graph mode using binary coded arguments (requires Option 31). To plot a circle using Style II commands, an $^{S}_{O}$ character is sent to the Plotter immediately followed by a coded X,Y coordinate pair. The X value specifies the radius of the circle to be drawn around the current pen position. The Y value is unused and the pen is left at the circle center.

NOTE

The specification of the circle radius modifies the graphic memory coordinates in the Plotter. Thus, the next graphic coordinate specification should be sent to the Plotter as a complete specification and should not be optimized.

Enter Graph mode to plot an arc using Style II commands. Then an $^{S}_{I}$ character is sent to the Plotter immediately followed by a coded X,Y coordinate defining the midpoint of the arc, which in turn is followed by another coded X,Y coordinate defining the end point of the arc. This causes an arc to be plotted from the current position through the specified midpoint to the end point.

Both DRAW CIRCLE and DRAW ARC commands require that the Plotter be equipped with 4663 Option 3l (Arcs, Circles, and Programmable Macros). Refer to the Graphic Commands section for more details concerning these functions.

If Option 3l is not present, the DRAW CIRCLE command is ignored and the DRAW ARC command causes Draws to the midpoint and then to the endpoint of the specified arc.

Point Plot Mode

To locate a dot at specified coordinates without performing Move or Draw operations, send an $^{F}_{S}$ character to the Plotter to force Point Plot mode. In this mode, the current graphic line type is saved. Then the graphic line type is changed to *point*, which causes a dot to be placed at each specified coordinate. This action is not affected by Move or Draw interpretations of coordinates.

Point Plot mode is exited by sending a $^{U}_{S}$ character or a $^{A}_{T}$ $^{F}_{F}$ attention sequence to the Plotter, forcing the mode to Alpha. When Graph mode is entered following Point Plot actions, the graphic line type (solid, dashed, etc.) that was previously in effect for Graph mode is restored. If an Attention Action command that changes the graphic line type is received while in Point Plot mode, the new specified line type does not take effect until Graph mode is entered again.

NOTE

Only graphic line types specified by Attention Action commands (and not via the SET LINE TYPE command) are restored when Graph mode is re-entered following Point Plot actions. Also, if Point Plot mode is exited via an $^{A}_{T}$ $^{F}_{F}$ Attention Action command, the graphic line type is forced to solid. For more information, refer to Attention Action Mode described next.

Attention Action Mode

The Attention Action commands allow special functions to be performed or parameters to be established without changing the basic command mode.

NOTE

Referring back to the Figure 4-12 mode diagram, there are three Attention Action modes: Alpha Attention, Graph Attention, and Point Attention. Unless the Attention Action command causes a mode change, the current mode (Alpha, Graph, or Point) will be reestablished after the Attention Action command process is completed.

Unless noted otherwise in the following discussion, the Attention Action commands perform the same function in all three modes (Alpha, Graph, or Point). The Attention Action command can appear at any point within the character sequences that are used to form commands for the specific mode. Also, unless noted, the Attention Action commands are always two characters long: the current AT character (specified in the Paramenter Entry Card) and a specified action character.

Tables 4-6 through 4-8 list the Attention Action commands and describe the resultant Plotter action. The use of any ASCII characters not listed in these tables are ignored.

NOTE

Each Attention Action command must be completed before another is specified. Two successive AT characters are equal to one AT character. Any following characters are ignored if they appear between an AT character and the second character of an Attention Action command (LF, CR, or NU).

Table 4-6

ATTENTION ACTION COMMANDS AND THEIR EFFECTS

Attention Action Command	Action
AT?	Is converted to a DL character and interpreted as a LOY graphic byte.
ATBL	Bell sounds. Can be used to ring bell without affecting the mode.
ATGS	Equivalent to GS
ATUS	Equivalent to US
ATFF	Forces the Plotter to Alpha mode, forces the graphics line type to solid, and performs the Move-To-Home function. Can be used to establish a known state for these parameters. Equivalent to a Style/I Print FF command[a].
ATSO	Activates the Alternate Alpha character set. Equivalent to a Style I Print SO command[a].
ATSI	Activates the Standard Alpha character set. Equivalent to a Style I Print SI command[a].

[a] **Refer to the Alpha Commands section for more details about these actions.**

Table 4-7 specifies the Attention Action commands that establish alpha character size values equivalent to the TEKTRONIX 4014 Series large, #2, #3, and small character sizes. Refer to the Alpha Commands section for more information on alpha parameters.

Table 4-7

SPECIFIERS FOR ALPHA CHARACTER SIZE

Attention Action Command	Character Width(ADUs)[a]	Character Height(ADUs)[a]	Corresponding Character Space(ADUs)	Corresponding Line Space(ADUs)
^{A}T 8	37.33	53.78	56	88
^{A}T 9	34.00	50.11	51	82
^{A}T ;	22.67	32.39	34	53
^{A}T :	20.67	29.33	31	48

[a] Assuming the default Alpha X-Ratio and Y-Ratio values of 3/2 (1.5) and 18/11 (1.64) respectively.

Table 4-8 specifies the Attention Action commands that establish line types, dash patterns, and dash pattern lengths that are equivalent to the TEKTRONIX 4010 Extended Graphics Module (EGM) line types and dash patterns. Also refer to Section 11 for more information about line types and dash patterns.

NOTE

For the Plotter, 4014 Extended Graphics Module (EGM) defocused and write-through vector line types are drawn the same as the corresponding normal line types.

Table 4-8

SPECIFIERS FOR LINE TYPES AND DASH PATTERNS

Attention Action Command	Line Type	Dash Pattern[a]	Dash Pattern Length (ADUs)
^{A}T '	Solid	N/A	N/A
^{A}T h	Solid	N/A	N/A
^{A}T p	Solid	N/A	N/A
^{A}T e	Solid	N/A	N/A
^{A}T m	Solid	N/A	N/A
^{A}T u	Solid	N/A	N/A
^{A}T f	Solid	N/A	N/A
^{A}T n	Solid	N/A	N/A
^{A}T v	Solid	N/A	N/A
^{A}T g	Solid	N/A	N/A
^{A}T o	Solid	N/A	N/A
^{A}T w	Solid	N/A	N/A
^{A}T a	dotted	1,1 (dot,space)	11
^{A}T i	dotted	1,1 (dot,space)	11
^{A}T q	dotted	1,1 (dot,space)	11
^{A}T b	dot-dash	5,1,1,1 (dash,space, dot,space)	88
^{A}T j	dot-dash	5,1,1,1 (dash,space, dot,space)	88
^{A}T r	dot-dash	5,1,1,1 (dash,space, dot,space)	88
^{A}T c	short dash	3,1 (dash,space)	44
^{A}T k	short dash	3,1 (dash,space)	44
^{A}T s	short dash	3,1 (dash,space)	44
^{A}T d	long dash	6,2 (dash,space)	176
^{A}T l	long dash	6,2 (dash,space)	176
^{A}T t	long dash	6,2 (dash,space)	176

[a] The number indicates the number of dashes, spaces, or dots the Plotter draws. For example, ^{A}T b plots 5 dashes, 1 space, 1 dot, and 1 space (— — — — — ● — — — — — ●).

MIXING STYLE I AND STYLE II COMMANDS

All Plotter functions can be controlled by Style I commands when the interface is in any Style II command mode (Alpha, Graph, Point, and Attention Action). In addition, many Plotter functions have a Style II equivalent command (where the capabilities of the Plotter overlap the capabilities of TEKTRONIX 4010 Series terminals). Either command style can be used when equivalent commands are available. Style I commands can be viewed as a large set of multicharacter Attention Action commands. Thus, Style I commands can appear within Style II commands anyplace an Attention Action command is appropriate.

When only Style I commands are used, the Serial interface is always in Alpha mode (because the Plotter powers up in that mode). For this reason, be sure you correctly terminate Style I commands; any stray characters following these commands are interpreted as PRINT command arguments and are printed.

OUTPUT RESPONSES

Output messages (consisting of DC1/DC3 control characters), control blocks, and response blocks are transmitted from the Plotter to the host in response to output-generating Plotter commands. The Plotter encodes these messages according to the Initial Command/Response Format in effect. The following paragraphs explain how the host decodes each type of response to retrieve the desired information. The DC1/DC3 control character responses require no decoding. Refer to *Plotter-To-Device Transmission* earlier in this section for more information on how these blocks are actually transmitted to the host.

Control Block

Use control blocks for Plotter functions where no data values are required. The only control block responses transmitted by the Plotter are the negative and positive Block Acknowledge characters (I and A) used in Block mode communications (see *Host-To-Plotter Transmission* earlier in this section). These blocks are always one character in length and are encoded as shown in Figure 4-16, regardless of the selection of the Initial Command/Response Format line of the Parameter Entry Card.

BIT POSITION	b_8	b_7	b_6	b_5	b_4	b_3	b_2	b_1
BIT VALUE	P	1	0	C0	C1	0	0	1

WHERE: P = PARITY BIT
C0 AND C1 = CONTROL BITS

3806-22

Figure 4-16. Control Block Coding.

The control bits (C0 and C1) are encoded as either "0 0" (a positive Block Acknowledge giving the ASCII character "A") or as "0 1" (a negative Block Acknowledge giving the ASCII character "I"). The "0 0 1" value for Bits 3 through 1 is the control block TAG value. The "1 0" value for Bits 7 and 6 indicate that this is the last byte of a binary coded block.

Response Block

A response block is used by the Plotter when responding to all other output-generating commands. The response block, which can range from 7 to 33 characters in length (depending upon the data values and the current Command/Response Format selection), consists of two data values along with a TAG value. The two data values are typically obtained from the encoding of several pieces of requested information. The information can then be decoded from the data values by reversing the encoding process. Refer to the specific output-generating command for more information.

For Formats 1 and 2 (the standard Plotter formats), the interpretation of TAG values is explained for each output-generating command in the command description sections. Since the TAG values have no fixed meaning, these formats are called *unlabeled.* Correct interpretation of the Plotter output responses requires the host to keep track of the order in which requests were made and to match up returned data with the requests. The Plotter returns data in the same order you request it.

For Formats 3 and 4 (4662 compatability formats), the TAG bits have preassigned values (which allow the host to match up responses with requests by looking at the TAG values that act as labels). However, even in Formats 3 or 4, the responses are returned in the same order they were requested.

Following is a description of the data blocks for each Command/Response Format. This description can be used along with the description of the output data encoded for a particular command (see Section 7) to determine the actual response bytes transmitted.

Format 1 — Unlabeled Binary Output. The packed-binary format consists of seven consecutive ASCII characters. These are encoded as shown in Figure 4-17 from two 16-bit numbers and three bits of TAG information. The coding (defined more formally in Figure 4-18) uses only the ASCII characters "Space" to "Underline" (ADE 32 to ADE 95) to permit return of responses to the host even if the host computer's input character set is restricted (such as when lowercase characters are not accepted).

For all responses except coordinate values, the 16-bit values are integers with a numeric range of 0 - 65535. For coordinate values, a binary point is assumed between Bits 3 and 4 giving a numerical coordinate range of 0 — 4095.9375. Digitized coordinates greater than 4095.9375 cannot be encoded in this format and thus, initiates an output value range error. The coordinate range to be digitized should either be restricted or Format 2 must be used. For Command/Response Formats 1 and 2, the digitized coordinate resolution is 16 bits. For Command/Response Formats 3 and 4, the digitized coordinate resolution is 12 bits, giving a numerical coordinate range of 0 — 4095.

NOTE

If the coordinate value to be transmitted has a value of 4096, it will be sent as 4095.9375 (for 16-bit resolution) or as 4095 (for 12-bit resolution) with no error indicated.

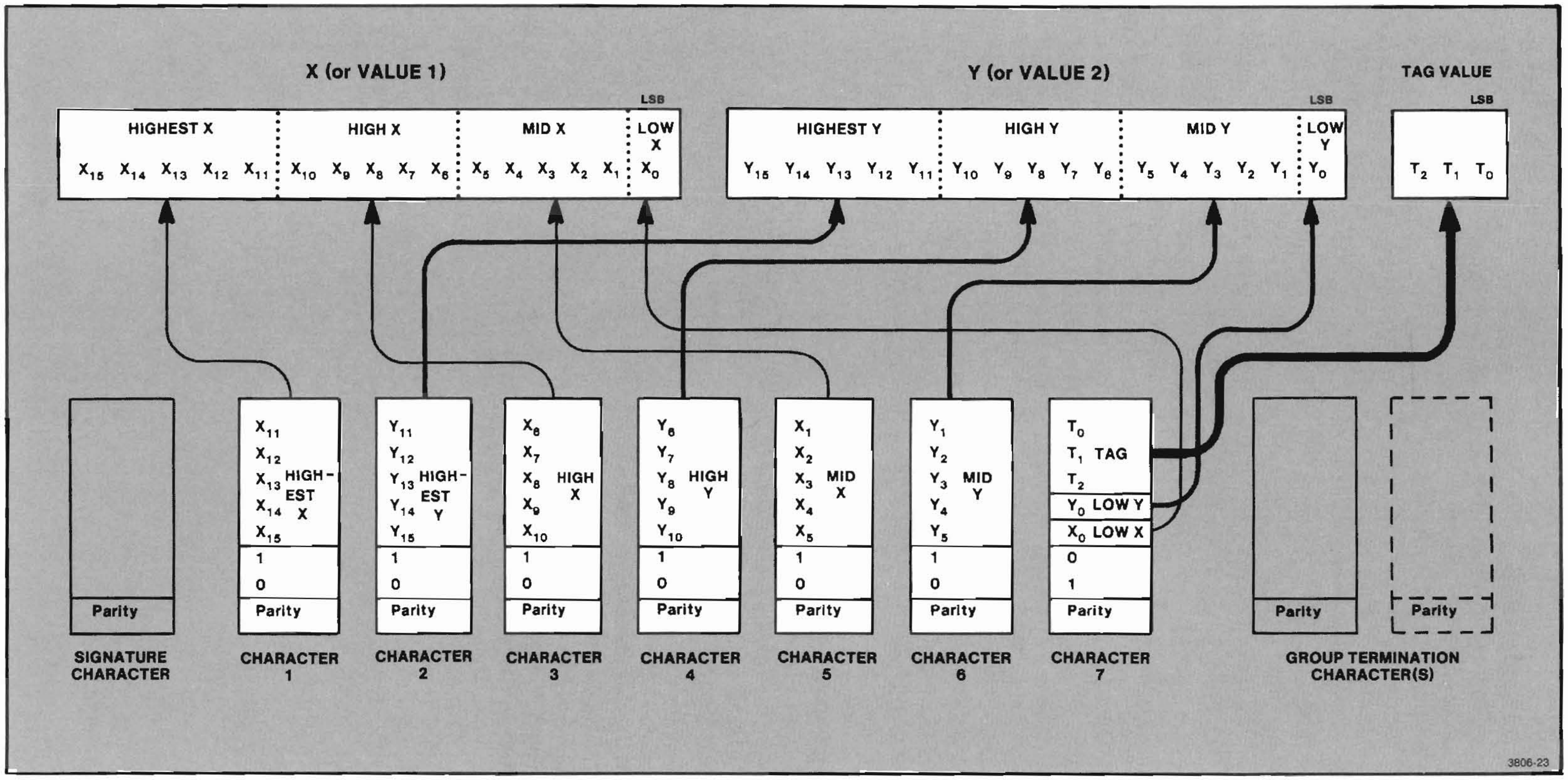

X (or VALUE 1)
LSB
HIGHEST X
HIGH X
MID X
LOW X
X_{15} X_{14} X_{13} X_{12} X_{11}
X_{10} X_9 X_8 X_7 X_6
X_5 X_4 X_3 X_2 X_1
X_0
Y (or VALUE 2)
LSB
HIGHEST Y
HIGH Y
MID Y
LOW Y
Y_{15} Y_{14} Y_{13} Y_{12} Y_{11}
Y_{10} Y_9 Y_8 Y_7 Y_6
Y_5 Y_4 Y_3 Y_2 Y_1
Y_0
TAG VALUE
LSB
T_2 T_1 T_0
Parity
SIGNATURE CHARACTER
X_{11} X_{12} X_{13} X_{14} X_{15}
HIGH-EST X
1
0
Parity
CHARACTER 1
Y_{11} Y_{12} Y_{13} Y_{14} Y_{15}
HIGH-EST Y
1
0
Parity
CHARACTER 2
X_6 X_7 X_8 X_9 X_{10}
HIGH X
1
0
Parity
CHARACTER 3
Y_6 Y_7 Y_8 Y_9 Y_{10}
HIGH Y
1
0
Parity
CHARACTER 4
X_1 X_2 X_3 X_4 X_5
MID X
1
0
Parity
CHARACTER 5
Y_1 Y_2 Y_3 Y_4 Y_5
MID Y
1
0
Parity
CHARACTER 6
T_0 T_1 T_2
TAG
Y_0 LOW Y
X_0 LOW X
0
1
Parity
CHARACTER 7
Parity
Parity
GROUP TERMINATION CHARACTER(S)
3806-23

Figure 4-17. Packed Binary Response Block Format.

CHARACTER NUMBER	CHARACTER NAME	7-BIT ASCII CHARACTER						
		FIXED BITS		DATA BITS				
		B_7	B_6	B_5	B_4	B_3	B_2	B_1
1	HIGHEST X	0	1	X_{15}	X_{14}	X_{13}	X_{12}	X_{11}
				5 MSB OF DATA VALUE 1				
2	HIGHEST Y	0	1	Y_{15}	Y_{14}	Y_{13}	Y_{12}	Y_{11}
				5 MSB OF DATA VALUE 2				
3	HIGH X	0	1	X_{10}	X_9	X_8	X_7	X_6
				5 INTERMEDIATE BITS OF DATA VALUE 1				
4	HIGH Y	0	1	Y_{10}	Y_9	Y_8	Y_7	Y_6
				5 INTERMEDIATE BITS OF DATA VALUE 2				
5	MID X	0	1	X_5	X_4	X_3	X_2	X_1
				NEXT 5 INTERMEDIATE BITS OF DATA VALUE 1				
6	MID Y	0	1	Y_5	Y_4	Y_3	Y_2	Y_1
				NEXT 5 INTERMEDIATE BITS OF DATA VALUE 2				
7	LOX, LOY, TAG	1	0	X_0	Y_0	T_2	T_1	T_0
						TAG VALUE		

LSB OF DATA VALUE 1

LSB OF DATA VALUE 2

3806-24

Figure 4-18. Packed Binary Response Format.

For Format 1, TAG bits may have the numeric values shown in Table 4-9, depending upon the response being transmitted.

Table 4-9

TAG VALUES FOR FORMAT 1

TAG Value (Decimal)	TAG Value (Binary)	Meaning
0	000	Not used
1	001	Control Block (see Control Block description)
2	010	Not Last Block
3	011	Not Last Block
4	100	Last Block
5	101	Last Block
6	110	Not Used
7	111	Not Used

The last block designation allows command responses that consist of a variable number of blocks to be terminated.

Format 2 — Unlabeled ASCII Output. This ASCII format consists of two data values and the TAG value, all expressed as ASCII decimal numbers separated by commas, as shown below:

Number,	Number,	Number
Value 1	Value 2	TAG Value

The format of the numbers depends upon the current output graphic units. If World Units are being used, the numbers (including the TAG value) are expressed in the following scientific notation form:

— X.XXXXXE± XX
or
ˢᴾ X.XXXXXE± XX

The sign of the mantissa is a space character if the number is positive and a minus sign if the number is negative. The mantissa accuracy can be up to five digits to the right of the decimal point. The exponent always contains two digits and is always preceeded by either a "+ " or a "—" sign.

If current output graphic units are Device Units (ADUs, GDUs, or millimeters), the output numbers (including the TAG value) will be in fixed point format with a fixed width shown below:

— XXXXX.XXX
or
ˢᴾ XXXXX.XXX

The sign is a ˢᴾ character if the number is positive and a "—" if the number is negative. Leading zeros and/or trailing spaces are generated to maintain both the field width (ten characters) and the location of the decimal point in the field.

For Format 2, the TAG number may have the values shown in Table 4-10, depending upon the response being transmitted.

Table 4-10

TAG VALUES FOR FORMAT 2

TAG Value (Decimal)	Meaning
0	Not Last Block
1	Not Last Block
2	Last Block
3	Last Block

The last block designation allows command responses that consist of a variable number of blocks to be terminated.

Format 3 — Labeled Binary Output. This packed-binary output format is the same as Format 1, with the exception of TAG value interpretations, which are shown in Table 4-11.

Table 4-11

TAG VALUES FOR FORMAT 3

TAG Value (Decimal)	TAG Value (Binary)	Meaning
0	000	Digitized Block — Pen Up
1	001	Control Block
2	010	Status Block
3	011	Size Block
4	100	Digitized Block — Pen Down
5	101	Not used
6	110	Not Used
7	111	Not Used

For this format, the type of response data (Digitized, Status, etc.) can be determined by referring to the TAG values. The indicated response types are the only ones that can be requested in this format (all defined responses can be transmitted when using Formats 1 or 2). For digitized responses (using Formats 3 and 4), the coordinate value Bits 3 through 0 of both data values are set to 0. This results in 12-bit resolution for digitized graphics.

Format 4 — Restricted Labeled Binary Output. This format is the same as Format 3 with one exception: Bit 0 of the X-coordinate value (for digitize responses) is always set to 1 to avoid encoding an "@" character in the output. Some host computers use the "@" character for special functions.

Section 5

GPIB INTERFACE

INTRODUCTION

This section describes five aspects of the GPIB interface:

- Connecting the Plotter to other GPIB devices.
- Making selections on the Parameter Entry Card that allow other GPIB devices to communicate with the Plotter.
- Controlling transmission of information to and from the Plotter.
- Understanding the various ways in which Plotter commands are formated.
- Decoding Plotter responses.

CONNECTING THE PLOTTER TO OTHER DEVICES

The General Purpose Interface Bus (GPIB) is a standardized interfacing system defined by IEEE Standard 488-1978 for programmable measuring apparatus. This standard defines the mechanical, electrical, and procedural (such as, how to send data bytes to and from the interface) characteristics of the bus. Standardization allows the Plotter to be easily installed in various systems with devices that communicate through the GPIB. However, GPIB information (consisting of commands and responses that are transfered over the GPIB bus) is not defined in the standard, and is unique to the Plotter. This subject is discussed further under *Introduction to GPIB Commands and Responses* in this section.

NOTE

The Plotter is designed to work with other devices that conform to IEEE Standard 488-1978. However, the Plotter's GPIB interface does exhibit some minor limitations in comparison to the standard GPIB. These limitations are discussed in Appendix E.

The GPIB allows devices to be interconnected in any convenient configuration if the following limitations are observed:

- The total length of GPIB cable in the system must be either less than 66 feet (20 m) or 6.5 feet (2 m) times the number of devices connected to the system, whichever is less.
- No more than 15 devices can be connected to the system. For operation with maximum reliability, individual cable lengths should be less than 13 feet (4 m) and at least two-thirds of the devices connected to the system should be powered up when the system is operating.

Refer to IEEE Standard 488-1978 for additional guidelines covering GPIB operation at maximum data transfer rates.

Figure 5-1 illustrates a typical system interconnection configuration showing the flexibility of GPIB. More than one Plotter can be connected in the system.

In addition, the Plotter can be connected to other GPIB devices with or without a GPIB controller present.

CONNECTING PLOTTER TO OTHER DEVICES

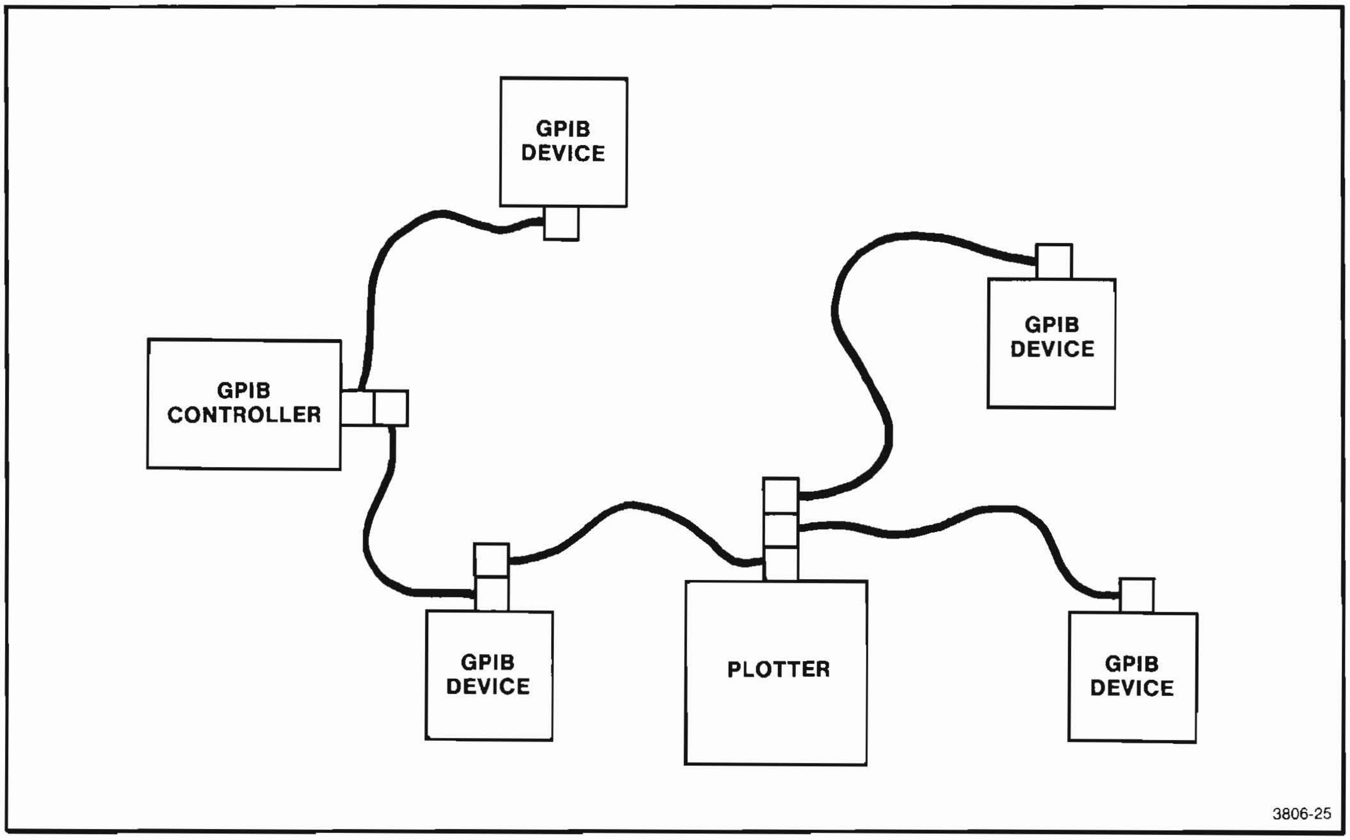


Figure 5-1. Typical GPIB System Interconnections.

SETTING UP THE PARAMETER ENTRY CARD

Selections made on the GPIB interface Parameter Entry Card lines permit the Plotter to operate in a variety of communications environments. The following describes the choices and helps determine selections that are applicable to particular communication environments.

The Parameter Entry Card lines described here are the Interface Select line, the Initial Command/Response Format line, the GPIB Device Address line, the Interface Mode line, and the Interface Functions line (see Figure 5-2). Refer to the description of the Parameter Entry Card in the *4663 Operator's Manual* for help in making any of the selections described in this section.

INTERFACE SELECT LINE

Selections made on the Interface Select line determines which hardware interface is active. For a Plotter shipped with an Option 01 Interface (Serial and GPIB), selecting "1" on the Interface Select line activates the installed Serial interface (RS-232-C or Current Loop). Selecting "2" on this line activates the GPIB interface. If Option 04 (GPIB only) is installed, the "1" selection activates the GPIB interface and the "2" selection is inactive. Table 5-1 summarizes the possible selections.

NOTE

If the straps on the Interface circuit board have been reset, the selection activated by "1" or "2" will be different from that shown in Table 5-1. Refer to the 4663 Service Manual for additional information.

Table 5-1

INTERFACE DESIGNATIONS

Interface(s) Installed	**Option Number**	**Interface Select Line**	
		1	**2**
Serial Only	(Standard)	Serial	Not used
Serial and GPIB	(Option 01)	Serial	GPIB
GPIB Only	(Option 04)	GPIB	Not used

INITIAL COMMAND/RESPONSE FORMAT LINE

Selections made on the Initial Command/Response Format line configures the GPIB interface to accept commands and to output responses in a specified format. Refer later in this section to *Introduction to GPIB Commands and Response* for a more detailed description of the various command and response forms used with each format.

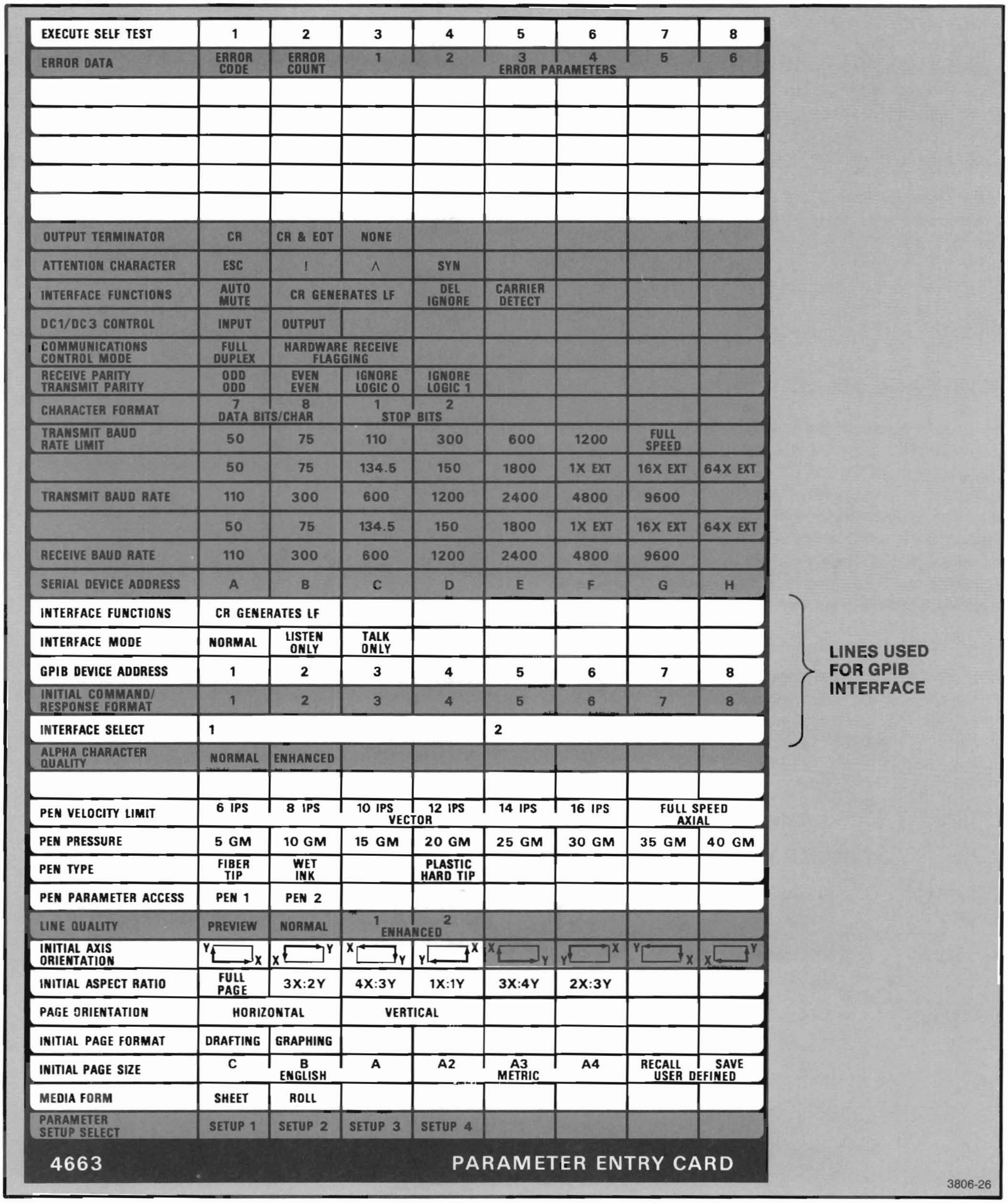
If you are installing the Plotter in an existing system, the selection made on the Initial Command/Response Format line must agree with the existing system formats. If the system is to be reprogrammed, then the format choice can be based upon the ease with which the formats are generated by the system. The format selection must also satisfy the message termination requirements of the system devices. The following paragraphs describe each of the Initial Command/Response Format line selections and indicate how an appropriate choice is made.

NOTE

The selection on the Initial Command/Response Format line configures the interface: (l) at power-up, (2) when a new selection is made on the Initial/Command Response Format line, (3) when the GPIB interface is first selected on the Interface Select line, and (4) after an INTERFACE PARAMETER RESET command. The SELECT COMMAND/RESPONSE FORMAT command, when received, overrides the Initial Command/Response Format selection, but does not modify it. This permits the configuration defined by the Parameter Entry Card selection to be restored when one of the three previous conditions listed occurs.

The following describes only the Initial Command/Response Formats as they apply to a GPIB interface. The function of each format changes when the Serial interface in selected. Refer to Section 4 for more information about the Serial Command/Response Formats.

SETTING UP PARAMETER ENTRY CARD



EXECUTE SELF TEST	1	2	3	4	5	6	7	8
ERROR DATA	ERROR CODE	ERROR COUNT	1	2	3 ERROR PARAMETERS	4	5	6
OUTPUT TERMINATOR	CR	CR & EOT	NONE					
ATTENTION CHARACTER	ESC	!	∧	SYN				
INTERFACE FUNCTIONS	AUTO MUTE	CR GENERATES LF		DEL IGNORE	CARRIER DETECT			
DC1/DC3 CONTROL	INPUT	OUTPUT						
COMMUNICATIONS CONTROL MODE	FULL DUPLEX	HARDWARE RECEIVE FLAGGING						
RECEIVE PARITY TRANSMIT PARITY	ODD ODD	EVEN EVEN	IGNORE LOGIC 0	IGNORE LOGIC 1				
CHARACTER FORMAT	7 DATA BITS/CHAR	8	1 STOP BITS	2				
TRANSMIT BAUD RATE LIMIT	50	75	110	300	600	1200	FULL SPEED	
	50	75	134.5	150	1800	1X EXT	16X EXT	64X EXT
TRANSMIT BAUD RATE	110	300	600	1200	2400	4800	9600	
	50	75	134.5	150	1800	1X EXT	16X EXT	64X EXT
RECEIVE BAUD RATE	110	300	600	1200	2400	4800	9600	
SERIAL DEVICE ADDRESS	A	B	C	D	E	F	G	H
INTERFACE FUNCTIONS	CR GENERATES LF							
INTERFACE MODE	NORMAL	LISTEN ONLY	TALK ONLY					
GPIB DEVICE ADDRESS	1	2	3	4	5	6	7	8
INITIAL COMMAND/ RESPONSE FORMAT	1	2	3	4	5	6	7	8
INTERFACE SELECT	1				2			
ALPHA CHARACTER QUALITY	NORMAL	ENHANCED						
PEN VELOCITY LIMIT	6 IPS	8 IPS	10 IPS VECTOR	12 IPS	14 IPS	16 IPS	FULL SPEED AXIAL	
PEN PRESSURE	5 GM	10 GM	15 GM	20 GM	25 GM	30 GM	35 GM	40 GM
PEN TYPE	FIBER TIP	WET INK		PLASTIC HARD TIP				
PEN PARAMETER ACCESS	PEN 1	PEN 2						
LINE QUALITY	PREVIEW	NORMAL	1 ENHANCED	2				
INITIAL AXIS ORIENTATION	Y X	X Y	X Y	Y X	X Y	Y X	Y X	X Y
INITIAL ASPECT RATIO	FULL PAGE	3X:2Y	4X:3Y	1X:1Y	3X:4Y	2X:3Y		
PAGE ORIENTATION	HORIZONTAL		VERTICAL					
INITIAL PAGE FORMAT	DRAFTING	GRAPHING						
INITIAL PAGE SIZE	C ENGLISH	B	A	A2 METRIC	A3	A4	RECALL USER DEFINED	SAVE
MEDIA FORM	SHEET	ROLL						
PARAMETER SETUP SELECT	SETUP 1	SETUP 2	SETUP 3	SETUP 4				

4663 PARAMETER ENTRY CARD

LINES USED FOR GPIB INTERFACE

3806-26

Figure 5-2. Parameter Entry Card Lines for the GPIB Interface.

Command/Response Format 1

Format 1 features are:

- 4663 Command Set
- Output Terminator EOI

This format permits all 4663 commands to be used. The arguments for PRINT and PRINT CENTERED CHARACTER commands are special string arguments that (1) can contain any character and (2) are terminated only by ET or by the EOI bus message sent with the last string data byte. For this format, response messages transmitted by the Plotter are terminated by the EOI bus message sent with the last data byte transmitted. This is the most compatible format for use with the following GPIB devices and controllers:

- TEKTRONIX 4050 Series Graphic Computer Systems
- TEKTRONIX 4924 Digital Cartridge Tape Recorder
- TEKTRONIX 4040 Series Computer/Controller Systems

This format is also used with other devices or controllers that allow input to be terminated with the EOI bus message sent with the last data byte transmitted.

Command/Response Format 2

Format 2 features are:

- 4663 Command Set
- Output Terminator CR, LF, and EOI

This format allows all 4663 commands to be used. The arguments for PRINT and PRINT CENTERED CHARACTER commands are special string arguments that can contain any character and are terminated only by ET or the EOI bus message sent with the last string data byte. For this format, response messages transmitted by the Plotter are terminated by a CR followed by a LF sent with the EOI bus message.

This is the most compatible format for use with the following GPIB devices and controllers:

- HP 9800 Series
- FLUKE 1720A
- PET 2001

This format is also used with other devices or controllers that permit input to be terminated by CR, CR LF, LF, or LF and an EOI bus message.

Command/Response Format 3

Format 3 features are:

- 4663 Command Set
- No Print String Arguments
- Output Terminator EOI

This format is identical to Format 1 except that print string arguments are not available. This format allows the same string forms to be used for all string arguments.

Command/Response Format 4

Format 4 features are:

- 4663 Command Set
- No Print String Arguments
- Output Terminator, CR, LF, and EOI

This format is identical to Format 2 except that print string arguments are not available. This allows the same string forms to be used for all string arguments.

Command/Response Format 5

Format 5 features are:

- 4662 Compatible Command Set
- Output Terminator EOI

This format permits a 4663 Plotter to be installed in systems that previously used a TEKTRONIX 4662 Plotter. All 4663 commands can be used, allowing the system to be upgraded. All 4663 output generating commands that do not have 4662 equivalents generate their output in the same form as for Format 1.

This format is the most compatible format for use with the following GPIB devices and controllers:

- TEKTRONIX 4020 Series Terminals with Option 04
- TEKTRONIX 4081
- TEKTRONIX 4014/4015 with Option 05 and a TEKTRONIX 4907

This format is also usable with any of the devices and controllers listed in Format 1 if the system software was written for a TEKTRONIX 4662.

Command/Response Format 6

Format 6 features are:

- 4662 Compatible Command Set
- Output Terminator, ᶜʀ, ᴸꜰ, and EOI

This format allows a 4663 to be installed in a system which previously used a TEKTRONIX 4662. All 4663 commands can be used, allowing the system to be upgraded. All 4663 output generating commands that do not have 4662 equivalents generate their output in the same form as for Format 2. All remaining output generating commands create their output in Format 6.

This is the most compatible format for use with any of the devices and controllers listed in Format 2 if the system software was written for a TEKTRONIX 4662. Refer to Appendix H for more information.

Command/Response Format 7

Format 7 features are:

- 4663 Serial Interface Commands
- No Output

This format permits commands to be transmitted to the Plotter via the GPIB in the form used with a Serial interface operating in Serial Command/Response Format 1. This allows commands generated by or received from a non-GPIB system to be transmitted (via GPIB) to the Plotter without modification. The attention and device address characters used in serial Plotter commands are assumed to be the characters ᴱc and **A**, respectively.

All output generating commands and Serial interface commands are ignored and no output is generated in this format.

This is the most compatible format for use with the following GPIB devices and controllers:

- TEKTRONIX 4014 Option 05 with a TEKTRONIX 4907

This format is also used with other devices or controllers that need to transmit Serial format type commands to the Plotter.

GPIB DEVICE ADDRESS LINE

Selections made on the GPIB Device Address line configure the interface to respond to GPIB bus messages that contain a primary address if the primary address agrees with the value selected. This address selection permits the Plotter to be uniquely addressed by the bus controller in a GPIB system consisting of several devices.

INTERFACE MODE LINE

Selections made on the Interface Mode line either (1) permit the programmable configuration of the system where the controller assigns listeners and talkers (normal), or (2) perform a manual configuration of the Plotter as a listener or as a talker (Listen Only, Talk Only).

INTERFACE FUNCTIONS LINE

The selection of CR GENERATES LF permits the interface to automatically generate a line feed character following each carriage return character that is sent to the Plotter for printing. Use this function if the data to be printed does not contain line feeds following each carriage return.

NOTE

The 4050 Series Graphic Systems do not automatically generate a ᴸꜰ character when sending print data. Thus, select the CR GENERATES LF function when communicating with these controllers.

COMMUNICATING WITH OTHER DEVICES

The following pages describe how devices instruct the Plotter to transmit or receive data bytes and how the Plotter interacts with a GPIB controller to change or report its status. These actions are described both in terms of the GPIB functions required by the Plotter interface and, for TEKTRONIX 4050 Series Controllers, the typical commands the controller uses to command the required GPIB functions. Refer to the specific operator's manual for controllers other than the TEKTRONIX 4050 Series to see how the needed GPIB functions are commanded. For precise details about GPIB operation, refer to both the IEEE Standard 488-1978 and Appendix E.

GPIB OPERATION

The GPIB allows data bytes, which consists of commands and responses, to be transmitted through the bus from a single *talker* (a data byte transmitter) to one or more *listeners* (data byte receivers). The designation of the talker and listener on the bus can be changed to allow data byte transfer between any connected devices. This designation either is changed for each data byte or remains in effect for an extended period to allow the transfer of many data bytes.

The designation of the talker and listener devices is done under control of a system controller. The controller uses special bus functions to designate the desired talker and listener(s), to initiate or interrupt the transmission of data bytes between devices, and to determine the status of the devices in the system. An example of a device with controller capability is the TEKTRONIX 4050 Series Graphic System. For systems that do not have a controller, the designation of the Plotter as a talker and listener and the initiation of data byte transfers are done manually through the Plotter's Parameter Entry Card and front panel functions.

INTERFACE SWITCHING FUNCTION

The Interface Switching function provides a flexible, manual means to control the Plotter's response to messages on the GPIB. Two front panel switches of the Plotter (OFFLINE/ONLINE and LOCAL/REMOTE) provide four different conditions for the Interface Switching function: Online/Remote, Online/Local, Offline/Remote, Offline/Local.

Online/Remote

This is the normal mode of operation and is automatically selected upon power-up. The Plotter receives all bus messages and executes received commands.

Online/Local

The Plotter receives all bus messages but ignores all commands received. This mode permits plotting to be inhibited without stopping GPIB communication when the controller expects a device at the current Plotter address.

Offline/Remote

The Plotter ignores all GPIB messages. This mode is equivalent to removing the GPIB cable from the Plotter.

Offline/Local

The Plotter ignores all GPIB messages. This mode is equivalent to removing the GPIB cable from the Plotter.

DEVICE-TO-PLOTTER TRANSMISSION

The following paragraphs describe the actions required to transfer commands (in the form of data bytes) from a talker to a Plotter acting as the listener. The temporary storage of Plotter commands is also described.

The Plotter As a Listener

The controller directs the Plotter to listen by sending the ATN (Attention) and the MLA (My Listen Address) bus messages at the same time. The MLA message contains a device address that must correspond to the device address specified on the Paramenter Entry Card. Refer to Figure 5-3 for a chart of GPIB message bytes.

Once the Plotter is addressed as a listener, device dependent messages consisting of Plotter commands are sent as data bytes (DABs) from the talker to the Plotter (listener). The EOI (End Or Identify) bus message sent with the last DAB indicates that the message is complete and terminates the last command of the message.

After all commands are sent to the Plotter, the controller may cancel the Plotter's listen function by sending the ATN and the UNL (unlisten) messages at the same time to the Plotter.

DAB Commands From a Controller. Here, data bytes (DABs) containing the command code and arguments are transmitted from the controller (acting as a talker) to the Plotter. For example, to transfer a DRAW command to the Plotter from a 4050 Series controller, the following basic statement is executed by the graphic system:

PRINT @**1,32: "D50,50";**

This statement has the 4050 Series Controller send the ATN and MLA 1 bus messages at the same time, which enables a Plotter having Device Address 1 to be the listener. The controller is then automatically enabled as the talker with the data bytes D50,50 transmitted through the GPIB to the Plotter. The EOI (End Or Identify) bus message is sent with the last data byte to tell the Plotter that the transmission is complete. The controller then sends UNT (Untalk) and UNL (Unlisten) messages to cancel the established talker and listener functions.

NOTE

The 4050 Series Graphic System functions as both a controller and a talker. Refer to the Print and WByte statements and to the Interfacing Information appendix (located in the 4050 Series Graphic System Reference Manual) for further discussion of these actions. The actual transmitted data bytes that define a specific command are discussed under Introduction to GPIB Commands in this section.

MSA Commands From a Controller. An alternative specification for the DRAW command to be transmitted from the 4050 Series Graphic System would be:

DRAW @**1:50,50**

This example has the 4050 Series controller send the ATN and MLA 1 bus messages at the same time and then follows with the MSA 20 (My Secondary Address 20) message while the ATN message is still true. The MLA 1 message enables a Plotter having Device Address 1 to be a listener. The MSA 20 message is equivalent to the **D** command code for the DRAW command shown in the previous example. This nonstandard use of the MSA message is unique to the 4050 Series Graphic System. Refer to *Introduction to GPIB Commands* in this section for a discussion of Plotter commands having equivalent MSA definitions. The remainder of the statement is processed as in the previous **PRINT** @**1,32:** statement used to transfer the DRAW arguments of **50,50** to the Plotter.

As long as the Plotter remains enabled as a listener, several commands in succession can be transmitted from the talker to the Plotter.

NOTE

The talker could be a device other than the 4050 Series Graphic System, such as a GPIB mass storage device. Here, the controller enables the mass storage device as the talker and allows it to transfer a plot to the Plotter. When the transfer is completed, the controller cancels the talker and listener functions.

DATA BUS BITS DIO7 DIO6 DIO5				Ø Ø Ø	Ø Ø 1	Ø 1 Ø	Ø 1 1	1 Ø Ø	1 Ø 1	1 1 Ø	1 1 1
DIO4	DIO3	DIO2	DIO1	ADDRESSED COMMANDS	UNIVERSAL COMMANDS	PRIMARY LISTEN ADDRESSES		PRIMARY TALK ADDRESSES		SECONDARY ADDRESSES	
Ø	Ø	Ø	Ø	NUL 0	DCE 16	SP MLA0 32	0 MLA16 48	@ MTA0 64	P MTA16 80	` MSA0 96	p MSA16 112
Ø	Ø	Ø	1	SOH GTL 1	DC1 LLO 17	! MLA1 33	1 MLA17 49	A MTA1 65	Q MTA17 81	a MSA1 97	q MSA17 113
Ø	Ø	1	Ø	STX 2	DC2 18	" MLA2 34	2 MLA18 50	B MTA2 66	R MTA18 82	b MSA2 98	r MSA18 114
Ø	Ø	1	1	ETX 3	DC3 19	# MLA3 35	3 MLA19 51	C MTA3 67	S MTA19 83	c MSA3 99	s MSA19 115
Ø	1	Ø	Ø	EOT SDC 4	DC4 DCL 20	$ MLA4 36	4 MLA20 52	D MTA4 68	T MTA20 84	d MSA4 100	t MSA20 116
Ø	1	Ø	1	ENQ PPC 5	NAK PPU 21	% MLA5 37	5 MLA21 53	E MTA5 69	U MTA21 85	e MSA5 101	u MSA21 117
Ø	1	1	Ø	ACK 6	SYN 22	& MLA6 38	6 MLA22 54	F MTA6 70	V MTA22 86	f MSA6 102	v MSA22 118
Ø	1	1	1	BEL 7	ETB 23	' MLA7 39	7 MLA23 55	G MTA7 71	W MTA23 87	g MSA7 103	w MSA23 119
1	Ø	Ø	Ø	BS GET 8	CAN SPE 24	(MLA8 40	8 MLA24 56	H MTA8 72	X MTA24 88	h MSA8 104	x MSA24 120
1	Ø	Ø	1	HT TCT 9	EM SPD 25	) MLA9 41	9 MLA25 57	I MTA9 73	Y MTA25 89	i MSA9 105	y MSA25 121
1	Ø	1	Ø	LF 10	SUB 26	* MLA10 42	: MLA26 58	J MTA10 74	Z MTA26 90	j MSA10 106	z MSA26 122
1	Ø	1	1	VT 11	ESC 27	+ MLA11 43	; MLA27 59	K MTA11 75	[MTA27 91	k MSA11 107	{ MSA27 123
1	1	Ø	Ø	FF 12	FS 28	, MLA12 44	< MLA28 60	L MTA12 76	\ MTA28 92	l MSA12 108	*| MSA28 124
1	1	Ø	1	CR 13	GS 29	- MLA13 45	= MLA29 61	M MTA13 77	] MTA29 93	m MSA13 109	} MSA29 125
1	1	1	Ø	SO 14	RS 30	. MLA14 46	> MLA30 62	N MTA14 78	^ MTA30 94	n MSA14 110	~ MSA30 126
1	1	1	1	SI 15	US 31	/ MLA15 47	? UNL 63	O MTA15 79	_ UNT 95	o MSA15 111	DEL 127

- **Shaded codes are those usable by the 4663 Plotter.**

- KEY

CAN	ASCII Character
SPE	GPIB Code
24	Decimal

MLAn	Primary Listen Address for device n
MTAn	Primary Talk Address for device n
MSAn	Secondary Address (MSA Commands)
UNL	UNLISTEN command
UNT	UNTALK command
SPE	SERIAL POLL ENABLE command
SPD	SERIAL POLL DISABLE command
SDC	Selected Device Clear Command
DCL	Device Clear Command

*| on some keyboards or systems

3806-27

Figure 5-3. GPIB Code Chart.

Offline Plotting. You can transfer a plot to a Plotter without a controller connected to the GPIB. A typical system involving a TEKTRONIX 4924 Digital Cartridge Tape Recorder and a Plotter is shown in Figure 5-4. To transfer a plot, enable the Plotter as a listener by selecting LISTEN ONLY on the Interface Mode line of the Parameter Entry Card. The desired tape file should be located on the 4924's tape cartridge. The 4924 is then enabled as a talker by pressing the TALK button on the 4924 front panel (the 4924's ONLINE button should be in the *out* position). The contents of the desired tape cartridge file will be transferred to the Plotter.

NOTE

When the plotter is in LISTEN ONLY mode, it does not require an MLA message with a device address that corresponds to its address. The Plotter will interpret all data bytes seen on the GPIB as commands. Therefore, other devices should not exchange data while the Plotter is connected to the bus in LISTEN ONLY mode.

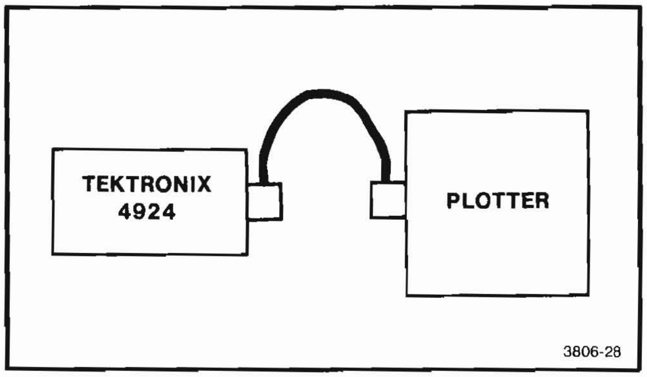


Figure 5-4. Typical OFFLINE Configuration.

Input Buffering

As commands are received from the talker, they are placed temporarily in an input buffer of the Plotter's memory. (Temporary storage is needed when commands are received faster than the Plotter can execute them.) When the Plotter is ready for another command, commands are withdrawn from the input buffer and executed. The following describes:

- How the input buffering function operates.
- How much memory is needed to store a Plotter command.
- How much memory is available under various conditions to store commands.

Buffer Operation. Because the GPIB is a handshake bus, data bytes transfer only at the speed the talker or listener devices can handle. If the Plotter is busy processing previous commands and cannot read the data bytes as fast as they are being transmitted by the talker, the data byte transfer automatically halts until the Plotter is ready. However, this action ties up the GPIB so that the controller cannot talk to other devices. When the Plotter has completed its processing and is waiting for the pen to finish its current motion, additional commands are received from the interface and stored in the input buffer. When all data bytes in the plot have been received from the talker, the controller can reconfigure the bus to talk to other devices while the Plotter is executing the buffered commands.

NOTE

The buffering is done only after the Plotter completes its command processing and is waiting for pen motion to finish. Therefore, when the Plotter is executing mostly nonmotion commands or short-motion commands (which require almost continuous processing), there is little buffering done.

Commands are received from the interface as data bytes. For Plotter commands, these data bytes are ASCII characters that comprise the command code and arguments of the Plotter commands. The interface converts commands expressed as ASCII characters into a corresponding internal form. Any interface commands are executed at once (refer to Section 8 for a listing of these commands). All other commands are stored in the input buffer.

Storage Requirements for Commands. All commands are received by the Plotter in the various command forms described under *Introduction to GPIB Commands And Responses*, later in this section.

The internal form into which the Plotter commands are converted requires the following storage in the input buffer:

- Two bytes of storage for each command code, PLUS
- One byte of storage for each character argument,
- And five bytes of storage for each numeric argument.

For commands with string arguments, a separate command is formed for each string character. For commands that have the command code followed by multiple groups of arguments, a separate command is formed for each argument group.

The internal form for each command is the same regardless of the command format used to transmit the command to the Plotter. For some cases, this means that more bytes of memory are needed for a command stored in the input buffer than it took characters to transmit the command to the Plotter. The ratio of these respective number of characters is defined as the *internal expansion factor.*

The following examples show different internal expansion factors:

- An alpha character takes one character to express and three bytes to store, giving an internal expansion factor of 3.
- A typical DRAW command, **D 100,10** takes seven bytes to express and twelve bytes to store, giving a 1.7 internal expansion factor.

Memory Limits for the Input Buffer. The Plotter memory available for buffer operation is used in two ways. First, some memory always establishes a minimum size for the input buffer. Second, the remaining memory is used on a "first come, first served" basis for (1) the expansion of the input buffer to store additional input commands, (2) the temporary buffering of responses formed by *output generating commands*, and (3) longer term storage for the following types of data:

- Graphic transform levels saved by a SAVE CURRENT TRANSFORM command.
- Programmable macros defined by Macro commands (available with Option 31 only).
- Downloadable characters defined by Downloadable Character commands (available with Option 32 only).

Refer to the individual command descriptions to determine the amount of memory space required for command storage. When stored data is subsequently deleted, allocated memory for that data becomes available again for other uses.

While a programmable macro or downloadable character definition is in process or if most available memory has stored programmable macros or downloadable characters, the input buffer is forced to assume its minimum size. The number of commands stored while in this minimum size depends upon the command type. For example, about 200 PRINT CHARACTER commands or about 50 MOVE and/or DRAW commands can be stored. The number of external characters required to define these commands depends upon the command style being used. If no transforms, programmable macros, or downloadable characters are defined and no output responses are stored, then the input buffer expands to a maximum size. In a standard Plotter with 8K of memory, approximately 340 MOVE and/or DRAW commands or 1360 PRINT CHARACTER commands are storable.

NOTE

When you initialize the interface by the front panel Initialize function, the input buffer is cleared.

PLOTTER-TO-DEVICE TRANSMISSION

The Plotter transmits several types of information to other devices. The following pages discuss the transmission of output information and include:

- How responses generated by Plotter commands are temporarily buffered.
- How these responses are transmitted (as data bytes) from a Plotter acting as a talker to a listening device.

Output Buffering

Plotter output (which is responses formed by Plotter commands) is transmitted in the form of output blocks. Each output block contains two numeric response values plus a numeric TAG value. Refer to *Introduction to Serial Commands and Responses* in this section for a description of the output blocks.

Only one output block is generated for commands whose response consists of one value (such as the READ FORM LENGTH command) or two values (such as the DIGITIZE command). Two or more output blocks are generated for commands that return more than two values. The size of these output response blocks may vary from 23 to 35 data characters, depending upon the numeric values of the responses. As the output response blocks are generated, they are are stored in the Plotter's memory until transmission is initiated. Refer to the command description sections for a discussion of the response values set up for each output generating command.

NOTE

The Plotter memory required to store output blocks reduces the amount available for input command storage, programmable macros, downloadable characters, and transforms. Thus, the buffering of output responses should be kept to a minimum.

Normally the response to one output generating command is read by the controller before another output generating command is received by the Plotter. Enough memory is always reserved for output buffering so that a single output response can be buffered even if all the remaining memory has been allocated to other functions.

A maximum of 127 response blocks can be stored if memory is available. The amount of buffer space needed to store the responses is the sum of the number of bytes in each response block plus one byte per output block. Because memory is allocated to buffering functions in 62-byte increments, you can determine the number of memory bytes required to store this number by rounding the number of bytes up to the next multiple of 62. The output buffer is cleared with the front panel Initialize function.

The Plotter As a Talker

The controller directs the Plotter to become a talker by sending the ATN (Attention) and MTA (My Talk Address) bus messages at the same time. The MTA message contains a device address that must correspond to the selected address for the GPIB Device Address line on the Parameter Entry Card. Refer back to Figure 5-3 for a chart of GPIB message bytes.

Once the Plotter is addressed as a talker and the bus indicates that a device has been addressed as a listener and is ready to receive, the Plotter begins to transmit bytes from the output buffer as bus data bytes. The Plotter transmits one output block from the output buffer followed by an output termination sequence (defined by the current Command/Response Format). Transmission then ceases until another MTA bus message is received. If the output buffer is empty, the bus pauses until output is generated.

The output termination sequence informs the controller that the last byte of the output block was transmitted by the talker and that the bus can be reconfigured for another use. The termination sequence also identifies the last data byte to the listener, so that the listener can perform any required end-of-transmission tasks (such as closing files). Different controllers and listeners look for different terminations. The most common terminations are:

- Where the EOI bus message is sent with the last data byte transmitted.
- Where the transmission is terminated by a carriage return or a carriage return followed by a line feed. For some GPIB devices the EOI bus message sent with the line feed may not be required.

For the first type of termination, the last data byte is included in the transmission. For the second type of termination, the termination characters are discarded when received by the listener. These two types of termination requirements are satisfied by the two Plotter output terminators that you select:

- The EOI bus message sent along with the last data byte of the message
- A carriage return followed by a line feed sent with the EOI bus message.

The termination used is specified by the current Command/Response format specification.

Online Digitizing Example. To cause a single digitize response to be generated and transferred to a 4050 Series Graphic System, the following BASIC statements would be executed by the Graphic System:

```
PRINT @1,32:"G";
INPUT @1,32: X,Y,T
```

During execution of the PRINT statement, the 4050 Series controller sets the Plotter at Device Address 1 as the listener (with the MLA 1 and ATN bus messages) and then sends the "G" (digitize) command code. Next, the controller cancels the listen function with the UNL bus message and sends the MTA 1 and ATN messages to enable the Plotter as a talker. The Plotter processes the "G" command code and generates an output block consisting of the desired XY coordinates and the associated TAG value.

The three values in the output block are converted into ASCII Decimal Floating Point numbers and entered into the output buffer. Command/Response Format 1 is used for 4050 Series Controllers. Thus, the values are then transmitted with the EOI bus message sent with the last data byte. The 4050 Series Controller reads these data bytes and assigns values of the numbers to the three target variables (X, Y, and T) in the INPUT BASIC statement. The 4050 Series Controller then sends the UNT and the ATN bus message to cancel the Plotter talk function. Manually digitized coordinates could be returned in a similar fashion by using the OPERATOR DIGITIZE ENABLE command (refer to *SRQ and Serial Poll Actions* later in this section.

Offline Digitizing Example. You can manually transfer digitized coordinates from the Plotter to a storage device offline without a controller connected to the GPIB. A typical system involving a TEKTRONIX 4924 Digital Cartridge Tape Recorder and the Plotter is shown in Figure 5-4. To transfer digitized coordinates, enable the Plotter as a talker by selecting TALK ONLY on the Paramenter Entry Card. You mut also select Command/Response Format 1. The desired tape file is located on the 4924's tape cartridge and the 4924 is enabled as a listener by pressing the LISTEN button on its front panel (the 4924's ONLINE button should be in the *out* position). Using the Plotter's front panel Move Point or Draw Point functions, digitized coordinates are generated, transferred to the 4924, and stored on the tape cartridge. The Last Point function generates the final coordinate and the final output termination sequence causes the 4924 to close the tape file.

NOTE

When the 4663 Plotter is in Talk Only mode, the EOI Bus Message is sent only when the Last Point function is used.

INTERACTING WITH THE CONTROLLER

The previous topic described the exchange of DABs between a listener and a talker via the GPIB. Information here describes Plotter related details of some functions performed by the controller to set up and terminate these exchanges.

Interrupting Data Exchanges

If a controller needs to interrupt the talker/listener's use of the GPIB, you must direct the controller to send the ATN bus message along with the UNT and the UNL bus messages. This cancels the talker and listener functions and permits the controller to send other messages on the bus. Once the controller is finished with the bus, the ATN bus message, plus the MTA amd MLA bus messages will reestablish the original talker/listener exchange. The Plotter allows exchanges to be interrupted and reestablishd, either as a listener or as a talker, except when the current Command/Response Format is 5 or 6. For these formats, a UNL message, received while a string argument is being received, will terminate the string argument. If the termination was premature, a UNL message causes undesired results.

IFC (Interface Clear) Actions

Often, it is desirable to force interfaces of GPIB devices to a known state. Before beginning a new operation, the controller forces the interfaces of all devices on the GPIB to a nonactive state simultaneously by sending the IFC (interface clear) message. For the Plotter, the IFC message causes any listener, talker, or serial poll actions to be cancelled. If the current Command/Response Format is 5 or 6, a Device Clear function is also performed. For other Command/Response Formats, the input or output buffers are not cleared.

The IFC message can occur at any time. To be sure of a response under all conditions, the IFC message should be asserted for at least 300 μsec (there should be a delay of about 200 μsec after the IFC message is removed before new GPIB messages are transmitted). While the IFC message is asserted, the Plotter ignores all GPIB commands except DEVICE CLEAR.

These actions are required due to hardware limitations. The Plotter may not recognize an IFC message that is asserted for only the minimum time required by IEEE Standard 488-1978 (100 μsec). The Plotter may not respond to an IFC message within the maximum time required by IEEE Standard 488-1978 (100 μsec).

DCL/SDL Actions (Device Clear/Selected Device Clear)

The DCL bus message (sent along with the ATN message) performs four functions:

- Causes the input and output buffers to be cleared (including clearing any partial commands received).
- Cancels any SRQ message.
- Cancels any Operator Digitize Enable (if the initial Command/Response Format is 5 or 6, and the Operator Digitize function is permanently enabled).
- Turns off the Plotter's POINT light.

The DCL bus message performs similar functions to all devices connected to the GPIB. The SDC message performs the Device Clear function only on the device addressed by the MLA message. (The SDC message must be preceeded by a MLA and ATN message and be sent with an ATN message.)

SRQ (Service Request) and Serial Poll Actions

Another general function of the GPIB controller is to respond to asynchronous requests for service from devices on the GPIB. Devices on the GPIB can request service by sending an SRQ message. If the controller has enabled this function, the SRQ message interrupts the controller's current task. The controller must then, if there is more than one possible request, determine what action is being requested. To do this, the controller performs a Serial Poll function, which reads a status byte (the Serial Poll Response Byte) from each device on the GPIB. A bus device indicates that it is sending the SRQ message by setting Bit 7 (Service Request) in its Serial Poll response byte. Other bits in the Serial Poll response byte then indicate which of the possible requests for that device are active. As soon as the Serial Poll response byte (which contains the active Service Request bit) is read, the device stops transmitting the SRQ messages. The Service Request bit of the Serial Poll response byte remains active until the condition which generated the request is satisfied.

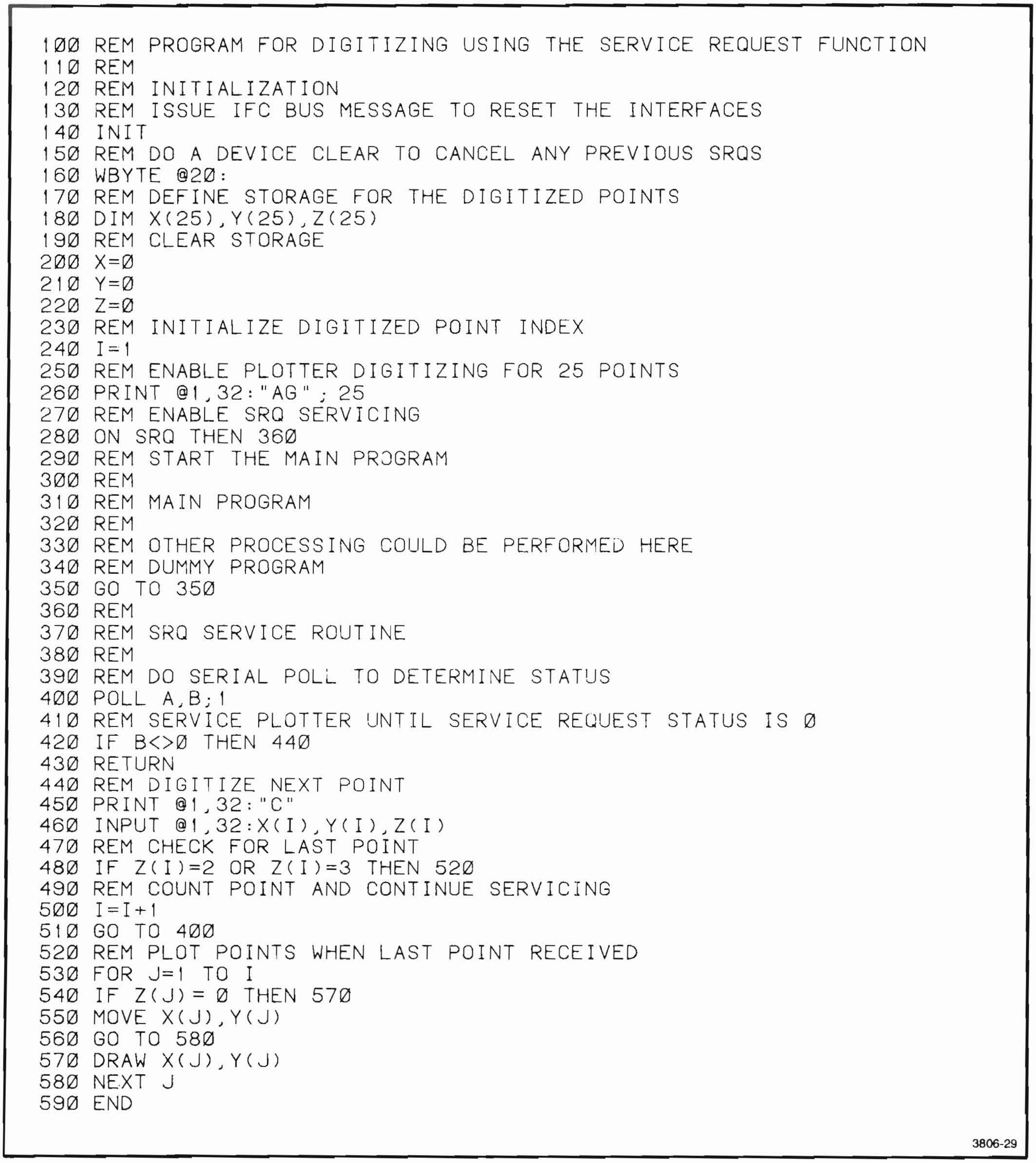
The only condition that causes the service request bit to be set is when an output block (generated by the Operator Digitize function) is ready to be read. The remaining bits in the Serial Poll response byte are not used. The value of the Serial Poll response byte can be either 0 or 64 (decimal). Thus, the SRQ message is transmitted when one of the front panel Point functions generates a set of digitized coordinates. The Service Request bit remains asserted as long as any sets of digitized coordinates in the output buffer are waiting to be read. Refer to Section 13 for more information on the Digitize function. The Plotter responds to a Serial Poll function when NORMAL or TALK ONLY has been selected on the Interface Mode line of the Paramenter Entry Card, but not when LISTEN ONLY is selected.

The SRQ function allows digitized points to be read from the Plotter without tying up the GPIB by waiting for the next points to be generated. For example, the short program for a 4050 Series Graphic System shown in Figure 5-5 uses the SRQ function to read digitized points into the graphic system while the main program performs other processing. When a Last Point is digitized, all previously digitized points are displayed.

Alternatively, digitized input can be read directly by the main program. An example of this operation appears in the short program for a 4050 Series Graphic System shown in Figure 5-6. Here, the GPIB halts until coordinates are generated. For controllers that recognize SRQ messages, this function must be disabled or a dummy service routine added to reset the response to the SRQ message (if this mode of coordinate input is desired).

If the digitized point is requested by a CALL DIGITIZE command rather than by an INPUT command, then the Operator Digitize function does not always generate the SRQ messages (as required for TEKTRONIX 4662 compatibility).

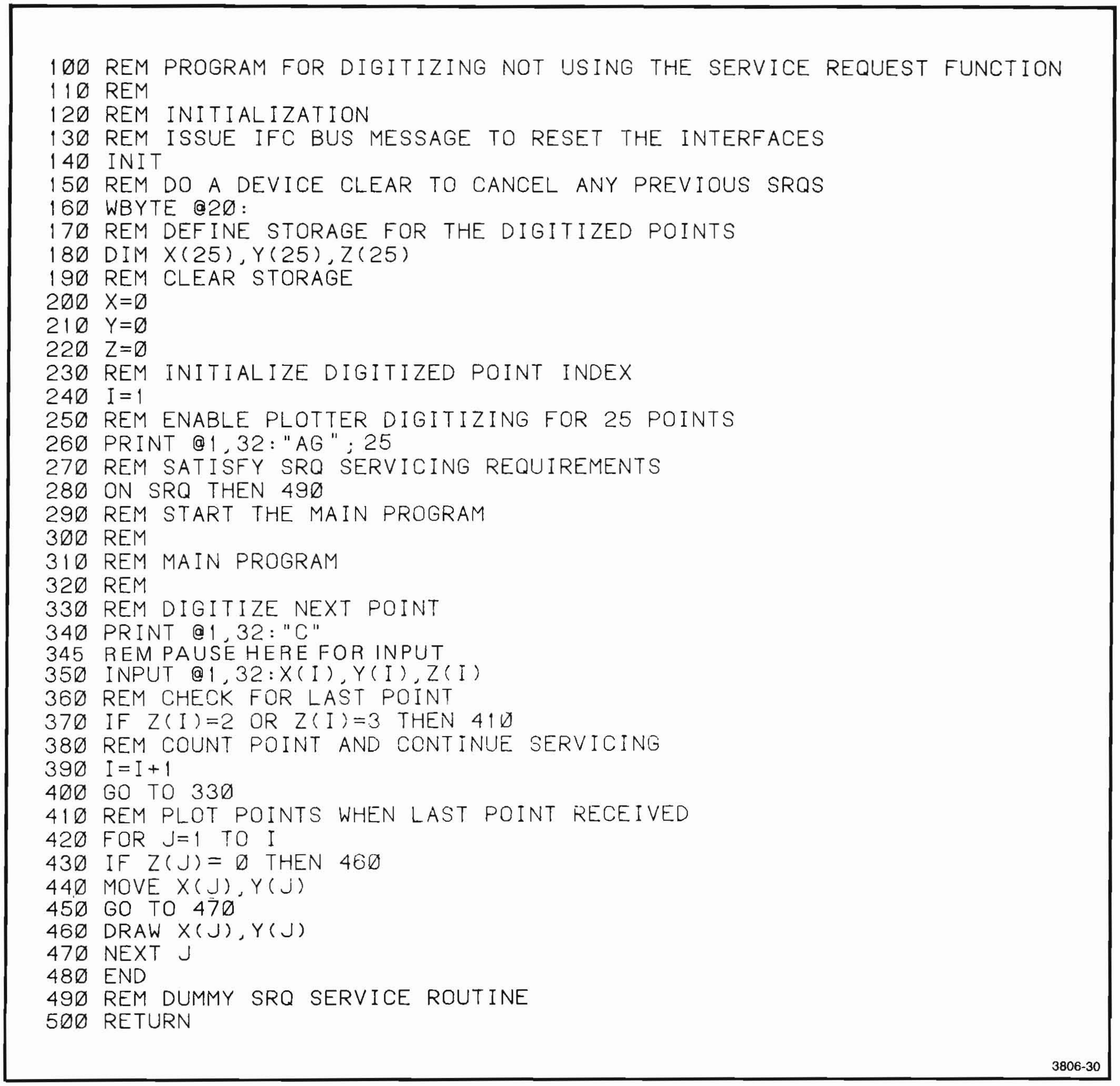
If the CALL DIGITIZE command has already addressed the Plotter as a talker when the digitize coordinates are generated, then the response is transmitted without the SRQ message being sent. However, if the next coordinate is generated before the Plotter is addressed as a talker again, then the SRQ message would be sent.



```
100 REM PROGRAM FOR DIGITIZING USING THE SERVICE REQUEST FUNCTION
110 REM
120 REM INITIALIZATION
130 REM ISSUE IFC BUS MESSAGE TO RESET THE INTERFACES
140 INIT
150 REM DO A DEVICE CLEAR TO CANCEL ANY PREVIOUS SRQS
160 WBYTE @20:
170 REM DEFINE STORAGE FOR THE DIGITIZED POINTS
180 DIM X(25),Y(25),Z(25)
190 REM CLEAR STORAGE
200 X=0
210 Y=0
220 Z=0
230 REM INITIALIZE DIGITIZED POINT INDEX
240 I=1
250 REM ENABLE PLOTTER DIGITIZING FOR 25 POINTS
260 PRINT @1,32:"AG";25
270 REM ENABLE SRQ SERVICING
280 ON SRQ THEN 360
290 REM START THE MAIN PROGRAM
300 REM
310 REM MAIN PROGRAM
320 REM
330 REM OTHER PROCESSING COULD BE PERFORMED HERE
340 REM DUMMY PROGRAM
350 GO TO 350
360 REM
370 REM SRQ SERVICE ROUTINE
380 REM
390 REM DO SERIAL POLL TO DETERMINE STATUS
400 POLL A,B;1
410 REM SERVICE PLOTTER UNTIL SERVICE REQUEST STATUS IS 0
420 IF B<>0 THEN 440
430 RETURN
440 REM DIGITIZE NEXT POINT
450 PRINT @1,32:"C"
460 INPUT @1,32:X(I),Y(I),Z(I)
470 REM CHECK FOR LAST POINT
480 IF Z(I)=2 OR Z(I)=3 THEN 520
490 REM COUNT POINT AND CONTINUE SERVICING
500 I=I+1
510 GO TO 400
520 REM PLOT POINTS WHEN LAST POINT RECEIVED
530 FOR J=1 TO I
540 IF Z(J) = 0 THEN 570
550 MOVE X(J),Y(J)
560 GO TO 580
570 DRAW X(J),Y(J)
580 NEXT J
590 END
```

3806-29

Figure 5-5. Operator Digitized Input Using the SRQ Function.



```
100 REM PROGRAM FOR DIGITIZING NOT USING THE SERVICE REQUEST FUNCTION
110 REM
120 REM INITIALIZATION
130 REM ISSUE IFC BUS MESSAGE TO RESET THE INTERFACES
140 INIT
150 REM DO A DEVICE CLEAR TO CANCEL ANY PREVIOUS SRQS
160 WBYTE @20:
170 REM DEFINE STORAGE FOR THE DIGITIZED POINTS
180 DIM X(25),Y(25),Z(25)
190 REM CLEAR STORAGE
200 X=0
210 Y=0
220 Z=0
230 REM INITIALIZE DIGITIZED POINT INDEX
240 I=1
250 REM ENABLE PLOTTER DIGITIZING FOR 25 POINTS
260 PRINT @1,32:"AG";25
270 REM SATISFY SRQ SERVICING REQUIREMENTS
280 ON SRQ THEN 490
290 REM START THE MAIN PROGRAM
300 REM
310 REM MAIN PROGRAM
320 REM
330 REM DIGITIZE NEXT POINT
340 PRINT @1,32:"C"
345 REM PAUSE HERE FOR INPUT
350 INPUT @1,32:X(I),Y(I),Z(I)
360 REM CHECK FOR LAST POINT
370 IF Z(I)=2 OR Z(I)=3 THEN 410
380 REM COUNT POINT AND CONTINUE SERVICING
390 I=I+1
400 GO TO 330
410 REM PLOT POINTS WHEN LAST POINT RECEIVED
420 FOR J=1 TO I
430 IF Z(J)= 0 THEN 460
440 MOVE X(J),Y(J)
450 GO TO 470
460 DRAW X(J),Y(J)
470 NEXT J
480 END
490 REM DUMMY SRQ SERVICE ROUTINE
500 RETURN
```

3806-30

Figure 5-6. Operator Digitize Input Without the SRQ Function.

INTRODUCTION TO GPIB COMMANDS AND RESPONSES

A wide range of command and argument forms permit communication with devices having varying output and input capabilities. The actual form of the commands and responses to be used is determined by the current Command/Response Format selection. Refer to *Setting Up the Parameter Entry Card* (previously described in this section) for help in selecting an appropriate Command/Response Format.

The following pages describe the structure of the actual commands sent over the GPIB to the Plotter. The user must refer to the specific device's manual to determine how transmission of these command forms is done over the GPIB. Some examples showing generation of these commands by a 4050 Series Graphic System will be shown and may give some guidance for the generation of Plotter commands by other devices.

NOTE

> *For GPIB Commands/Response Format 7, refer to the structure of the commands under Introduction to Serial Commands and Responses later in this section.*

PLOTTER COMMANDS

The general command form recognized by the GPIB interface is:

<Command Code> <Arguments>

Each portion of these commands is discussed along with any separators and terminators required to correctly punctuate the commands. Also presented is the Plotter's reaction to incomplete or unrecognized commands.

Command Code

The command code consists of one or two ASCII characters (either uppercase or lowercase) that uniquely identifies the command. The command code for each Plotter operation is defined in the command description sections of this manual. For Command/Response Formats 3 or 4, the command code can be followed by a space character used as a delimiter. For all formats where the command is not a PRINT or PRINT CENTERED CHARACTER command, optional carriage return or line feed characters following the command code are discarded.

NOTE

> *The single letter commands* **A** *or* **C** *(or* **D** *if null arguments are specified) should always be followed by a command terminator to ensure correct interpretation of the commands. (See NULL Arguments later in this section.*

Arguments

Any arguments defined for a particular command follow the command code. Specific arguments for each command are defined in the command description sections of this manual. Command arguments can be of two general types: numeric or string. Individual arguments are partitioned from each other by argument separators and the end of the argument set is marked by a command terminator. Refer to *Argument Separators and Command Terminators* later in this section.

Numeric Arguments. *Numeric arguments* are expressed in any of three standard forms: integer, floating point, or scientific notation. Examples of each form are shown below:

● −15	integer
● + 15.8	floating point
● 1.58E+ 01	scientific notation (E format)

When entering numeric arguments, follow these rules:

- When a numeric argument is positive, the "+ " sign is optional.
- Leading zeros are ignored.
- In the scientific notation form, express the mantissa as in an integer or floating point form.
- The E is either uppercase or lowercase, and the exponent is expressed as 1 or 2 digits.
- A "+ " sign is not required if the exponent is positive.
- A space character can be substituted for a "+ " sign in either the mantissa or the exponent. No other spaces should appear within the argument.
- Numeric arguments are always terminated by any character that is not part of a specified numeric form. (For example, a character which is not one of the digits 0 through 9 and the characters + — e E and a period.)

These numeric argument forms accomodate the number formats specified in ANSI Standard X3.42. The units, resolution, and allowable range of numeric arguments are unique to each command and are noted in the command description sections.

One special type of numeric argument is a selector argument used to specify one of a set of prenumbered choices defined for some commands. For Formats 5 and 6, leading "?" or "= " characters before a numeric argument are ignored.

String Arguments. A *string argument* represents a sequence of one or more ASCII characters in one of three forms:

- As a "delimited string"
- As an "undelimited string"
- A special form of "print string" is required for PRINT or PRINT CENTERED CHARACTER command arguments when the Command/Response Format is 1, 2, 5, or 6.

The delimited string form is described first. The following shows three simple examples with each representing the ASCII character A.

- "A"
- 'A'
- /65

A delimited string can be formed by preceding and following a group of one or more ASCII characters by the same delimiter (either ' or "). Alternatively, a delimited string may consist of a / (slant character) followed by the numeric argument that is equal to the ADE (ASCII Decimal Equivalent) value assigned to a given ASCII character. The ADE values for each ASCII character are shown in the ASCII Code Chart Appendix B. The delimited string representation is used to specify any ASCII characters with the exception of the string delimiter character(s) that cannot be sent by the device transmitting the Plotter commands. The ADE representation is used for any ASCII character. The two representations can be mixed to form a single, more complex delimited string.

Here are additional examples of delimited strings:

Delimited String	**Characters Represented**
"ABC"	ABC
'AB'/13/10'CD"	ABCRLFCD
'HeSPsaid:SP"ISPcan"'""t'	HeSPsaid:SP"ISPcan't"

String arguments can also be in the *undelimited string* form. An undelimited string is a sequence of ASCII characters, where the first character cannot be any of the following characters:

/ ' " SP , CR LF ;

After the first character, the string can contain any character. An undelimited string is terminated either by an EX character or by the EOI bus message received with the last byte of the string. In Formats 5 and 6, the UNL bus message also terminates these strings.

Print strings are special string arguments that are used in Command/Response Formats 1, 2, 5, and 6 as the arguments for PRINT or PRINT CENTERED commands. These strings begin immediately following the command code and can contain any ASCII character. These strings are terminated the same as undelimited strings.

Argument Separators and Command Terminators

Argument separators are special sequences of characters that define the end of one argument and the beginning of the next. A space or a comma is the most common argument separator. However, an argument separator can also be expressed in one of two more general forms:

- A sequence of one or more space, carriage return, or line feed characters.
- A sequence of zero or more space, carriage return, or line feed characters followed by a comma, followed by another sequence of zero or more space, carriage return, or line feed characters.

NOTE

The argument separator implies that another argument follows (if it is allowed). Thus, be sure that the last argument of a command is not followed by a space or comma character (which would be interpreted as an argument separator).

A *command terminator* is a character that marks the end of the command's last argument and indicates that no more arguments follow. A command is not executed by the Plotter until it has been terminated. There are two general types of command terminators: *discarded* and *reprocessed*.

Discarded command terminators explicitly delimit the command and, after they are recognized, are discarded without further interpretation. The discarded type command terminators are the semicolon and the EOI bus message sent together with the last byte of the command.

Reprocessed command termination characters terminate a command but are also then reprocessed as part of a following command. An example is when the command code for one command terminates the previous command.

Be sure that the last argument of a command is properly terminated. For example, a command code that terminates a numeric argument may not terminate a string argument. Various types of string arguments require their own terminators.

NOTE

In Formats 5 and 6, commands that have no arguments require no command terminators. In Formats 1, 2, 3, or 4, a command terminator with these commands is optional.

Multiple Argument Groups

A complete set of arguments for a command is defined as an *argument group*. For some commands, more than one argument group can follow the command code when multiple occurrences of the command are desired. For example, the DRAW command can be followed by multiple pairs of X,Y coordinates. The final argument group is followed by a command terminator. The commands that allow multiple argument groups are described in the command description sections.

NOTE

Commands allowing multiple argument groups should be carefully terminated. For example, if a space or comma character appears following the command, more arguments are assumed, causing undesired results.

NULL Arguments

A NULL argument can be specified in place of an expected numeric or string argument. A NULL numeric argument is given an 0 value and a NULL string argument is given a value of a single ASCII NULL character $^{N}{}_{U}$. NULL argument values are established by simply not specifying the argument and including only the argument separator (or terminator).

NOTE

The argument separators being used to specify NULL arguments must be unique. For example, two consecutive commas specify a NULL argument, but two consecutive spaces do not. For Command/Response Formats 3 or 4, a space following a command code is interpreted as a command code delimiter and not an argument separator defining a NULL argument. Thus, the argument separator must be a comma if a leading NULL argument is to be specified. NULL arguments are not allowed when using Command/Response Formats 5 or 6.

Optional Arguments

Some Plotter commands have optional arguments, which appear at the end of the argument list. These optional arguments are eliminated by entering a command terminator instead of a command separator before the optional argument position. The effects of optional argument selection in such commands are discussed in the individual command description sections.

NOTE

Optional arguments must be entered when using Command/Response Formats 5 or 6.

Command Interpretation

Errors in forming communications commands may cause the Plotter to receive unrecognizable commands. The following outlines the Plotter's response to these commands:

- If the command code is not recognized, then an Unrecognized Command error is generated and any following command arguments are discarded until the command is terminated. An undelimited string argument will terminate the discarded command.
- Errors in argument specifications (which cause a delimited string argument to be recognized when a numeric argument was expected) cause an Argument Type error to be generated and the command discarded.
- A numeric argument is interpreted as an undelimited string argument if a string argument is expected.
- A undelimited string argument is interpreted as a command terminator if a numeric argument is expected (unless the undelimited string is a number).

When an error is made while sending a command to the Plotter, a command termination should be sent before beginning another command. If the command code had been entered, then the partial command is executed using any arguments specified and assuming NULL arguments for the remainder. Careful termination of commands assists in the recovery from command specification errors.

OUTPUT RESPONSES

Output generating Plotter commands cause one or more output blocks to be created. The following paragraphs describe how the listening device should decode these output blocks to obtain the desired information.

Each output block setup by a Plotter command consists of two 16-bit data values and a TAG value (except for the READ STATUS command response in Command/Response Formats 5 and 6, which returns only one data value). For most commands, the two data values are obtained from encoding several pieces of requested information. The information can be decoded from the data values by reversing the encoding process. The encoding process for each output generating command is described in the command description sections. The meaning of the TAG value is explained for each output generating command in the command description sections.

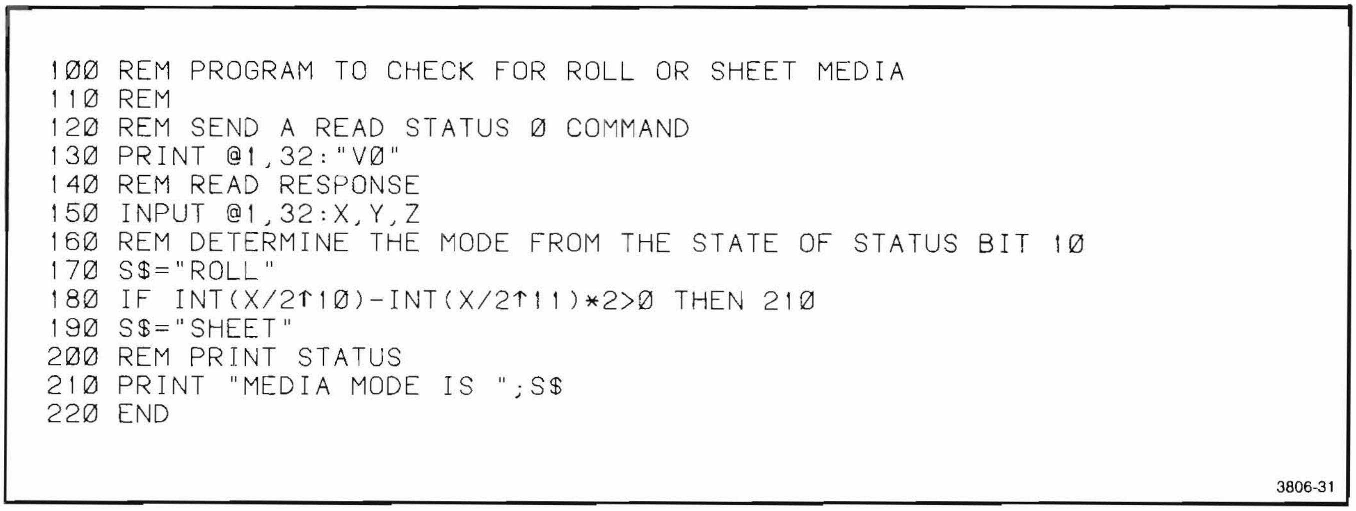
In forming output blocks, the two data values and the TAG value are converted to ASCII decimal numbers in scientific notation with each value separated by commas and then transmitted through the output buffer. Refer to *Plotter-To-Device Transmission*, earlier in this section for more information.

For example, Figure 5-7 shows a short program for a 4050 Series Graphic System that determines the media type status returned by a READ STATUS 0 command response. Figure 5-8 shows another example of how to decode an IDENTIFY command response. Both examples use Format 1.

The listening device can use the TAG value to determine when the last output block of a multiple block response has been received. In Format 1, a TAG value of 2 or 3 indicates that the block is the last block of the response.

For example, in a READ MACRO STATUS command, the number of blocks transmitted depends upon the number of macros currently defined. The final block of this response is identified by a particular value of the TAG number. (Refer to the READ MACRO STATUS command in Section 14 for more details on the response encoding.)

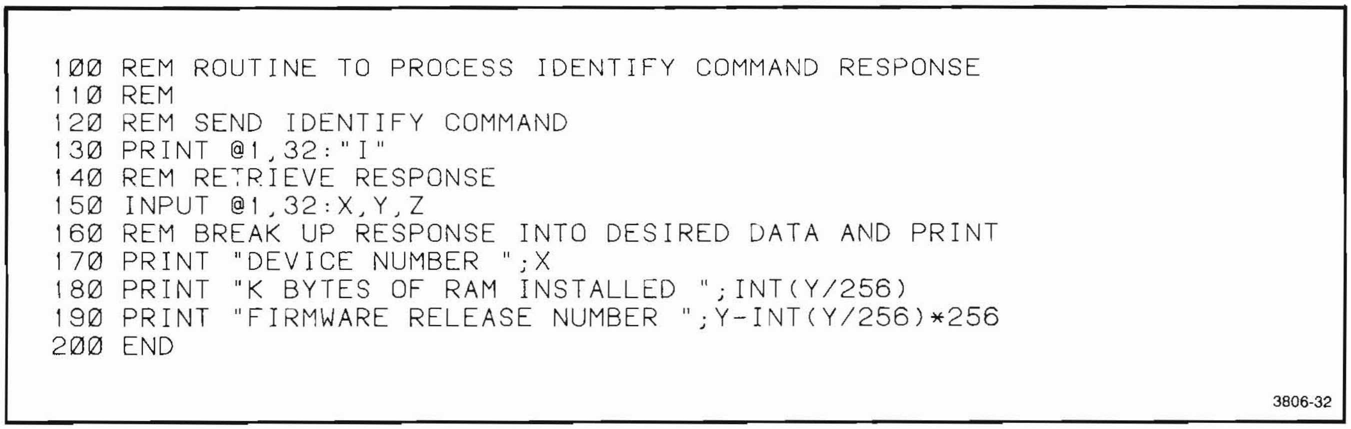
In Figures 5-7 and 5-8, the X, Y, and Z variables of the INPUT statements are the target variables whose value will be established by the three values contained in the response.



```
1ØØ REM PROGRAM TO CHECK FOR ROLL OR SHEET MEDIA
11Ø REM
12Ø REM SEND A READ STATUS Ø COMMAND
13Ø PRINT @1,32:"VØ"
14Ø REM READ RESPONSE
15Ø INPUT @1,32:X,Y,Z
16Ø REM DETERMINE THE MODE FROM THE STATE OF STATUS BIT 1Ø
17Ø S$="ROLL"
18Ø IF INT(X/2↑1Ø)-INT(X/2↑11)*2>Ø THEN 21Ø
19Ø S$="SHEET"
2ØØ REM PRINT STATUS
21Ø PRINT "MEDIA MODE IS ";S$
22Ø END
```

3806-31

Figure 5-7. GPIB Example Using READ STATUS Command Response.



```
1ØØ REM ROUTINE TO PROCESS IDENTIFY COMMAND RESPONSE
11Ø REM
12Ø REM SEND IDENTIFY COMMAND
13Ø PRINT @1,32:"I"
14Ø REM RETRIEVE RESPONSE
15Ø INPUT @1,32:X,Y,Z
16Ø REM BREAK UP RESPONSE INTO DESIRED DATA AND PRINT
17Ø PRINT "DEVICE NUMBER ";X
18Ø PRINT "K BYTES OF RAM INSTALLED ";INT(Y/256)
19Ø PRINT "FIRMWARE RELEASE NUMBER ";Y-INT(Y/256)*256
2ØØ END
```

3806-32

Figure 5-8. GPIB Example Using IDENTIFY Command Response.

CONTROLLING THE PLOTTER FROM A 4050 SERIES GRAPHIC SYSTEM

The following text uses the 4050 Series Graphic System to illustrate how a GPIB device is programmed to communicate with the Plotter. Three methods are shown: one byte at a time, DAB commands, and MSA commands. Although the examples are written using the 4050 Series BASIC language, most controllers have similar capabilities or functions. Refer to the programmer's manual for the specific GPIB device to determine how to implement these same functions. Refer to the following 4050 Series manuals:

- *4050 Series Graphic Computing System Operator's Manual*
- *PLOT 50 Introduction to Programming in BASIC*
- *4050 Series Graphic System Reference Manual*

In the following examples, the Plotter's device address is asssumed to be 1.

Plotter Communication One Byte at a Time

The following shows how a 4050 Series RByte or WByte statement can be used to command Move, Print, and Digitize operations by programming the communication one byte at a time. Although these functions, which transfer single bytes over the GPIB, are seldom used to control the Plotter from 4050 Series Controllers, they do serve to illustrate the GPIB actions required to communicate with the Plotter. Almost all GPIB devices have this capability and this approach to communication can almost always be used.

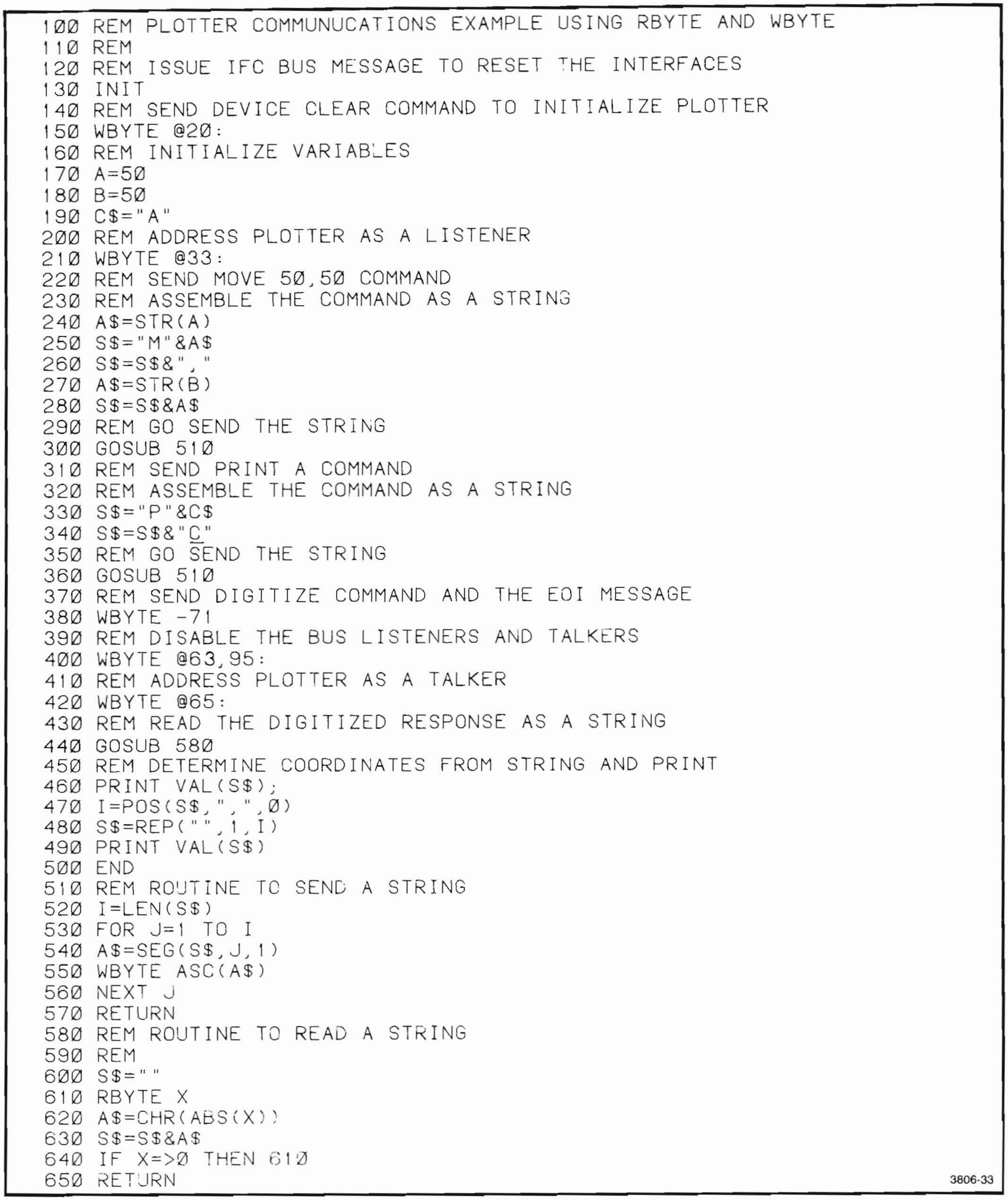
Figure 5-9 illustrates a program that sends commands in data byte (DAB) form to move the pen to 50,50, print an "A", and then digitize the pen position.

The two digit numbers in the WByte statement are the ASCII decimal equivalents (ADE's) of DABs to be transmitted over the GPIB. The ADE value for each DAB character was shown previously in Figure 5-3. The "@" character causes the GPIB's ATN bus message to be sent, and the ":" character causes the ATN message to be removed. For example, "WByte @35:" causes the MLA 1 bus message to be sent, which enables the Plotter (at Device Address 1) to be a listener.

NOTE

> *When the 4050 Series Graphic System issues a RByte or WByte statement, it automatically assigns itself as a listener or talker as required even though it is not explicitly enabled in the command.*

While the Plotter is enabled as a listener, the MOVE, PRINT, and DIGITIZE commands are transmitted. The Plotter assumes Command/Response Format 1, so the PRINT command is terminated with an $^{E}{}_{X}$ character. The negative value used to express the DIGITIZE command causes the EOI bus message to be sent with the "G" character to indicate end-of-transmission to the Plotter. This also terminates the DIGITIZE command and causes its execution. After the bus is cleared by the UNL message, the Plotter is enabled as a talker so it can transmit the digitize response. As the number of characters of the response is not predefined, the program reads response bytes until the Plotter sends the EOI message along with the last byte of the response (the EOI message is detected when the sign of X is minus). The desired coordinates are then extracted from the string of response bytes and printed.



```
100 REM PLOTTER COMMUNUCATIONS EXAMPLE USING RBYTE AND WBYTE
110 REM
120 REM ISSUE IFC BUS MESSAGE TO RESET THE INTERFACES
130 INIT
140 REM SEND DEVICE CLEAR COMMAND TO INITIALIZE PLOTTER
150 WBYTE @20:
160 REM INITIALIZE VARIABLES
170 A=50
180 B=50
190 C$="A"
200 REM ADDRESS PLOTTER AS A LISTENER
210 WBYTE @33:
220 REM SEND MOVE 50,50 COMMAND
230 REM ASSEMBLE THE COMMAND AS A STRING
240 A$=STR(A)
250 S$="M"&A$
260 S$=S$&","
270 A$=STR(B)
280 S$=S$&A$
290 REM GO SEND THE STRING
300 GOSUB 510
310 REM SEND PRINT A COMMAND
320 REM ASSEMBLE THE COMMAND AS A STRING
330 S$="P"&C$
340 S$=S$&"C"
350 REM GO SEND THE STRING
360 GOSUB 510
370 REM SEND DIGITIZE COMMAND AND THE EOI MESSAGE
380 WBYTE -71
390 REM DISABLE THE BUS LISTENERS AND TALKERS
400 WBYTE @63,95:
410 REM ADDRESS PLOTTER AS A TALKER
420 WBYTE @65:
430 REM READ THE DIGITIZED RESPONSE AS A STRING
440 GOSUB 580
450 REM DETERMINE COORDINATES FROM STRING AND PRINT
460 PRINT VAL(S$);
470 I=POS(S$,",",0)
480 S$=REP("",1,I)
490 PRINT VAL(S$)
500 END
510 REM ROUTINE TO SEND A STRING
520 I=LEN(S$)
530 FOR J=1 TO I
540 A$=SEG(S$,J,1)
550 WBYTE ASC(A$)
560 NEXT J
570 RETURN
580 REM ROUTINE TO READ A STRING
590 REM
600 S$=""
610 RBYTE X
620 A$=CHR(ABS(X))
630 S$=S$&A$
640 IF X=>0 THEN 610
650 RETURN
```

3806-33

Figure 5-9. GPIB Communications Example Using RByte and WByte to Send DAB Commands.

Plotter Communications Using DAB Commands

The following illustrates how higher level 4050 Series Basic commands control the Plotter more easily than with the RByte and WByte functions. The commands used in this section are in DAB form, which you can generate through most GPIB devices.

Figure 5-10 shows the same program as shown in Figure 5-9, except the PRINT command is used instead of RByte and WByte functions to send the commands. The "@1" characters in the statements direct the commands to GPIB Device 1 (the Plotter). The "32" entry tells the 4050 Series Graphic System not to use the MSA command form (described under the following heading). The Print statement automatically causes Device 1 (the Plotter) to be addressed as a listener before the data bytes are transmitted and the GPIB is cleared with the UNT and UNL messages after the bytes are transmitted. The ";" character (after the Print statement) suppresses a $^{C}{}_{R}$ character, which is otherwise always added to the bytes transmitted. The input command causes the Plotter to be addressed as a talker.

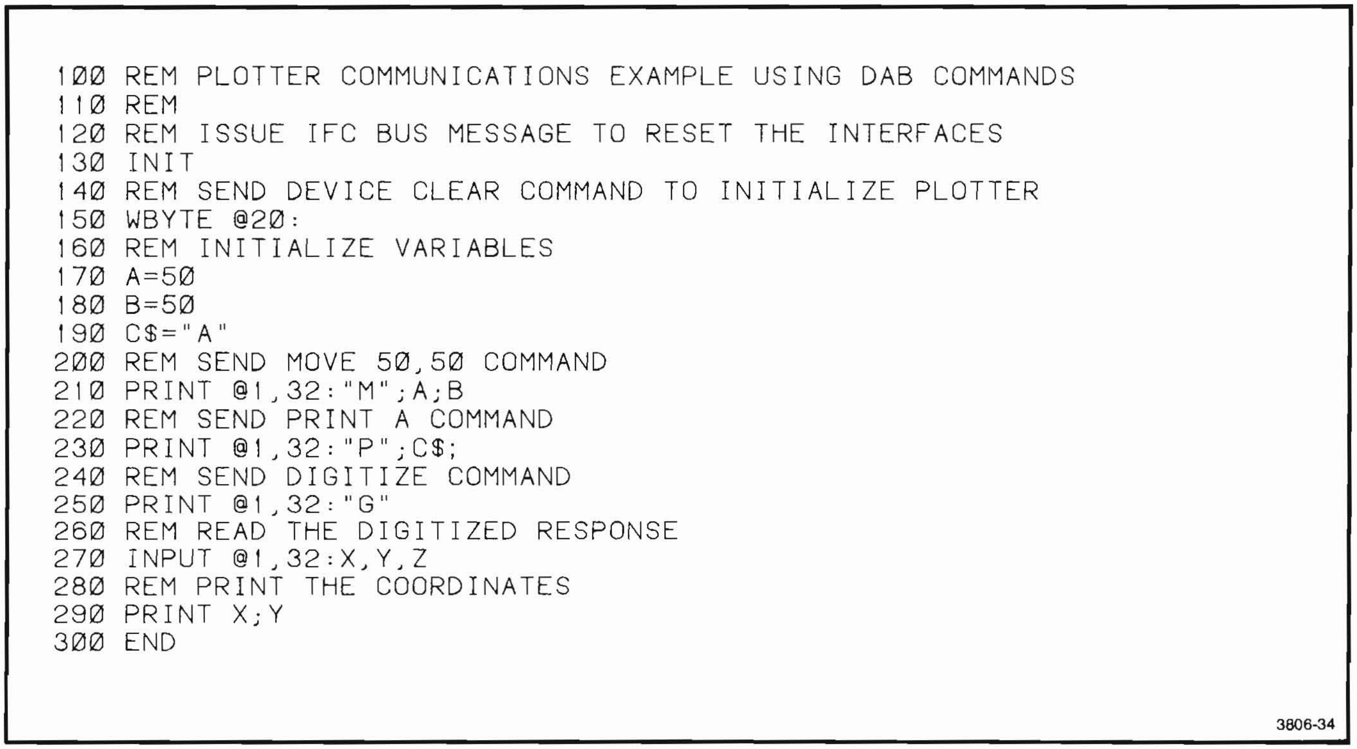
The semicolon used in place of a comma as an argument delimiter suppresses formatting spaces, which are otherwise automatically added between output variables (resulting in less efficient transmission to the Plotter).

Plotter commands can be expressed in BASIC language statements of many different forms. The following are all equivalent forms of the MOVE 50,50 command from Figure 5-10:

PRINT @1,32: "M50,50"

or

C$="M50$^{S}{}_{P}$50"
PRINT @1,32:C$



```
100 REM PLOTTER COMMUNICATIONS EXAMPLE USING DAB COMMANDS
110 REM
120 REM ISSUE IFC BUS MESSAGE TO RESET THE INTERFACES
130 INIT
140 REM SEND DEVICE CLEAR COMMAND TO INITIALIZE PLOTTER
150 WBYTE @20:
160 REM INITIALIZE VARIABLES
170 A=50
180 B=50
190 C$="A"
200 REM SEND MOVE 50,50 COMMAND
210 PRINT @1,32:"M";A;B
220 REM SEND PRINT A COMMAND
230 PRINT @1,32:"P";C$;
240 REM SEND DIGITIZE COMMAND
250 PRINT @1,32:"G"
260 REM READ THE DIGITIZED RESPONSE
270 INPUT @1,32:X,Y,Z
280 REM PRINT THE COORDINATES
290 PRINT X;Y
300 END
```

Figure 5-10. GPIB Communications Example Using PRINT to Send DAB Commands.

If the command is expressed as a string constant or string variable, the characters between the quote marks are transmitted to the Plotter as specified. In the first form, a comma is the argument separator. In the second form, a space is the argument separator. The command is terminated by the GPIB EOI message sent at the end of the PRINT command.

```
   PRINT @1,32:"M";50;50
or
   A=50
   B=50
   PRINT @1,32:"M";A;B
or
   A=50
   B=50
   C$="M"
   PRINT @1,32:C$;A;B
```

In these forms, the command code is expessed as a string constant or string variable. Arguments are numeric constants or numeric variables. The semicolon character suppresses formating spaces, which would otherwise be automatically inserted by the 4050 Series Graphic System before the numbers (resulting in extra characters to transmit). When a numeric value is transmitted, it is automatically preceded by a space character unless the value follows a string value. Thus, the command actually transmitted in all three cases is:

M50$^{S}_{P}$50

where the space is the argument separator. In the following command expression, the numeric variable B follows the string value C$, which causes the automatic space character to be suppressed:

```
   B=50
   C$="M50$^{S}_{P}$"
   PRINT @1,32:C$;B
```

Thus, the space used as the argument separator must be included in the string value.

In the following command expression, a string constant follows a numeric constant:

```
   B=50
   C$="$^{S}_{P}$50"
   PRINT @1,32:"M";B;C$
```

Again, the transmitted command in all three cases is:

M50$^{S}_{P}$50

Here, the space is also used as the argument separator and must be included in the string value.

NOTE

The comma or semicolon characters appearing as delimiters in BASIC program statements (and not within string definitions) are not transmitted to the Plotter. The argument separators and command terminators required for Plotter commands MUST be specified in addition to the delimiters required in the BASIC program statements.

Multiple Plotter commands can be combined on the same line, as shown in the following examples using the MOVE 50,50 and the PRINT A commands from Figure 5-10:

```
   PRINT @1,32:"M";A;B;"P";C$
or
   PRINT @1,32:"M50$^{S}_{P}$50PA"
```

These are only two of the several possible expressions of these commands. If the DIGITIZE command is combined with the previous two, the print string would have to be terminated with an $^{E}_{X}$ character as shown below (the previous example shows the EOI bus message terminating the print string argument and the command line):

```
   PRINT @1,32:"M50$^{S}_{P}$50PA$^{E}_{X}$G"
```

For clarity, these commands could be separated by a command terminator character, as shown next:

```
   PRINT @1,32:"M50$^{S}_{P}$50;PA$^{E}_{X}$;G"
```

NOTE

The single letter Plotter commands **A** *or* **C** *(or* **D** *if NULL arguments are used) should always be followed by a command terminator to ensure correct interpretation of the commands if another command follows on the same line.*

Plotter Communications Using MSA Commands

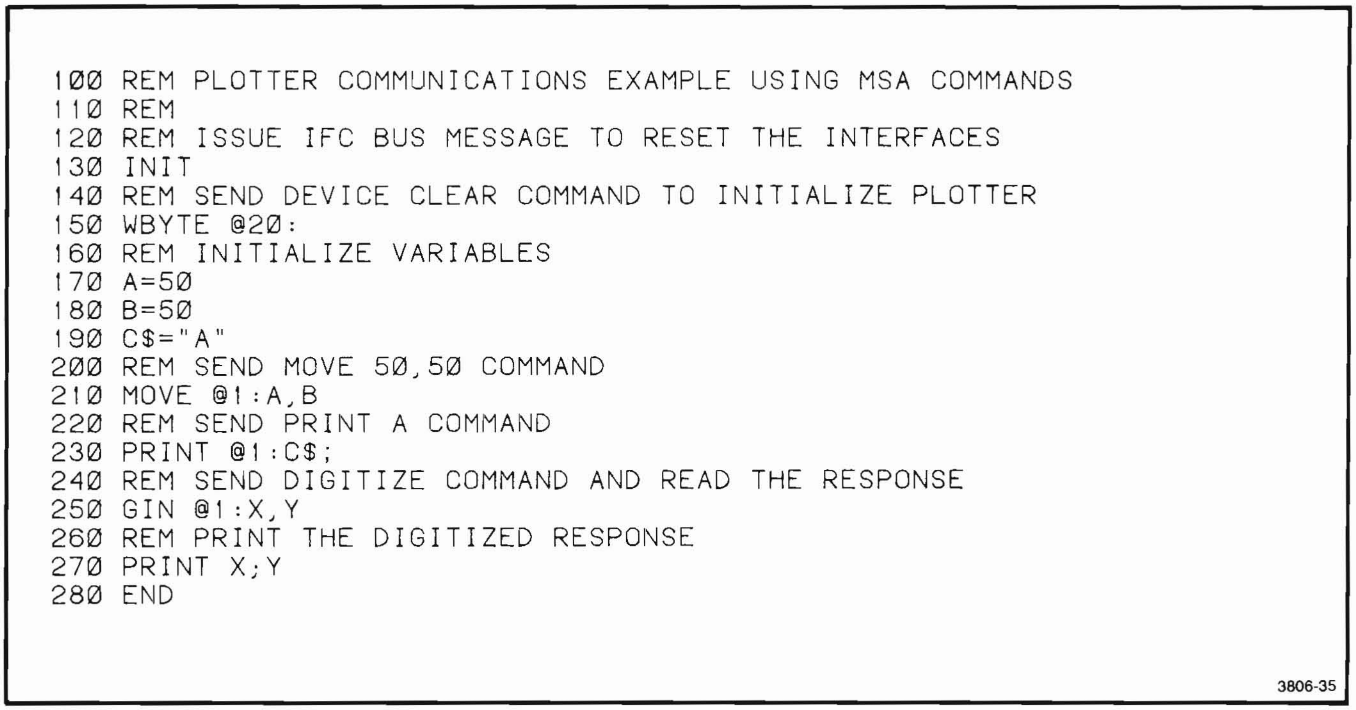
Figure 5-11 shows the same example used in the previous discussion using the MSA command forms. This example illustrates how 4050 Series BASIC statements can be used to generate the MSA form of the Plotter commands using the PRINT and INPUT Keywords. Refer to the command description sections or to Table 5-2 to determine if a MSA equivalent form is given for the desired Plotter command.

NOTE

This optional use of the GPIB MSA messages does not conform to the IEEE Standard 488-1978 definition of MSA messages. However, it does provide a convenient way of writing programs using functions common to both the Plotter and the display portion of the 4050 Series controllers. You can direct output to either the Plotter or the display by simply changing the device address from 1 = Plotter to 32 = display.

NOTE

The INPUT statement is not required in this form because the GIN statement automatically does an Input function.



```
100 REM PLOTTER COMMUNICATIONS EXAMPLE USING MSA COMMANDS
110 REM
120 REM ISSUE IFC BUS MESSAGE TO RESET THE INTERFACES
130 INIT
140 REM SEND DEVICE CLEAR COMMAND TO INITIALIZE PLOTTER
150 WBYTE @20:
160 REM INITIALIZE VARIABLES
170 A=50
180 B=50
190 C$="A"
200 REM SEND MOVE 50,50 COMMAND
210 MOVE @1:A,B
220 REM SEND PRINT A COMMAND
230 PRINT @1:C$;
240 REM SEND DIGITIZE COMMAND AND READ THE RESPONSE
250 GIN @1:X,Y
260 REM PRINT THE DIGITIZED RESPONSE
270 PRINT X;Y
280 END
```

3806-35

Figure 5-11. GPIB Communications Example for MSA Command Form.

The Plotter accepts the MSA commands shown in Table 5-2 for compatibility with 4050 Series Graphics Systems. All MSA commands not listed here are ignored. Table 5-2 also indicates any corresponding 4050 Series BASIC language keywords that are used to generate these commands.

Table 5-2

MSA COMMANDS

MSA	Equivalent 4663 Command	Corresponding 4050 Series Keyword
00 [c]	READ STATUS	[a]
07	RESET ALPHA PARAMETERS	[a]
08	PEN CHANGE	[a]
12	PRINT	PRINT
13 [c]	IDENTIFY (SIZE)	[a]
17	SET ALPHA SCALE	[a]
18	SELECT STANDARD ALPHA FONT	[a]
19	PRINT	LIST (TLIST)
20	DRAW	DRAW (RDRAW)
21	MOVE	MOVE (RMOVE)
22	MOVE TO HOME	PAGE
23	MOVE TO HOME	HOME
24	DIGITIZE	GIN
25	SET ALPHA ROTATION	[a]
26	PROMPT LIGHT	[a]
27 [c]	CALL DIGITIZE	[a]
None	None	AXIS [b]

[a] No corresponding 4050 Series keyword.

[b] The 4050 Series Graphic System sends MOVE and DRAW commands to the Plotter to draw the axis. No corresponding MSA command or 4663 AXIS command is generated.

[c] The INPUT keyword is used instead of the PRINT keyword to generate the Plotter command for these secondary addresses.

All commands listed in Table 5-2 can be generated in the MSA form as the commands shown in Figure 5-11 (for example, using the PRINT and INPUT keywords). Alternatively, the available keywords listed in Table 5-2 can be used to generate the commands. For example:

PRINT @1,20:"50,50"

or

DRAW @1:50,50

Both examples express a Plotter DRAW command. However, the 4050 Series Graphic System performs additional processing of commands produced with the MOVE, RMOVE, DRAW, and RDRAW keywords (which are not done if the commands are generated using the PRINT keyword). For example, the Window and Viewport established by the 4050 Series Graphic System will transform commands (containing MOVE and DRAW keywords) before they are sent to the Plotter. To avoid this transformation in the 4050 Series Graphic System, set the Graphic System Viewport and Window to the same values. (The Viewport and Window are automatically set to the same values upon power-up or when an INIT statement is executed.) Commands generated by 4050 Series BASIC statements that contain the MSA or DAB Plotter command forms will not be transformed before being sent to the Plotter.

If Plotter DRAW commands generated by DRAW or RDRAW keywords contain coordinates outside the 4050 Series Graphic System's Window, the coordinates will be "clipped". Thus, the 4050 Series Graphic System's Window should be set larger than any coordinates generated by DRAW or RDRAW keywords to avoid any "clipped" action.

Plotter MOVE or RMOVE commands (either generated by MOVE and RMOVE keywords or generated by MOVE and DRAW commands expressed in MSA or DAB form) are not "clipped" before transmission to the Plotter

NOTE

The clipping action of the 4050 Series Graphic System may give unexpected results if you generate graphics by commands expressed using a combination of Keyword, MSA, and DAB forms. Use only MOVE (RMOVE) and DRAW (RDRAW) keywords if you are using the clipping funtions in the 4050 Series Graphic System.

Section 6

INTRODUCTION TO COMMANDS AND ARGUMENTS

INTRODUCTION

This section covers the following topics, all related to the command description sections (Sections 7 through 16):

- Conventions and Notations
- Command Description Overview
- Syntax for Serial Interface Applications
- Syntax for GPIB Interface Applications
- Syntax for 4050 BASIC Applications (Using the GPIB Interface)

To begin, the programmer must be familiar with the manual conventions and notations presented at the beginning of the manual (before Section 1) and also following this introduction.

The user must also be familiar with the fundamental command structure and argument types of one application interface (either Serial, GPIB, or 4050 Series BASIC GPIB). For details refer back to either *Introduction to Serial Commands and Responses*, or *Introduction to GPIB Commands and Responses*, located in the previous two sections.

Once these prerequisites are achieved, the user can go directly to the desired command descriptions (using the *Command Description Overview* and specific interface applications described in this section).

CONVENTIONS AND NOTATIONS

The following summarizes the associated conventions and notations used throughout this manual.

CHARACTERS

ASCII Characters

The 94 printing ASCII characters (numbers, symbols, and uppercase and lowercase letters) are represented by their normal symbols.

The ASCII Control, SPACE, and DELETE characters are each represented by an appropriate single symbol. See the ASCII Code Chart appendix.

NOTE

> *The ASCII SPACE character is always shown as* SP*, and not as a blank space between printed characters.*

Examples: **& 1 2 A B a b EC SP DT**

The last three ASCII characters are ESCAPE, SPACE, and DELETE.

Special Characters

Two symbols are replaced by certain single ASCII characters. The two symbols, AT and AD, represent the attention and address characters, respectively, and are replaced as follows:

- AT — attention character, replaceable by

  EC ! Λ or **SY**

- AD — address character, replaceable by

  A B C D E F G or **H**

LITERAL VS. VARIABLE ELEMENTS

Boldface Type

A literal element, which is entered exactly as shown, is shown in bold.

Regular Type

A variable element, which is replaced by appropriate specific information, is shown in regular type; a single element can be represented by one symbol, one word, or words connected by hyphens.

Examples

AT AD **R** rotation-angle

where three elements are replaced and R is entered exactly as shown.

! A R 45

where five ASCII characters are entered exactly as shown.

COMMAND ELEMENTS

Element Types

The last words of a 4663 Plotter variable element name indicates the element type required by the 4663, as shown below:

Last Words	**Element Type**
selector	selector
string	string
print-string	print string
pa	primary address
target-variable	target variable

Other endings imply a numeric element.

Examples:

- axis-selector is a selector element
- print-character-string is a string element
- radius is a numeric element

Omitted Numeric Elements

When a numeric element is omitted, a value of zero is assumed. Any exceptions are noted where they occur.

Argument Separators

A nonbold comma (,) between elements can be replaced by either a bold comma (**,**) or one or more SP characters. A nonbold semicolon (;) between variable elements can be replaced by either a bold semicolon (**;**) or a bold comma (**,**). Exceptions are noted where they apply.

Command Terminators

Command terminators are generally NOT shown in the individual command descriptions. Specific termination varies with the interface and the Command/Response Format used. Refer to Sections 4 and 5 for instructions on how to terminate commands.

NOTATIONS

Brackets [ ]

An element inside brackets is *optional*. Stacked elements within brackets indicate selection of one or none of the elements.

Example: [**0**] select **0**, **1**, or neither
[**1**]

Braces { }

When multiple elements are stacked within braces, one element *must* be selected.

Example: {**0**}
{**1**} select **0**, **1**, or **2**
{**2**}

Dots . . .

Three dots (ellipsis) indicate that a previous element *may* be repeated.

Example: SP . . . one or more SP characters

Indented for Continuation

If a command is continued on to the next printed line, the additional line is indented.

Example: AT AD **X** x-value,y-value[,x-value,
y-value] . . .

COMMAND DESCRIPTION OVERVIEW

Each command section is divided into two parts: general concepts and specific command descriptions. Each command description follows a general format and is comprised of all or some of the following:

- Purpose
- Syntax Boxes
  There are up to three applications:
  - Serial
  - GPIB
  - 4050 BASIC
- For 4050 BASIC, there are up to three forms:
  - Data Byte Form
  - Secondary Address Form
  - Special Keyword Form
- Parameters
  Includes restrictions, meanings assigned to selector arguments, etc.
- Outputs
- Comments
- Errors
  Includes error messages, causes of errors, and Plotter responses.
- Examples

For each of the interface applications, the syntax boxes show a a recommended general command form that can be used under most conditions. All syntax boxes (with the exception of the general GPIB form) include specific examples of the commands. The contents of the syntax boxes and related considerations are discussed separately for each of the following interface applications: Serial Interface, GPIB Interface, and 4050 Series BASIC GPIB.

SYNTAX FOR SERIAL INTERFACE APPLICATIONS

Plotter commands are transmitted via the Serial interface as a series of ASCII characters. The general syntax form and sample are shown for each Style I command in the first syntax box for each command description. (Refer to Section 4 for Style II command information.)

Refer to Figure 6-1. Each Style I command consists of an attention character, an address character, a one- or two-character command code and, for many commands, one or more arguments and argument separators. For Command/Response Formats 1 or 2 only, the command may include an optional command termination character (either ᶜR or ;). Unless otherwise noted, the information in a Serial syntax box applies for Command/Response Formats 1-4.

NOTE

Some commands request information from the Plotter. In these cases, the syntax box contains either One Output Message Block Produced or Two Output Message Blocks Produced; for the READ MACRO STATUS Command the syntax box contains the statement One Output Message Block Per Macro. For these commands, additional control procedures are involved.

ATTENTION CHARACTER

This character is to be replaced by the attention character selected on the Parameter Entry Card. The choices are: ^{E}C, !, Λ, or ^{S}Y.

ADDRESS CHARACTER

This character is to be replaced by the address character selected on the Parameter Entry Card. The choices are **A, B, C, D, E, F, G,** or **H**.

COMMAND CODE

The command code consists of one or two characters. Letter characters may be either uppercase or lowercase.

ARGUMENTS

Argument names identify the argument and the argument type. See *Introduction To Serial Commands And Responses* in Section 4.

ARGUMENT SEPARATORS

Each nonbold comma (,) shown as an argument separator is replaced by a bold comma (**,**) or by one or more ^{S}P characters. Any exceptions are noted where they apply.

COMMAND TERMINATION

Command termination occurs either when the next command's attention character is received; in Command/Response Formats 3 or 4 only, termination occurs immediately for commands without arguments. For Command/Response Formats 1 and 2, an optional command terminator character (either ^{C}R or ;) may be used to terminate commands.

SAMPLE

The syntax sample shows one valid command of the general form presented, where the attention character is ^{E}C and the address character is **A**.

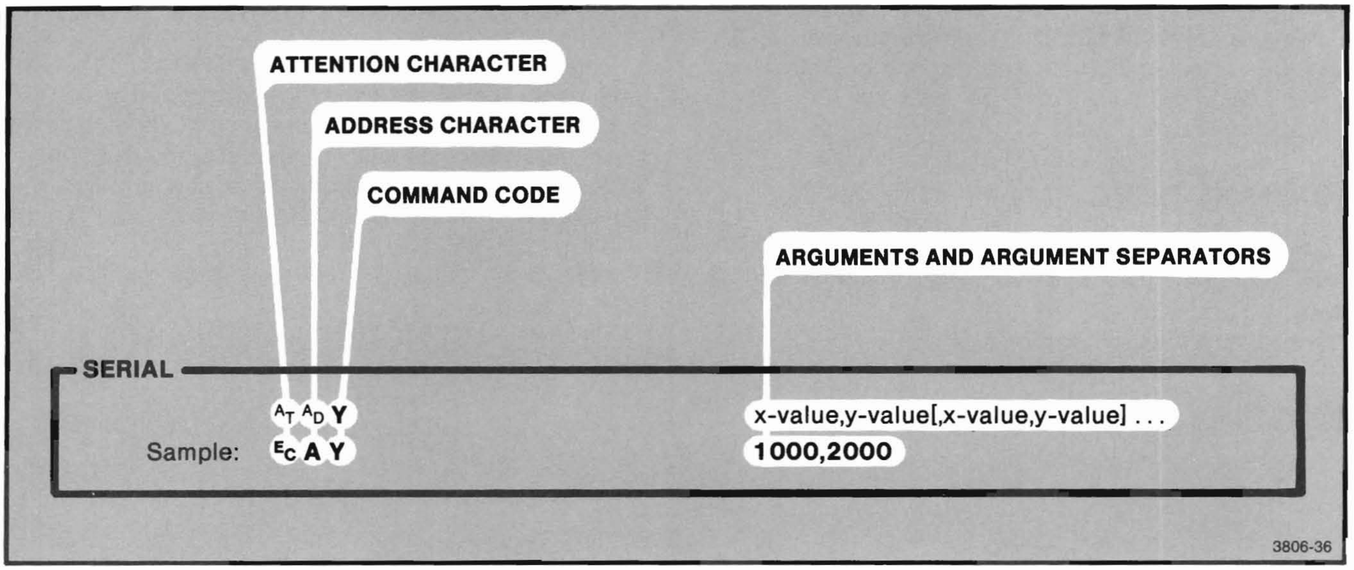


Figure 6-1. A Typical Command Form and Sample for the Serial Interface.

SYNTAX FOR GPIB INTERFACE APPLICATIONS

NOTE

If you are using 4050 Series BASIC GPIB, skip to the Syntax for 4050 Series BASIC GPIB Applications heading.

Plotter commands are transmitted over the GPIB bus as a series of ASCII characters, accompanied by control information. One or more commands may be included in one GPIB transmission. The general command form in the second syntax box shows the command code and arguments only. See Figure 6-2. Associated control information for GPIB communication is not shown. (Refer to the GPIB Interface section for more information.) Unless otherwise noted, the general form and sample applies for Command/Response Formats 1 through 6.

NOTE

Some commands request information from the Plotter. In these cases, the syntax box contains either One Output Message Block Produced, or Two Output Message Blocks Produced; for the READ STATUS MACRO command, the syntax box contains the statement One Output Message Block Per Macro. For these commands, additional control procedures are involved (see the GPIB Interface section, Section 5).

COMMAND CODE

The command code consists of one or two characters. Letter characters may be either uppercase or lowercase.

NOTE

For some commands, a GPIB Secondary Address may be transmitted instead of the command code. For these commands the 4050 BASIC Secondary Address form shows which secondary address is assigned for the command.

ARGUMENTS

Argument names identify the argument and the argument type. See *Introduction To GPIB Commands And Responses* in Section 5 for argument type explanations.

ARGUMENT SEPARATORS

Each nonbold comma (,) shown as an argument separator is replaced by a bold comma (**,**) or by one or more ^{S}P characters. Specific exceptions are noted where they apply.

COMMAND TERMINATION (NOT SHOWN)

A command may be terminated by a semicolon (;) termination character (if several commands are sent at once). For Command/Response Formats 5 and 6, the command is automatically terminated at the end of a GPIB transmission. For specific details on how to terminate commands, refer to Section 5.

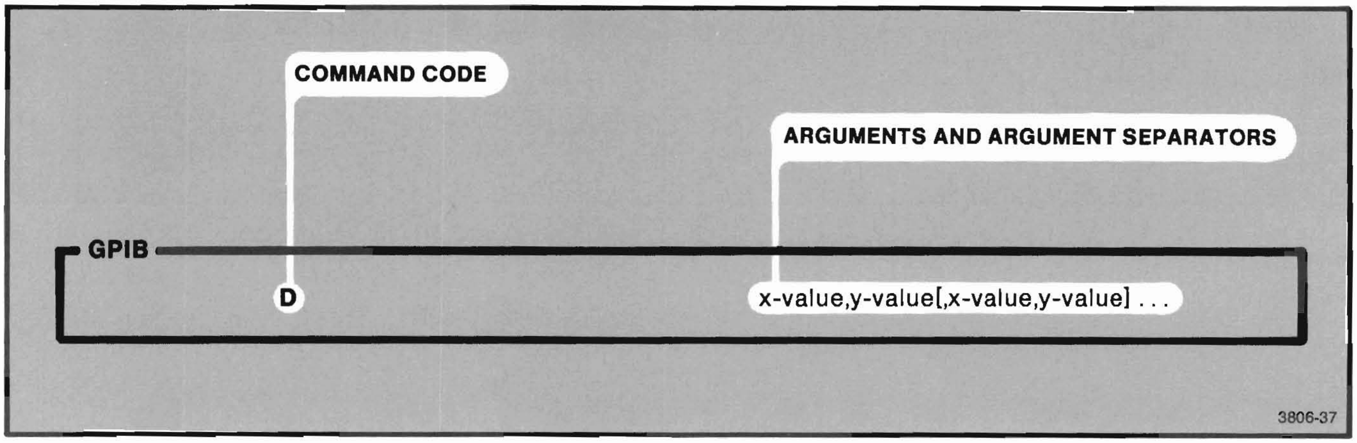

COMMAND CODE
ARGUMENTS AND ARGUMENT SEPARATORS
GPIB
D
x-value,y-value[,x-value,y-value] . . .
3806-37

Figure 6-2. A Typical Command Form for GPIB.

SYNTAX FOR 4050 SERIES BASIC GPIB APPLICATIONS

In the third syntax box for each command description, up to three forms of 4050 BASIC statements are available for each command. Each general form illustrates one way to generate a correct Plotter command. (No attempt is made to show all possible ways of generating Plotter arguments using BASIC.) Unless otherwise noted, information applies for Command/ Response Formats 1 through 6. Refer to Figure 6-3.

PRIMARY ADDRESS AND SECONDARY ADDRESS

As shown in Figure 6-3, all three forms require a GPIB primary address value that matches the selection made on the GPIB device address line of the Parameter Entry card. One form requires the specific GPIB secondary address value assigned to a specific command. A primary address or secondary address value may be represented within a BASIC statement by any valid BASIC numerical expression. For example, a primary address of one may be represented by 1 or by D where D = 1.

COMMAND CODE

Each command requires either a command code (one or two characters) or the equivalent secondary address assigned to that command.

ARGUMENTS AND ARGUMENT SEPARATORS

For Plotter commands that can have arguments, all three forms show how to generate Plotter arguments and separators within a BASIC statement. Each Plotter selector or numeric argument may be generated by a corresponding BASIC numeric expression; the BASIC statement produces the corresponding Plotter argument (and argument separators as needed). Also, one or more Plotter arguments (and argument separators) may be generated by a BASIC string variable or string constant if care is taken to include each required Plotter argument separator.

NOTE

Whenever a BASIC string-constant or string-variable used for the first of two Plotter numeric arguments is followed by a semicolon (;) and a numeric expression, the automatic $^{S}{}_{P}$ before the second argument is suppressed. The Plotter requires a separator; for example, either a $^{S}{}_{P}$ or a comma (,). Programming actions that avoid the problem are:

- *Use a comma (,) instead of a semicolon (;) between the string constant or string variable and the following numeric expression.*
- *Include a Plotter Argument Separator character (such as a comma) as the last character of the BASIC string-constant or string-variable.*

You must use a comma (,) instead of a semicolon (;) between the string-constant or string-variable and the following argument if the following argument is a 4050 BASIC array variable.

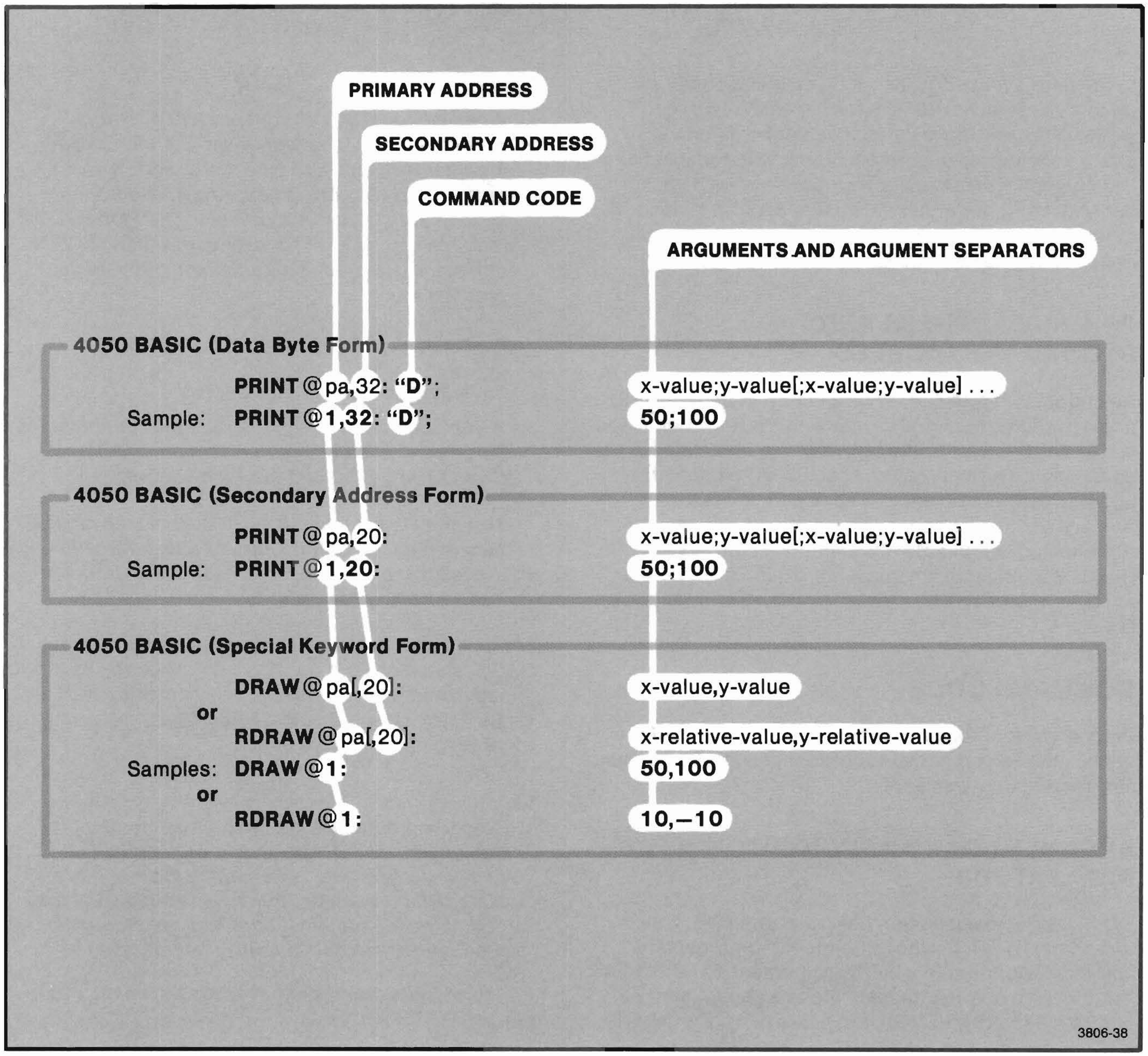
For commands that require a print-string argument, the BASIC statement can contain a string-constant, string-variable, numeric expression, or a combination of all three. The Plotter prints the actual character string that the BASIC statement produces.

NOTE

Do not forget to include the print string termination character ($^{E}{}_{X}$) whenever a print string is followed by another command in the same BASIC statement. Remember that a semicolon (;) at the end of the statement suppresses the automatic $^{C}{}_{R}$ produced at the end of a BASIC print statement.

For commands that require a string argument, you must either:

- Produce a delimited string that begins and ends with the proper delimiter character (quotation mark or apostrophe).
- Produce an undelimited string that begins with a valid character (such as an uppercase or lowercase letter) and ends with a proper terminator (the $^{E}{}_{X}$ termination character is required for most applications).



4050 BASIC (Data Byte Form)

```
         PRINT @pa,32: "D";         x-value;y-value[;x-value;y-value] . . .
Sample:  PRINT @1,32: "D";          50;100
```

4050 BASIC (Secondary Address Form)

```
         PRINT @pa,20:              x-value;y-value[;x-value;y-value] . . .
Sample:  PRINT @1,20:               50;100
```

4050 BASIC (Special Keyword Form)

```
          DRAW @pa[,20]:            x-value,y-value
      or
          RDRAW @pa[,20]:           x-relative-value,y-relative-value
Samples:  DRAW @1:                  50,100
      or
          RDRAW @1:                 10,—10
```

Figure 6-3. The Three Command Forms for 4050 BASIC Applications.

Section 7

SERIAL INTERFACE COMMANDS

INTRODUCTION

Serial interface commands control the communication between a user's host computer or other devices and the Plotter. Before using these commands, please read Section 4 of this manual.

COMMAND DESCRIPTIONS

Because these commands deal only with Serial interface operation, the GPIB and 4050 Series GPIB syntax boxes are not included for the command descriptions. There are twelve commands described in this section. These are:

- DEVICE ON
- DEVICE OFF
- BLOCK START
- BLOCK END
- SET TURNAROUND DELAY
- SET BLOCK SIZE
- SET BYPASS CANCEL CHARACTER
- SET SIGNATURE CHARACTER
- SET PROMPT STRING
- INTERFACE PARAMETER RESET
- DATA RESET
- SELECT COMMAND/RESPONSE FORMAT

Each command is described separately in the paragraphs that follow.

DEVICE ON

Purpose: This command turns the Plotter to the logical on state.

SERIAL

A_T A_D **E**

Sample: E_C **A E**

Comments

After a DEVICE ON command, the Plotter is in the logical on state and will interpret all characters received as commands. The AUTO MUTE function is enabled if it has been chosen on the Parameter Entry Card. This means that in a loop-thru configuration, received characters beginning with the command character **E** will not be sent to the terminal. This condition remains in effect until a DEVICE OFF command is received.

A BLOCK START command also turns the Plotter to the logical on state.

NOTE

At power-up or after a front panel Initialize is done, the Plotter is in the logical off state. An exception to this rule occurs if an Initial Command/Response Format of 3 or 4 is used and an Initial Aspect Ratio of 4X:3Y is selected on the Parameter Entry Card. Then the Plotter will be logically on at power-up or after a front panel Initialize. This is emulation of the TEKTRONIX 4662's Copy Mode feature. A DEVICE OFF command may be used in this mode to turn the Plotter to the logical off state.

DEVICE OFF

Purpose: This command turns the Plotter to the logical off state.

SERIAL

$^{A}{}_{T}$ $^{A}{}_{D}$ **F**

Sample: $^{E}{}_{C}$ **A F**

Comments

After a DEVICE OFF command, the Plotter is in the logical off state and all received characters are *not* interpreted as commands (until either a DEVICE ON or BLOCK START command is recognized). The AUTO MUTE selection on the Parameter Entry Card is disabled. The terminal then receives all characters sent by the host computer.

With AUTO MUTE selected at higher communication rates (more than 1200 baud), a few characters immediately following a DEVICE OFF command may be garbled when received by a loop-thru terminal. Refer to Section 4 for more information.

The Plotter is logically off at power-up or after a front panel Initialize, except as noted in the comments under the DEVICE ON command.

NOTE

In Command/Response Formats 1 and 2, the DEVICE OFF command will not be processed until a terminator character is received.

BLOCK START

Purpose: This command is used to begin a block in Block Mode communications, and also turns the Plotter to the logical on state (this action is equivalent to a DEVICE ON command).

SERIAL

$^{A}{}_{T}$ $^{A}{}_{D}$ **(**

Sample: $^{E}{}_{C}$**A(**

Comments

A BLOCK START command initiates Block Mode communication. The Serial interface starts a checksum calculation beginning with the “(” command code character and ending with the “)‘ character of the BLOCK END command.

References

A complete description of Block Mode communications is given in Section 4 of this manual.

BLOCK END

Purpose: This command ends a block in Block Mode communications. The Plotter sends an **A** (or an **I**) indicating that the block has been received correctly (or incorrectly) and that the Plotter is ready to accept another block of data.

SERIAL

	$^{A}{}_{T}$ $^{A}{}_{D}$ **)**	checksum-value
Sample:	$^{E}{}_{C}$ **A)**	**373**

Parameters

checksum-value

Block checksum value, 0-4095. Special value: 0 means no checksum verification, positive block acknowledge response generated

Outputs

one character (plus signature character and output termination character(s) as selected)

Block acknowledge responses, **A** or **I**:

- **A** = positive block acknowledge (block is accepted, ready for next block)
- **I** = negative block acknowledge (block is rejected, retransmit last block)

Comments

The checksum is the sum of the ASCII decimal equivalent (ADE) values of the characters contained in the block. The sum begins with the "**(**" character of the BLOCK START command and ends with the "**)**" character of the BLOCK END command. The rules for calculation of checksums are presented in Section 4.

References

See Section 4 for complete information on Block Mode communication.

Errors

Error 9
(Block Checksum Error)

Error 44
(Integer Outside Legal Range) Values greater than 4095 result in rejection of the block; negative values are taken as zero (block is accepted).

SET TURNAROUND DELAY

Purpose: This command sets the turnaround delay time.

SERIAL

	$^{A}{}_{T}$ $^{A}{}_{D}$ **G**	delay-time
Sample:	$^{E}{}_{C}$ **A G**	**200**

Parameters

delay-time
Delay time, in milliseconds, 0-65535. Resolution is approximately 8 ms.

Comments

Some host computers cannot accept input from a device immediately after sending information to it. A time delay called the turnaround delay must occur or data is lost. This command forces the Plotter to wait the requested time delay before transmitting to the host computer. The same delay time is also used between Plotter output messages.

The default value for the delay-time is zero. This is the value at either power-up or after an INTERFACE PARAMETER RESET command is received.

References

See *Communication With Other Devices* in Section 4 for a complete discussion of Plotter to host communication.

Errors

Error 44
(Integer Outside Legal Range) Values less than zero are taken as zero; values greater than 65535 are taken as 65535.

SET BLOCK SIZE

Purpose: This command sets the maximum size of blocks that will be sent during Block Mode communications.

SERIAL

	${}^{A}{}_{T}$ ${}^{A}{}_{D}$ **H**	maximum-block-size
Sample:	${}^{E}{}_{C}$ **A H**	**132**

Parameters

maximum-block-size
Maximum number of characters that will be sent in a block, 0-65535.

Comments

The maximum-block-size value should include room for the BLOCK START and BLOCK END commands (including checksum and terminator characters).

The default value for block size is zero. This is the value at power-up or after the INTERFACE PARAMETER RESET command is received.

References

See Section 4 for a discussion of Block Mode communications and recommendations for block size values.

Errors

Error 8
(Block Size Exceeds Input Buffer Size)

SET BYPASS CANCEL CHARACTER

Purpose: This command sets the bypass cancel character.

SERIAL

	$^{\mathrm{A}}{}_{\mathrm{T}}$ $^{\mathrm{A}}{}_{\mathrm{D}}$**U**	bypass-cancel-character-string
Sample:	$^{\mathrm{E}}{}_{\mathrm{C}}$ **A U**	$^{\mathrm{L}}{}_{\mathrm{F}}$

Parameters

bypass-cancel-character-string
Represents the single character to be used as the bypass cancel character. An omitted argument or the character $^{\mathrm{N}}{}_{\mathrm{U}}$ (ADE 0) disables this function.

Comments

Output messages from the Plotter may be echoed (returned to the Plotter) by the host computer. When the correct bypass cancel character is selected, the Plotter ignores the host computer echo.

The bypass cancel character selected should be the last character echoed from a Plotter output message. For example, when the Plotter output terminator is $^{\mathrm{C}}{}_{\mathrm{R}}$ (selected on the Parameter Entry Card), and the host computer echoes $^{\mathrm{C}}{}_{\mathrm{R}}$ with $^{\mathrm{C}}{}_{\mathrm{R}}$ $^{\mathrm{L}}{}_{\mathrm{F}}$, the bypass cancel character should be set to $^{\mathrm{L}}{}_{\mathrm{F}}$.

The default value for the bypass cancel character is $^{\mathrm{N}}{}_{\mathrm{U}}$. This is the value at power-up or after an INTERFACE PARAMETER RESET command is received.

References

See Section 4 for a description of string argument forms.

Errors

Error 45
(Too Many Entries or String Too Long) The first character becomes the bypass cancel character.

Examples

The sample shown in the syntax box sets the bypass cancel character to $^{\mathrm{L}}{}_{\mathrm{F}}$.

SET SIGNATURE CHARACTER

Purpose: This command sets the signature character that the Plotter prefixes to each output block.

SERIAL

	$^{A}\mathrm{T}$ $^{A}\mathrm{D}$ **S**	signature-character-string
Sample:	$^{E}\mathrm{C}$ **A S**	**a**

Parameters

signature-character-string
Represents the single character to be used as the signature character. An omitted argument or the character $^{N}\mathrm{U}$ (ADE 0) disables this function.

Comments

This command causes output blocks from the Plotter to be prefixed with the specified signature character. This character can be used to identify the specific Plotter originating an output message when more than one plotter is used in the same loop-thru configuration.

The default (power-up) value of the signature character is $^{N}\mathrm{U}$ (no signature character). The INTERFACE PARAMETER RESET command also resets the signature character to $^{N}\mathrm{U}$.

References

See Section 4 for a description of string arguments.

Errors

Error 45
(Too Many Entries or String Too Long) The first character becomes the signature character.

SET PROMPT STRING

Purpose: This command sets the prompt character string, which the Plotter must receive before any output message is sent to the host computer.

SERIAL

	${}^{A}{}_{T}$ ${}^{A}{}_{D}$ **R**	prompt-character-string
Sample:	${}^{E}{}_{C}$ **A R**	**"?${}^{S}{}_{P}$"**

Parameters

prompt-character-string
Represents the prompt character string, 0-4 characters; omitted argument or ${}^{N}{}_{U}$ turns off Prompt Mode.

Comments

Some host computers can accept input from a device, such as a terminal or Plotter, only at certain times. A prompt string (e.g., "?") is sent to a device to indicate input is expected (and will be accepted). The SET PROMPT STRING command causes the Plotter not to transmit output messages to the host computer until the specified prompt string is received. A prompt string must be received for each output message.

The default value of the prompt-string is ${}^{N}{}_{U}$. This is the value at power-up or after the INTERFACE PARAMETER RESET command is received. A prompt-string of ${}^{N}{}_{U}$ indicates that Prompt Mode output protocol is not enabled.

References

See Section 4 for a complete description of Prompt Mode and string argument forms.

Errors

Error 45
(Too Many Entries or String Too Long) The first four characters are used.

INTERFACE PARAMETER RESET

Purpose: This command resets interface parameters to default (power-up) values.

SERIAL

$^{A}_{T}$ $^{A}_{D}$ **CR**

Sample: $^{E}_{C}$ **A CR**

Comments

This command resets the following interface parameters to default values:

Block Size	to	0
Turnaround Delay	to	0
Bypass Cancel Character	to	$^{N}_{U}$
Signature Character	to	$^{N}_{U}$
Prompt String	to	$^{N}_{U}$

Additional effects of this command are:

- The Command/Response Format is reset to the value selected on the Parameter Entry Card.
- The type of Device Units is reset to ADUs (Addressable Device Units).
- The Point light is turned off and blinking is stopped. The light will be on steadily in Command/Response Formats 3 and 4.
- The Operator Digitize enable state, if enabled by the OPERATOR DIGITIZE command, is canceled. No last output block is sent by the Plotter. For Command/Response Formats 3 and 4, the permanently enabled Digitize state is entered.

References

See Section 13 for more information.

DATA RESET

Purpose: This command deletes commands from the input buffer that have not been executed. Later commands are not affected.

SERIAL

${}^{A}{}_{T}$ ${}^{A}{}_{D}$ **CD**

Sample: ${}^{E}{}_{C}$ **A CD**

Comments

This command should be used carefully. Any commands that have already been read from the interface input buffer (and are being processed by the Plotter) will be executed. Other stored commands are deleted.

SELECT COMMAND/RESPONSE FORMAT

Purpose: This command selects a Command/Response Format, overriding the selection made on the Initial Command/Response Format line of the Parameter Entry Card.

SERIAL

	${}^{A}{}_{T}$ ${}^{A}{}_{D}$ CC	format-selector
Sample:	${}^{E}{}_{C}$ **A CC**	**2**

Parameters

format-selector
Command/Response Format, 1-4

Comments

The Command/Response Format determines the argument and command terminator syntax expected for input commands and the type and form of output messages that can be generated.

These additional effects result from this command:

- The type of Device Units is reset to ADUs.

  NOTE

  If the SET WINDOW command is used when the current Device Units are not ADUs, this command changes the Window's numerical range. To restore the Window to the desired values, either reselect the type of Device Units type used with the original SET WINDOW command or redefine the window using the SET WINDOW command. See Section 11 for more information.

- The Point (Prompt) light is turned off, except in Command/Response Formats 3 and 4, when the Point light is on steadily. The Plotter stops scanning for the prompt string (if one is defined).
- The Operator Digitize enable state (if enabled by the OPERATOR DIGITIZE command) is canceled. No last output block is sent by the Plotter. In Command/Response Formats 3 and 4, the Plotter enters the permanently enabled Digitize state. See Section 13 for more information.

References

See *Setting Up the Parameter Entry Card* in Section 4 for a complete description of Command/Response Formats.

Section 8

GPIB INTERFACE COMMANDS

INTRODUCTION

GPIB interface commands affect the form and type of communication between the Plotter and the controller (such as the TEKTRONIX 4050 Series Graphics Computing Systems).

Definition of GPIB characteristics includes certain signals and commands that are recognized by the Plotter. These include the Interface Clear signal and four GPIB commands: DEVICE CLEAR, SELECTED DEVICE CLEAR, SERIAL POLL ENABLE, and DISABLE.

The use of these commands is different from the Plotter commands presented in this section. Usually, the controller has a keyword command that generates these signals or commands. The controller's manual should be consulted for what commands are available and how to use them. The GPIB Interface section of this manual, Section 5, describes the functions supported by the Plotter.

COMMAND DESCRIPTIONS

There are three commands described in this section:

- DATA RESET
- SELECT COMMAND/RESPONSE FORMAT
- INTERFACE PARAMETER RESET

The following pages describe each command separately.

Because the commands in this section deal only with GPIB interface operation, the Serial syntax box is not included for these command descriptions.

DATA RESET

Purpose: This command deletes previous commands sent to the Plotter that have not yet been processed. Subsequent commands are not affected.

GPIB

```
CD
```

4050 BASIC (Data Byte Form)

```
         PRINT @pa,32: "CD"
Sample:  PRINT @1,32: "CD"
```

Comments

This command should be used carefully. Any commands that have already been read from the interface input buffer (and are being processed by the Plotter) will be executed. Other stored commands are deleted.

SELECT COMMAND/RESPONSE FORMAT

Purpose: This command selects a Command/Response Format, overriding the selection made on the Initial Command/Response Format line of the Parameter Entry Card.

GPIB

	CC	format-selector

4050 BASIC (Data Byte Form)

	PRINT @pa,32: **"CC"**;	format-selector
Sample:	**PRINT @1,32: "CC";**	**3**

Parameters

format-selector
Command/Response Format selector, 1-7

Comments

The Command/Response Format determines the type of command and argument syntax expected, as well as the form and nature of output messages.

This command performs these additional actions:

- The type of Device Units is set to GDUs (Graphic Device Units).
- The Operator Digitize enable state, if enabled by the OPERATOR DIGITIZE command, is canceled. No last output block is sent by the Plotter. For GPIB Command/Response Formats 5 and 6, the permanently enabled Digitize state is entered. See Section 13 for more information.
- The Point light is turned off, except in Command/Response Formats 5 and 6 when it is on steadily.

The following is a brief description of the available GPIB Command/Response Formats:

- Formats 1, 3, and 5 are compatible with the TEKTRONIX 4050 Series desk-top computing systems. Formats 2, 4, and 6 are compatible with Hewlett-Packard desk-top calculators, such as the HP 9845.
- Formats 5 and 6 are compatible with the TEKTRONIX 4662 Plotter. However, there are some differences that are noted in Section 5.
- Format 7 is a special format that accepts Serial interface commands but does not allow any output to be generated.

References

A complete description of GPIB Command/Response Formats is given in the Section 5.

INTERFACE PARAMETER RESET

Purpose: This command resets both the GPIB dependent parameters and states to the default (power-up) values.

GPIB

```
              CR
```

4050 BASIC (Data Byte Form)

```
              PRINT @pa,32: "CR"
     Sample:  PRINT @1,32: "CR"
```

Comments

This command performs the following actions:

- The Command/Response Format is reset to the value selected on the Initial Command/Response Format line of the Parameter Entry Card.
- The type of Device Units is set to GDUs (Graphic Device Units).
- The Operator Digitize enable state, if enabled by the OPERATOR DIGITIZE command, is canceled. No last output block is sent by the Plotter. For GPIB Command/Response Formats 5 and 6, the permanently enabled Digitize state is entered. See Section 13 for more information.
- The Point light is turned off, except in Command/Response Formats 5 and 6 when it is on steadily.

Section 9

DEVICE COMMANDS

CONCEPTS

Most of the Device commands generate output messages. The form in which these are sent from the Plotter depends on the interface type and Command/Response Formats in use. Thus, it is important to understand the information on output messages in the Serial Interface or GPIB Interface sections, Sections 4 and 5 of this manual.

COMMAND DESCRIPTIONS

There are five commands described in this section:

- DEVICE RESET
- READ STATUS
- READ ERROR
- IDENTIFY
- SIZE

The following paragraphs describe each command separately.

DEVICE RESET

Purpose: This command resets graphic and alphanumeric parameters to power-up (default) values.

SERIAL

	$\mathrm{A_T}$ $\mathrm{A_D}$ **N**	[selector]
Sample:	$\mathrm{E_C}$ **A N**	**0**

GPIB

	N	[selector]

4050 BASIC (Data Byte Form)

	PRINT @pa,32: **"N"**	[;selector]
Sample:	**PRINT @1,32: "N";**	**0**

Parameters

selector
Selector 0-3, omitted selector equivalent to 0

Comments

The following is a list of actions that may occur as a result of DEVICE RESET command. Those listed with an asterisk (*) occur only for certain values of the selector argument. (See Table 9-1.) The remaining actions take place for all values of this argument.

1. Alphanumeric parameters are reset to default values. This is equivalent to an ALPHA RESET command. See the description of this command in the Alpha Commands section, Section 10, for a complete list of actions.

2. The following graphics parameters are reset:

   - The type of Graphic Units is reset to world units for both input and output.
   - The type of Device Units is reset to the default for the specific interface and Command/Response Format (ADUs for the Serial interface and for GPIB Format 7; GDUs for GPIB Formats 1 through 6).
   - The coordinate type is set to absolute.
   - Line type is reset to solid. Any defined dash patterns are lost, and dash length is set to 0.
   - UP is selected for Pen 1.
   - The effects of Plotter Transform commands except VIEWPORT and WINDOW are canceled. Any stored transforms are lost. (See Section 11 for descriptions of these commands.)
   - *The numerical ranges of the Viewport and Window are reset to initial values. See Section 1 about the rules for determining numerical ranges.

     The numerical adjustment of the Viewport and Window values is important if the Page has been altered with the front panel Set Page functions. Note that numerical adjustment only occurs when a DEVICE RESET command is received (or if the Page is SAVEd using the Parameter Entry Card.)
   - *The physical size and location of the Page is reset to the power-up size and position. This effect is equivalent to reselecting the current Page Size on the Parameter Entry Card. Any alterations to the size and location of the Page by front panel Set Page switches are lost. (These can be saved by using the SAVE selection on the Initial Page Size line of the Parameter Entry Card.)

- *The clipping control is reset to Viewport clipping. (See the SET CLIPPING CONTROL command in Section 11.)
- *The Formlength is reset to the default value for the Page size and location currently in use. This length is defined as the distance from the arrow on the lower right-hand corner of the platen to a distance 1.25 in (31.75 mm) to the left of the Page boundary.

3. Device Status Bits 13 and 14 are set. (See the READ STATUS command. The Device Status is returned when an argument of 0 is used.)

   - Bit 14 (Local Position Modification) is set to 1.
   - Bit 13 (Local Parameter Modification) is set to 1.

4. These additional DEVICE RESET actions occur:

   - The pen carriage moves to the load point position.
   - *All programmable macros are deleted.
   - *All downloaded characters are deleted.
   - The Operator Digitize enable state, if enabled by the OPERATOR DIGITIZE command, is canceled. No last output block is sent by the Plotter. In Serial Command/Response Formats 3 and 4 and GPIB Formats 5 and 6, the permanently enabled Digitize state is enabled.
   - The Auto Macro state is cancelled, if active. (See the AUTO MACRO command in the Programmable Macros Section.)
   - The Point light (on the Plotter front panel) is turned off (blinking is stopped. The Point light will be on steady in Command/Response Formats 5 and 6).

Table 9-1

EFFECTS OF SELECTOR VALUES

Value	Description of Effects
0	All the listed effects except: ● The physical size and location of the Page is not changed.
1	All of the effects of value 0 except: ● Programmable macros are not deleted. ● Downloaded characters are not deleted.
2	All the effects of value 1 except: ● Viewport and Window values are not changed. *NOTE* *If the Device Units were changed to a type other than the default for the interface/format in use before the SET WINDOW command was called, a DEVICE RESET 2 command will result in a distorted Window. To restore the correct numerical range to the Window, the type of Device Units must be set to the type in use when the Window was defined.* ● The current clipping control is not changed. ● The Formlength is not changed.
3	All the effects of value 1 plus: ● The physical size and location of the Page is reset.

READ STATUS

Purpose: This command causes the Plotter to generate an output message containing Plotter status information.

SERIAL

	(One Output Message Block Produced)	
	$^{A}_{T}$ $^{A}_{D}$ **O**	status-register-selector
Sample:	$^{E}_{C}$ **A O**	**1**

GPIB

	(One Output Message Block Produced)	
	V	status-register-selector

4050 BASIC (Data Byte Form)

	PRINT @pa,32: **"V"**;	status-register-selector
	INPUT @pa,32:	value-1-target-variable, value-2-target-variable, tag-value-target-variable
Sample:	**PRINT @1,32: "V";**	**0**
	INPUT @1,32:	**V1,V2,T**

4050 BASIC (Secondary Address Form)

	PRINT @pa,**0:**	status-register-selector
	INPUT @pa,**32:**	value-1-target-variable, value-2-target-variable, tag-value-target-variable
Sample:	**PRINT @1,0:**	**0**
	INPUT @1,32:	**V1,V2,T**

Parameters

status-register-selector
Status register selector, 0-3 or 0-7 (See *Comments*)

Outputs

See Comments information.

Comments

Legal status-register-selector ranges and the type and form of status information returned depend on the Command/Response Format and interface used. A complete description of the status information will be given later. There are three distinct cases for input argument and output responses:

1. Serial Command/Response Formats 1 through 2 and GPIB Command/Response Formats 1 through 4.

   The legal status-register-selector range is 0-3. Table 9-2 defines Plotter responses.

2. Serial Interface, Command/Response Formats 3 and 4.

   The legal status-register-selector range is 0-7. Table 9-3 defines Plotter responses.

3. GPIB Interface, Command/Response Formats 5 and 6.

   The legal status-register-selector range is 0-7. Table 9-4 defines Plotter responses.

NOTE

For selector values 0-3 and 6-7 only: the output message block consists of one, rather than three values. If using 4050 BASIC commands, use only one target variable; if more are used, the system will "hang."

Table 9-2

STATUS INFORMATION FOR SERIAL FORMATS 1 THROUGH 2 AND GPIB FORMATS 1 THROUGH 4

status-register-selector	Status Information			
			TAG Value	
	Value 1	Value 2	Serial Format 1	Others
0	Device Status	Current Available Memory	4	2
1	Max. Available Memory	Installed Options	4	2
2	Current X-Position	Current Y-Position	4 (pen up) 5 (pen down)	2 (pen up) 3 (pen down)
3	Pen Status	Accumulated Plot Time	4	2

Table 9-3

STATUS INFORMATION FOR SERIAL FORMATS 3 AND 4

status-register-selector	Status Information		
	Value 1	Value 2	TAG Value
0	0	Device Status	2
1	1	Current Available Memory	2
2	2	Maximum Available Memory	2
3	3	Installed Options	2
4	Current X-Position	Current Y-Position	0 (pen up) 4 (pen down)
5	Current X-Position	Current Y-Position	0 (pen up) 4 (pen down)
6	6	Pen Status	2
7	7	Accumulated Plot Time	2

Table 9-4

STATUS INFORMATION FOR GPIB FORMATS 5 AND 6

status-register-selector	Status Information		
	Value 1	Value 2	TAG Value
0	Device Status	none	none
1	Current Available Memory	none	none
2	Maximum Available Memory	none	none
3	Installed Options	none	none
4	Current X-Position	Current Y-Position	0 (pen up) 1 (pen down)
5	Current X-Position	Current Y-Position	0 (pen up) 1 (pen down)
6	Pen Status	none	none
7	Accumulated Plot Time	none	none

STATUS INFORMATION

The following paragraphs describe each word of status information that has been shown for this command (Device Status, Current Available Memory, Maximum Available Memory, Installed Options, Current X and Y Position, Pen Status, and Accumulated Plot Time).

Device Status

This is a 16-bit integer value in which each binary digit indicates if a particular condition is true or false. One bit position is unused; thus, 15 conditions may be indicated. (See Table 9-5.) A binary value of one for a specified position indicates that the state associated with that position is true. A zero indicates the state is not true. Each bit position is described in subsequent paragraphs. Bit 15 is the most-significant bit and Bit 0 is the least-significant bit.

Table 9-5

DEVICE STATUS BITS

Bit Number	Description
0	Internal Error
1	Command Response Error
2	Communication Error
3	Y Below
4	Y Above
5	X Left
6	X Right
7	Page Y Mirrored
8	Page X Mirrored
9	Pause or Busy
10	Roll Mode
11	Not Used
12	Local Data Reset
13	Local Parameter Modification
14	Local Position Modification
15	Out of Media

Bit 15 (Out of Media). This bit is only used when Media Advance (Option 36) is present and Roll mode is selected on the Parameter Entry Card. Bit 15 is set to one if an out-of-media condition occurs during any media motion. A media advance less than one inch will usually result in an out-of-media condition. This bit is set to one at power-up (only if Roll mode is selected).

NOTE

An out-of-media condition may be incorrectly set during manual media motion or front panel Set Form Length. This bit value should not be used with these functions.

Bit 14 (Local Position Modification). This bit is set to one at power-up, after a DEVICE RESET command, whenever the joystick is used, or after the following front panel functions are used: all Media group functions, all Lower Left/Upper Right functions, and Initialize. This bit will be set to zero after a READ STATUS command that returns Device Status (argument of zero).

This bit permits the host computer program to determine if an action occurred locally at the Plotter that may have caused a change in position (since the last time that Device Status was read.)

Bit 13 (Local Parameter Modification). This bit is set to one during power-up, after a DEVICE RESET command, a front panel Set Viewport Lower Left or Upper Right, or a front panel Set FormLength has occurred. This bit is set to zero after a READ STATUS command that requests Device Status (argument of zero).

A value of one for this status bit signifies that some action has occurred locally at the Plotter that may have modified programmed parameters.

Bit 12 (Local Data Reset). This bit is set to one at power-up, and after a front panel Data Reset or Initialize. It is cleared by a READ STATUS command that requests Device Status. Bit 12 indicates that commands from a host computer program may have been deleted before execution.

Bit 11 (Not Used). This bit is always set to zero.

Bit 10 (Roll Mode). This bit is set to one if Roll mode is selected on the Parameter Entry Card. It is zero if Sheet mode is selected.

Bit 9 (Pause or Busy). A Pause/Busy state occurs when a front panel Pause state is active or a front panel Media function is in process. Under these conditions, commands are not executed. However, a READ STATUS (0) command will be executed if there are no preceding unprocessed commands.

Bit 8 (Page X Mirrored). This bit is set to one if the Page lower-left x-coordinate has been defined to the right of the Page upper-right x-coordinate. Bit 8 is set to zero otherwise.

Bit 7 (Page Y Mirrored). This bit is set to one if the Page lower left y-coordinate has been defined below the Page upper right y-coordinate. Bit 7 is set to zero otherwise.

Bit 6 (X Right), Bit 5 (X Left), Bit 4 (Y Above), Bit 3 (Y Below). Bits 6 through 3 indicate when the current pen position is outside the current clipping limits. These limits are defined by the current Viewport boundary or Page boundary if Page clipping is program-selected (see the SET CLIPPING CONTROL command.)

Refer to Figure 9-1. Left, Right, Above, and Below are defined with respect to the horizontally oriented platen surface. These directions are not altered by any Page, Viewport, or axis-orientation operation.

X LEFT Y ABOVE	Y ABOVE	X RIGHT Y ABOVE
X LEFT	X AND Y ON OR WITHIN CLIPPING LIMITS.	X RIGHT
X LEFT Y BELOW	Y BELOW	X RIGHT Y BELOW

3806-66

Figure 9-1. Clipping Limit Positions.

Bit 2 (Communications Error). A communications error such as parity or framing error will set this bit to one. The error can be identified using the READ ERROR command (if no other error has occurred first). This bit is cleared by a front panel Error Reset or by a READ ERROR command.

Bit 1 (Command/Response Error). Command/response errors occur when an invalid command, argument, or format is received, or when an invalid response value results. This bit is cleared by a front panel Error Reset or by a READ ERROR command.

Bit 0 (Internal Error). An internal error condition will set this bit to one. A non-fatal error can be cleared by a front panel Error Reset or by sending the READ ERROR command.

Current Available Memory

This integer value is the number of bytes of memory currently available in the Plotter for storage of programmable macros, downloaded characters, and transforms. The numerical range returned is from zero to the maximum available memory. On the Serial interface, the range is less than the maximum available memory.

Maximum Available Memory

This is the maximum memory which is available for the storage of programmable macros, downloaded characters, and transforms. The numerical value depends on the interface selected and on which plotter options are installed.

Installed Options

This status value indicates which 4663 Plotter options are installed. See Table 9-6. The presence of a particular option is determined by examining its binary position. A value of one indicates the option is present. Otherwise, a zero exists for the bit. (Bit 15 is the most significant bit while Bit 0 is the least significant bit.)

Table 9-6

INSTALLED OPTIONS STATUS

Bit	Description
Bit 15 to Bit 5	Bit positions 15 through 5 are unused in a standard Plotter and are set to zero.
Bit 4	Math Character Set (Option 32)
Bit 3	Downloadable Characters (Option 32)
Bit 2	Programmable Macros (Option 31)
Bit 1	Circular Interpolation (Option 31)
Bit 0	Media Advance (Option 36)

Current X and Y Position

These values are the current pen position. The Plotter response is the same as with a DIGITIZE command.

Pen Status

This status value indicates the currently selected pen and if the pen is up or down. See Table 9-7.

Table 9-7

PEN STATUS

Bit	Bit Value	Pen Action
Bit 15 to Bit 3	Bits are not used and set to zero	None
Bit 2 to Bit 1	Bit 1 = 1, Bit 2 = 0 Bit 1 = 0, Bit 2 = 1	Pen 1 Pen 2
Bit 0	0 1	Selected pen is up. Selected pen is down.

Accumulated Plot Time

This is approximately the total number of hours of pen carriage motion since the Parameter memory was cleared. (A battery within the Plotter maintains Parameter Entry Card selections and accumulated plot time).

It is possible that this value will not initialize to zero when the Parameter memory is first powered on. For this case, only changes in accumulated plot time should be considered; the actual value may not be meaningful.

References

See either the Serial Interface or GPIB Interface sections (Sections 4 and 5) for information about decoding output responses.

READ ERROR

Purpose: This command causes the Plotter to transmit a two-block message containing error information. In addition, the front panel Error Reset light is turned off, and the error data is cleared.

SERIAL

(Command/Response Formats 1-2 Only)
(Two Output Message Blocks Produced)

$^{A}{T}$ $^{A}{D}$ **CE**

Sample: $^{E}{C}$ **A CE**

GPIB

(Two Output Message Blocks Produced)

CE

4050 BASIC (Data Byte Form)

```
         PRINT @pa,32: "CE"
         INPUT @pa,32:              first-value-1-target-variable,
                                    first-value-2-target-variable,
                                    first-tag-value-target-variable

         INPUT @pa,32:              last-value-1-target-variable,
                                    last-value-2-target-variable,
                                    last-tag-value-target-variable
Sample:  PRINT @1,32: "CE"
         INPUT @1,32:               V1,V2,T1
         INPUT @1,32:               V3,V4,T2
```

Outputs

First output message block:

value 1
Error-number x 256 + error-count

value 2
Error-parameter-1 x 256 + error-parameter-2

tag value
See Table 9-8.

Last output message block:

value 1
Error-parameter-3 x 256 + error-parameter-4

value 2
Error-parameter-5 x 256 + error-parameter-6

tag value
See Table 9-8.

Eight items of information are included in the outputs. Each has an integer value in the range 0-255:

error-number
Identifies the error that occurred. A value of zero indicates no error.

error-count
This is the total number of errors that have occurred since power-up or the last time an error was cleared. (See *Comments*).

error-parameters-1 through -6
These give information on the internal source and cause of the error. They are intended for debugging use only. One or more parameters are set for each error reported.

Table 9-8

TAG VALUES FOR READ ERROR COMMAND

TAG Meaning	Serial Formats		GPIB Formats
	1	2	1 thru 6
First Block	2	0	0
Last Block	4	2	2

Comments

The error-number information identifies the first error that occurs after power-up or since the last READ ERROR command or front panel Error Reset function. Subsequent errors increment the error-count.

References

See either the Serial Interface or GPIB Interface sections (Sections 4 and 5) for information on decoding output responses from the Plotter.

Also, see Appendix J for descriptions of error numbers.

Errors

Error 48
(Invalid Command for Selected Output Format)

IDENTIFY

Purpose: This command causes the Plotter to transmit a one-block message that identifies the Plotter, the amount of memory installed, and the firmware release number.

SERIAL

(Command/Response Formats 1-2 Only)
(One Output Message Block Produced)

A_T A_D **Q**

Sample: E_C **A Q**

GPIB

(Command/Response Formats 1-4 Only)
(One Output Message Block Produced)

I

4050 BASIC (Data Byte Form)

(Command/Response Formats 1-4 Only)

```
        PRINT @pa,32: "I"
        INPUT @pa,32:                 value-1-target-variable,
                                      value-2-target-variable,
                                      tag-value-target-variable
Sample: PRINT @1,32: "I"              V1,V2,T
        INPUT @1,32:                  V1,V2,T
```

4050 BASIC (Secondary Address Form)

(Command/Response Formats 1-4 Only)

```
        INPUT @pa[,13]:               value-1-target-variable,
                                      value-2-target-variable,
                                      tag-value-target-variable
Sample: INPUT @1,13:                  V1,V2,T
```

Outputs

value 1
4663 (Plotter model number)

value 2
Number-kilobytes-memory x 256 + firmware-release-number

tag value
See Table 9-9.

Two items of output information are returned in value 2:

- number-kilobytes-memory is the amount of memory (RAM) installed (8 for a standard Plotter).
- firmware-release-number is an integer value indicating the version number of the firmware installed in the Plotter.

Reference

See elther Section 4 or Section 5 for information on decoding output responses from the Plotter.

See the SIZE command if the Plotter uses other than the Command/Response Formats indicated in Table 9-9.

Table 9-9

TAG VALUES FOR IDENTIFY COMMAND

Format	TAG Value
Serial Format 1	4
Serial Format 2	2
GPIB Formats 1 through 4	2

SIZE

Purpose: This command causes the Plotter to transmit a one-block message that gives the size of the plotting area and an identification word.

SERIAL

(Command/Response Formats 3-4 Only)
(One Output Message Block Produced)

$^{A}{}_{T}$ $^{A}{}_{D}$ **Q**

Sample: $^{E}{}_{C}$ **A Q**

GPIB

(Command/Response Formats 5-6 Only)
(One Output Message Block Produced)

I

4050 BASIC (Data Byte Form)

(Command/Response Formats 5-6 Only)

PRINT @pa,32: "I"
INPUT @pa,32: value-1-target-variable, value-2-target-variable, tag-value-target-variable

Sample: **PRINT @1,32: "I"**
INPUT @1,32: V1,V2,T

4050 BASIC (Secondary Address Form)

(Command/Response Formats 5-6 Only)

INPUT @pa[,13]: value-1-target-variable, value-2-target-variable, tag-value-target-variable

Sample: **INPUT @1,13: V1,V2,T**

Outputs

Output information is interface-dependent, as follows:

Serial Interface:

value 1
5649 (represents x-size x 256 + y-size, where x-size equals 22 in and y-size equals 17 in)

value 2
Identification word (see Table 9-10)

tag value
3

GPIB Interface:

value 1
150 (represents x-size in GDUs)

value 2
100 (represents y-size in GDUs)

tag value
Identification word (see Table 9-10)

Table 9-10

IDENTIFICATION WORD

Bit[a]	Information
Bit 0 to Bit 3	Firmware release number.
Bit 4 and Bit 5	Both bits are set to one indicating that more than 1600 bytes are available at power-up for the input buffer.
Bit 6 Bit 7 Bit 8 Bit 9 Bit 10	Media Advance Installed (Option 36) Circular Interpolation Installed (Option 31) Programmable Macros Installed (Option 31) Downloadable Characters Installed (Option 32) Math Character Set Installed (Option 32)
Bit 11 to Bit 15	Set to zero on standard Plotter.

[a] Bit 0 is the least-significant bit.

References

See either Section 4 or Section 5 for information on decoding output responses from the Plotter.

See the IDENTIFY command if the Plotter uses Command/Response Formats other than those listed in the syntax boxes.

Section 10

ALPHA COMMANDS

INTRODUCTION

The Plotter has an internal alpha generator that enables the user to print and position alphanumeric characters and symbols which are defined in nine resident fonts. (With Option 32 installed, 16 fonts are available.) Four action commands and twelve setup commands are available. Refer to Table 10-1.

For the PRINT action command, the alpha generater prints 96 ASCII characters (${}^{S}{}_{P}$ through ${}^{D}{}_{L}$) and responds to the nine control characters (${}^{B}{}_{L}$ through ${}^{S}{}_{I}$). For the other three action commands listed in Table 10-1, the alpha generator prints symbols centered about the current pen carriage position (PRINT CENTERED command), moves the pen carriage a specified number of character spaces and line spaces (CHARACTER MOVE command), or moves the pen carriage to the first character position on the Viewport (MOVE TO HOME command).

The alpha setup commands, and the Alpha Character Quality line on the Parameter Entry Card are used to control the four alpha action commands. Refer again to Table 10-1 for the interrelationship between the setup commands and the action commands. This section includes fundamental alpha concepts and descriptions of the resident character sets, followed by more information for each setup and action command.

Table 10-1

INTERRELATIONSHIPS OF ALPHA SETUP AND ACTION COMMANDS

Setup Commands	Action Commands			
	Print	Print Centered	Move To Home	Character Move
SET ALPHA SIZE	X	X	X	X
SET ALPHA RATIO	X			X
SET ALPHA DIMENSION	X	X	X	X
SET ALPHA ROTATION (Angle)	X	X	X	X
SET ALPHA ROTATION (Left Margin)	X			
SET ALPHA SLANT	a	X		
SET ALPHA SCALE	X	X	X	X
SET TAB SEPARATION	X			
SET ALPHA MARGIN SEPARATION	X			X
SELECT ALPHA SPACING CONTROL	X	b		
SELECT STANDARD ALPHA FONT (If active)	X	X		
SELECT ALTERNATE ALPHA FONT (If active)	X	X		
RESET ALPHA PARAMETERS	X	X	X	X
SET ARC SMOOTHNESS	X	X		

[a] ALPHA SLANT affects noncontrol characters only.
[b] Default ALPHA SPACING CONTROL is recommended.

CONCEPTS

The following general concepts are presented here:

- PRINT Command Fundamentals
- The Automatic Move-To-Home Function
- The Home Position
- Setting Size and Spacing of Printed Characters
- Resident Alpha Fonts
- Spacing Control for the PRINT Command
- Adding Margin Control for PRINT and CHARACTER MOVE Commands

PRINT COMMAND FUNDAMENTALS

For each character (${}^{S}{}_{P}$ through ${}^{D}{}_{L}$) within a PRINT command, the alpha generator does the following:

- Moves and Draws as necessary to print the selected character with the character's lower-left corner located at the current pen position.
- Moves to the start point for the next character (as determined by the SET ALPHA SIZE, SET ALPHA RATIO, and SET ALPHA SCALE set-up commands).

The alpha generator prints characters from the active font, according to all alpha setup commands for size, rotation, slant, scale, etc. The alpha generator also processes control characters ${}^{B}{}_{L}$ through ${}^{S}{}_{I}$. The PRINT command procedure just described applies for all setups where default SELECT ALPHA SPACING CONTROL and SET ALPHA MARGIN SEPARATION commands apply. (Spacing control and margin control are described later in this section.)

THE AUTOMATIC MOVE-TO-HOME FUNCTION

The Plotter expects the first PRINT command after power-up to be preceded by a MOVE or DRAW command or Joystick action. If this expectation is not met, the Plotter takes corrective action by preceding the PRINT command with a Move-To-Home action. (The home position is defined below.) The automatic Move-To-Home function also applies for the first PRINT command after a front panel Media Change, Locate Lower Left, or Locate Upper Right action.

THE HOME POSITION

The home position is a point located one character height below the upper left corner of the Viewport. (If the alpha rotation angle is nonzero, the home position is rotated by this angle about the upper-left corner of the Viewport.)

SETTING THE SIZE AND SPACING OF PRINTED CHARACTERS

Either the SET ALPHA SIZE or the SET ALPHA DIMENSION commands sets the character width and height used by alpha action commands. Both commands are specified in terms of Initial Window units; current Window parameters, if different, have no effect. Default sizes are described in the description of the SET ALPHA SIZE command.

The spacing between characters is determined by the SET ALPHA RATIO command, in conjunction with character width and character height. The relationship between these setup parameters, pen position, and the printed character is illustrated for the PRINT command in Figure 10-1.

Both the character size and spacing can be increased or decreased by an additional setup command: the SET ALPHA SCALE command. For example, alpha scale factors of 2,2 set the size and spacing of printed characters and character moves to be two times the size and spacing of the default scale factor.

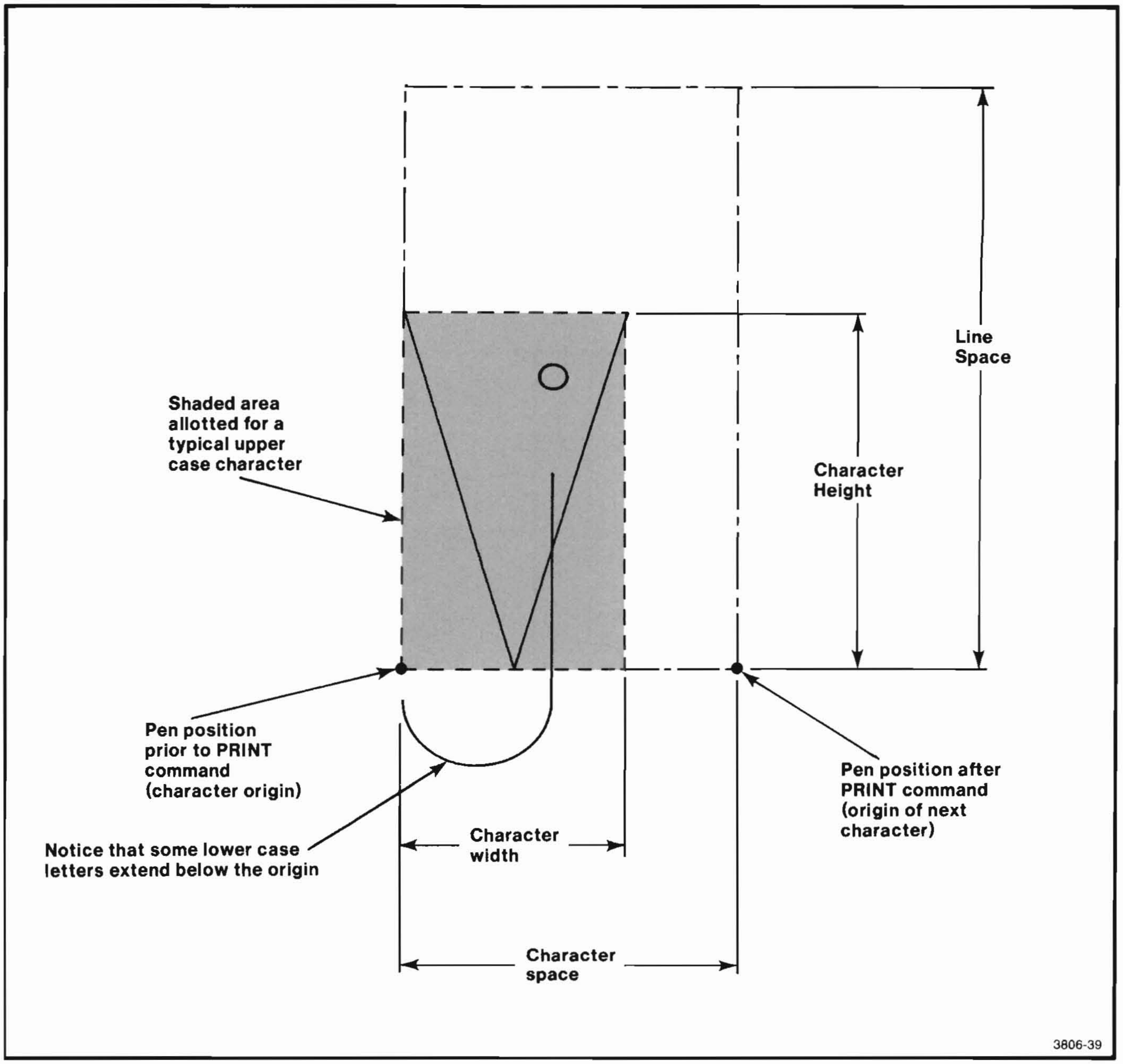
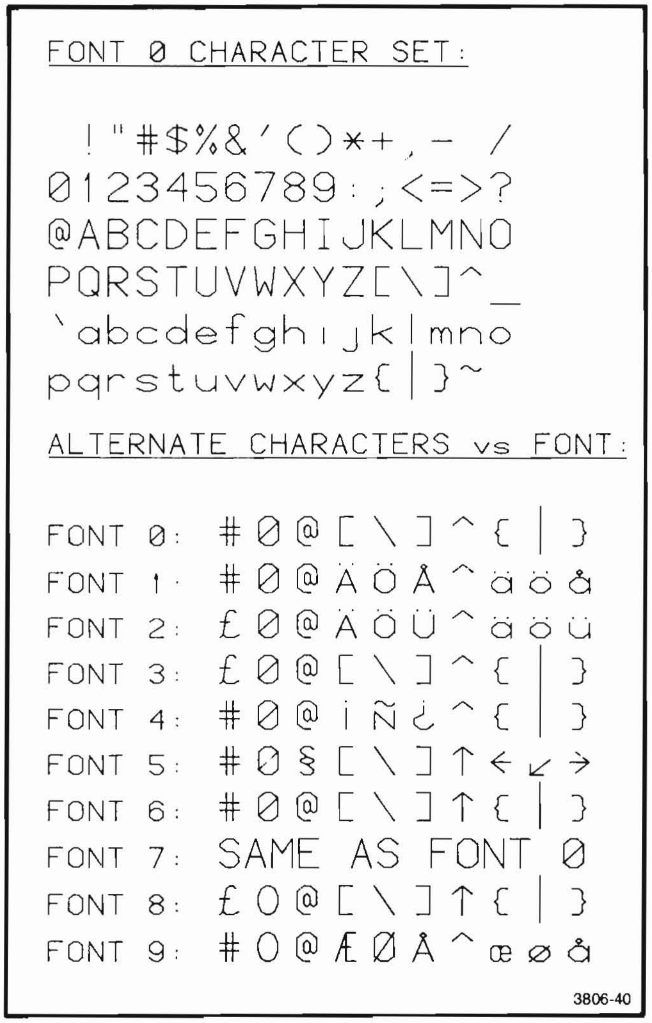

Line Space
Shaded area allotted for a typical upper case character
Character Height
Pen position prior to PRINT command (character origin)
Notice that some lower case letters extend below the origin
Pen position after PRINT command (origin of next character)
Character width
Character space
3806-39

Figure 10-1. Alpha Size and Spacing for the PRINT Command.

RESIDENT ALPHA FONTS

The alpha generator has resident character definitions for ASCII characters SP through DL in nine fonts (or in 12 resident fonts if Option 32 is installed). Nine resident fonts are available for NORMAL or ENHANCED selections of the Alpha Character Quality line on the Parameter Entry Card. NORMAL quality characters use fewer Moves and Draws and are printed more quickly than the ENHANCED characters. ENHANCED quality characters offer added smoothness (which can be controlled by the SET ARC SMOOTHNESS command). The nine resident alpha fonts are shown in Figures 10-2 (NORMAL) and 10-3 (ENHANCED) with the complete Font 0 shown at the top of the figures. Font 0 includes two nonprinting characters, SP and DL. (The DL character is a nonprinting and nonadvancing character.) All other fonts are identical to Font 0 except for the character substitutions shown in Figures 10-2 and 10-3.

For Option 32, additional character definitions are resident in Fonts 10, 11, and 15. Fonts 10 and 11 are shown in Figures 10-4 and 10-5. All characters unchanged from Font 0 are shaded and are shown with ENHANCED alpha character quality.



FONT Ø CHARACTER SET:

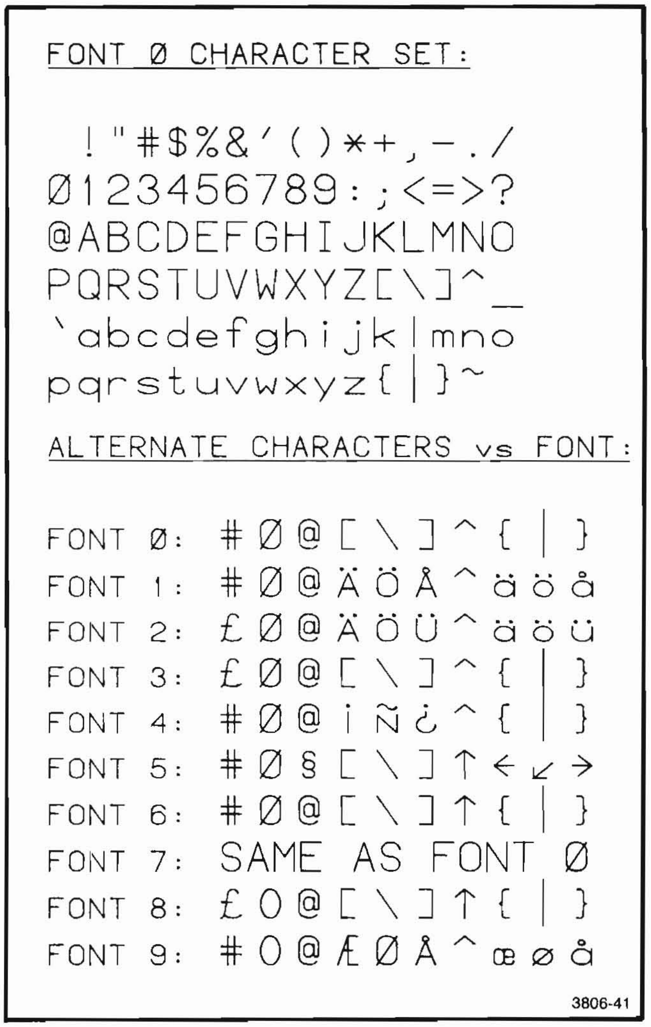
! " # $ % & ' ( ) * + , - . /
Ø 1 2 3 4 5 6 7 8 9 : ; < = > ?
@ A B C D E F G H I J K L M N O
P Q R S T U V W X Y Z [\ ] ^ _
` a b c d e f g h i j k l m n o
p q r s t u v w x y z { | } ~

ALTERNATE CHARACTERS vs FONT:

FONT Ø: # Ø @ [\ ] ^ { | }
FONT 1: # Ø @ Ä Ö Å ^ ä ö å
FONT 2: £ Ø @ Ä Ö Ü ^ ä ö ü
FONT 3: £ Ø @ [\ ] ^ { | }
FONT 4: # Ø @ ¡ Ñ ¿ ^ { | }
FONT 5: # Ø § [\ ] ↑ ← ↙ →
FONT 6: # Ø @ [\ ] ↑ { | }
FONT 7: SAME AS FONT Ø
FONT 8: £ O @ [\ ] ↑ { | }
FONT 9: # O @ Æ Ø Å ^ æ ø å

3806-40

Figure 10-2. NORMAL Alpha Character Quality.



FONT Ø CHARACTER SET:

! " # $ % & ' ( ) * + , - . /
Ø 1 2 3 4 5 6 7 8 9 : ; < = > ?
@ A B C D E F G H I J K L M N O
P Q R S T U V W X Y Z [\ ] ^ _
` a b c d e f g h i j k l m n o
p q r s t u v w x y z { | } ~

ALTERNATE CHARACTERS vs FONT:

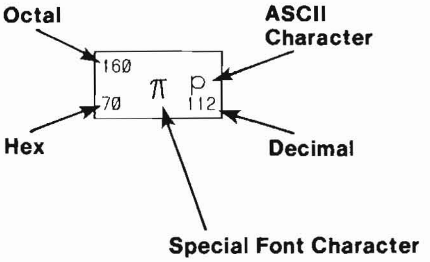
FONT Ø: # Ø @ [\ ] ^ { | }
FONT 1: # Ø @ Ä Ö Å ^ ä ö å
FONT 2: £ Ø @ Ä Ö Ü ^ ä ö ü
FONT 3: £ Ø @ [\ ] ^ { | }
FONT 4: # Ø @ ¡ Ñ ¿ ^ { | }
FONT 5: # Ø § [\ ] ↑ ← ↙ →
FONT 6: # Ø @ [\ ] ↑ { | }
FONT 7: SAME AS FONT Ø
FONT 8: £ O @ [\ ] ↑ { | }
FONT 9: # O @ Æ Ø Å ^ æ ø å

3806-41

Figure 10-3. ENHANCED Alpha Character Quality.

▨ 4Ø 2Ø 32	▨ 6Ø 3Ø Ø Ø 48	▨ 1ØØ 4Ø @ @ 64	12Ø 5Ø Π P 8Ø	▨ 14Ø 6Ø \` \` 96	16Ø 7Ø π p 112
▨ 41 21 ! ! 33	▨ 61 31 1 1 49	▨ 1Ø1 41 A A 65	▨ 121 51 Q Q 81	141 61 α a 97	▨ 161 71 q q 113
▨ 42 22 " " 34	▨ 62 32 2 2 5Ø	▨ 1Ø2 42 B B 66	122 52 P R 82	142 62 β b 98	162 72 ρ r 114
▨ 43 23 # # 35	▨ 63 33 3 3 51	1Ø3 43 ψ C 67	123 53 Σ S 83	143 63 ψ c 99	163 73 σ s 115
▨ 44 24 $ $ 36	▨ 64 34 4 4 52	1Ø4 44 Δ D 68	▨ 124 54 T T 84	144 64 δ d 1ØØ	164 74 τ t 116
▨ 45 25 % % 37	▨ 65 35 5 5 53	▨ 1Ø5 45 E E 69	125 55 Θ U 85	145 65 ε e 1Ø1	165 75 ϑ u 117
▨ 46 26 & & 38	▨ 66 36 6 6 54	1Ø6 46 Φ F 7Ø	126 56 Ω V 86	146 66 φ f 1Ø2	166 76 ω v 118
▨ 47 27 ' ' 39	▨ 67 37 7 7 55	1Ø7 47 Γ G 71	▨ 127 57 W W 87	147 67 γ g 1Ø3	167 77 ς w 119
▨ 5Ø 28 ((4Ø	▨ 7Ø 38 8 8 56	▨ 11Ø 48 H H 72	▨ 13Ø 58 X X 88	15Ø 68 η h 1Ø4	17Ø 78 χ x 12Ø
▨ 51 29)) 41	▨ 71 39 9 9 57	▨ 111 49 I I 73	131 59 ϒ Y 89	151 69 ι i 1Ø5	171 79 υ y 121
▨ 52 2A * * 42	▨ 72 3A : : 58	112 4A Ξ J 74	▨ 132 5A Z Z 9Ø	152 6A ξ j 1Ø6	172 7A ζ z 122
▨ 53 2B + + 43	▨ 73 3B ; ; 59	▨ 113 4B K K 75	▨ 133 5B [[91	153 6B κ k 1Ø7	▨ 173 7B { { 123
▨ 54 2C , , 44	▨ 74 3C < < 6Ø	114 4C Λ L 76	▨ 134 5C \ \ 92	154 6C λ l 1Ø8	174 7C ∂ | 124
▨ 55 2D – – 45	▨ 75 3D = = 61	▨ 115 4D M M 77	▨ 135 5D]] 93	155 6D μ m 1Ø9	▨ 175 7D } } 125
▨ 56 2E . . 46	▨ 76 3E > > 62	▨ 116 4E N N 78	▨ 136 5E ^ ^ 94	156 6E ν n 11Ø	▨ 176 7E ~ ~ 126
▨ 57 2F / / 47	▨ 77 3F ? ? 63	▨ 117 4F O O 79	▨ 137 5F — _ 95	157 6F o o 111	▨ 177 7F 127

Key



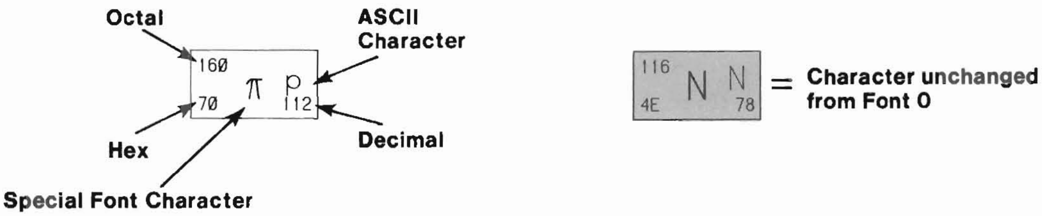
▨ 116 4E N N 78 = Character unchanged from Font 0

3806-42

Figure 10-4. Resident Font 10 (Option 32).

4Ø 2Ø 32	6Ø ∘ Ø 3Ø 48	1ØØ 2 @ 4Ø 64	12Ø Π P 5Ø 8Ø	14Ø ⊣ \ 6Ø 96	16Ø π p 7Ø 112
41 · ! 21 33	61 ∫ 1 31 49	1Ø1 A A 41 65	121 √ Q 51 81	141 α a 61 97	161 — q 71 113
42 □ " 22 34	62 ∴ 2 32 5Ø	1Ø2 B B 42 66	122 P R 52 82	142 β b 62 98	162 ρ r 72 114
43 ○ # 23 35	63 ∃ 3 33 51	1Ø3 ψ C 43 67	123 Σ S 53 83	143 ψ c 63 99	163 σ s 73 115
44 △ $ 24 36	64 ⊂ 4 34 52	1Ø4 Δ D 44 68	124 T T 54 84	144 δ d 64 1ØØ	164 τ t 74 116
45 + % 25 37	65 ∪ 5 35 53	1Ø5 E E 45 69	125 Θ U 55 85	145 ε e 65 1Ø1	165 ϑ u 75 117
46 ◇ & 26 38	66 ⊃ 6 36 54	1Ø6 Φ F 46 7Ø	126 Ω V 56 86	146 φ f 66 1Ø2	166 ω v 76 118
47 ⊠ ' 27 39	67 ∩ 7 37 55	1Ø7 Γ G 47 71	127 W W 57 87	147 γ g 67 1Ø3	167 ς w 77 119
5Ø ⌑ (28 4Ø	7Ø ∞ 8 38 56	11Ø H H 48 72	13Ø X X 58 88	15Ø η h 68 1Ø4	17Ø χ x 78 12Ø
51 ⊕) 29 41	71 ∇ 9 39 57	111 I I 49 73	131 ϒ Y 59 89	151 ι i 69 1Ø5	171 υ y 79 121
52 ⊞ * 2A 42	72 : : 3A 58	112 Ξ J 4A 74	132 Z Z 5A 9Ø	152 ξ j 6A 1Ø6	172 ζ z 7A 122
53 + + 2B 43	73 ≠ ; 3B 59	113 K K 4B 75	133 [[5B 91	153 κ k 6B 1Ø7	173 ℓ { 7B 123
54 ⊥ , 2C 44	74 ≤ < 3C 6Ø	114 Λ L 4C 76	134 ± \ 5C 92	154 λ l 6C 1Ø8	174 ∂ | 7C 124
55 ‖ - 2D 45	75 ≡ = 3D 61	115 M M 4D 77	135]] 5D 93	155 μ m 6D 1Ø9	175 ↓ } 7D 125
56 · . 2E 46	76 ≥ > 3E 62	116 N N 4E 78	136 ↑ ^ 5E 94	156 ν n 6E 11Ø	176 ~ ~ 7E 126
57 ÷ / 2F 47	77 ∝ ? 3F 63	117 O O 4F 79	137 — _ 5F 95	157 o o 6F 111	177 7F 127

Key



1ØØ 2 @ 4Ø 64	137 — _ 5F 95	14Ø ⊣ \ 6Ø 96

Non-Advancing Characters in Font 11

3806-43

Figure 10-5. Resident Font 11 (Option 32).

NOTE

Character definitions provided by Option 32 are NOT affected by the Alpha Character Quality line on the Parameter Entry Card. When Option 32 characters are mixed with other resident characters, be sure to select the proper alpha character quality, so that all characters will have the same alpha quality as the Option 32 characters. Select ENHANCED quality when using resident fonts 10 and 11.

Option 32 provides a single character substitution in Font 15 (other characters are the same as in Font 0). The new character replaces $^{D}{}_{L}$ and consists of a grid that covers the usual extremes for printed letters (including the descender region for some lowercase letters) — a tool useful when designing new characters. Figure 10-6 shows three copies of this special character, with an X and a y superimposed over the last two.

For details on how to select and activate a font number, see the descriptions of the SET STANDARD ALPHA FONT and SET ALTERNATE ALPHA FONT commands in this section. For information on how to create a new character definition, or a new font, see Section 15.

SPACING CONTROL FOR THE PRINT COMMAND

The SELECT ALPHA SPACING CONTROL set-up command controls PRINT command actions. For a selection of zero, subsequent PRINT actions allocate a constant width for each character, printing narrow characters at the center of this region. The distance from the start of the region allocated for one character to the start of the next is constant. For a selection of one, the alpha generator does the following for each character printed (based on the actual width of the character):

1. The selected character is printed (by Moves and Draws) such that the character's left edge is located at the current pen carriage position.

2. Then the pen carriage moves to the start point for the next character (a constant distance from the right edge of the character just printed).

For either selection, the spacing between typical uppercase letters (for example, A's) is identical, as defined by the SET ALPHA RATIO command. For the default selection of zero, additional space is added before and after narrow characters (for example, i's). For the selection of one, the spacing from the end of one character to the start of the next is constant. Figure 10-7 shows the same words printed with spacing control selections set to zero (the default) and one.

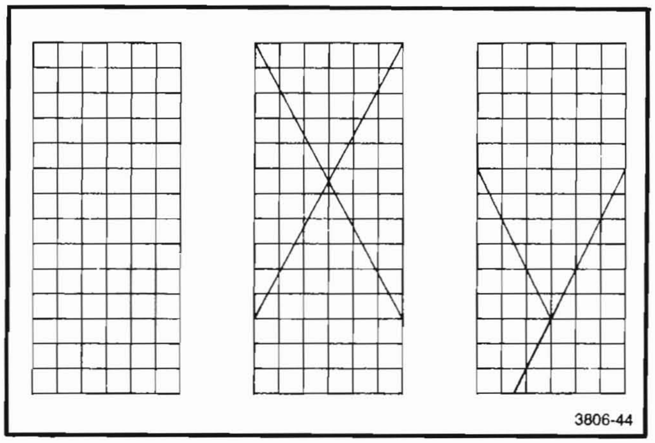
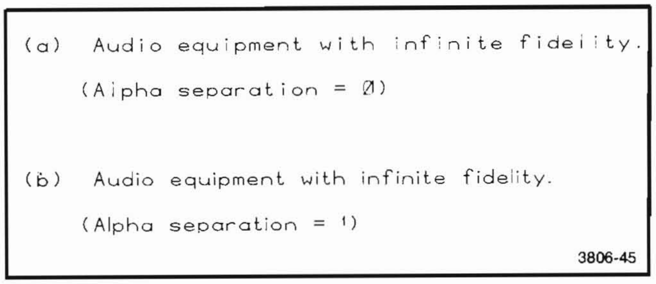


Figure 10-6. The Grid Character in Font 15 (Option 32).



(a) Audio equipment with infinite fidelity.

(Alpha separation = Ø)

(b) Audio equipment with infinite fidelity.

(Alpha separation = 1)

Figure 10-7. Comparison of Alpha Spacing Control Selections.

ADDING MARGIN CONTROL FOR PRINT AND CHARACTER MOVE COMMANDS

The SET MARGIN SEPARATION command activates margin control whenever a separation greater than zero character spaces is specified. With margin control active, the Plotter uses the following procedure for each character printed:

1. If the character is too wide to fit within the right margin, the alpha generator performs a Carriage Return and a Linefeed. If the pen carriage is to the left of the left margin, the pen carriage moves to the left margin.
2. The alpha generator prints the character and moves to the origin of the next character.
3. If the new origin is to the right of the right margin, the alpha generator performs a Carriage Return and a Linefeed.

For the CHARACTER MOVE command the alpha generator performs margin control according to this third step.

COMMAND DESCRIPTIONS

There are 16 commands described in this section:

- SET ALPHA SIZE
- SET ALPHA RATIO
- SET ALPHA DIMENSION
- SET ALPHA ROTATION
- SET ALPHA SLANT
- SET ALPHA SCALE
- SET TAB SEPARATION
- SET ALPHA MARGIN SEPARATION
- SELECT ALPHA SPACING CONTROL
- SELECT STANDARD ALPHA FONT
- SELECT ALTERNATE ALPHA FONT
- RESET ALPHA PARAMETERS
- PRINT
- PRINT CENTERED
- MOVE-TO-HOME
- CHARACTER MOVE

The following pages describe each command separately.

SET ALPHA SIZE

Purpose: This command sets the character width and the character height for a typical uppercase letter (for example, A). The width and height, specified in Initial World Units, are used by all subsequent alpha action commands.

SERIAL

	^{A}T ^{A}D **BZ**	character-width,character-height
Sample:	^{E}C **A BZ**	**12,22**

GPIB

	BZ	character-width,character-height

4050 BASIC (Data Byte Form)

	PRINT@pa,32: **"BZ"**;	character-width;character-height
Sample:	**PRINT@1,32: "BZ";**	**12;22**

Parameters

character-width
Width of a typical uppercase letter, in Initial World Units

character-height
Height of a typical uppercase letter, in Initial World Units

Comments

The default alpha size (set at power-up or after an ALPHA RESET or DEVICE RESET command) depends on the selection of Device Units and on the size of the Initial Window. The default Device Units are ADUs for a Serial interface and GDUs for a GPIB interface. Table 10-2 shows default alpha sizes for C-SIZE, DRAFTING, and HORIZONTAL selections on the Parameter Entry Card. Table 10-3 shows how to calculate default alpha size for any Parameter Entry Card selection. Alpha size is expressed in Table 10-3 as a fraction of the longer axis of the Viewport or Initial Window. To determine PHYSICAL size, replace L with the physical length of the longer axis of the Viewport. To determine numerical size, replace L with the numerical range of the longer axis of the Initial Window.

Table 10-2

DEFAULT ALPHA SIZE FOR C-SIZE/DRAFTING/HORIZONTAL PARAMETER ENTRY CARD SELECTIONS

Device Units	Aspect Ratio	Character Width	Character Height	Character Space[a]	Line Space[a]
ADU (Addressable Device Units) Default for Serial Interface	(Any)	37 1/3	53 7/9	56	88
GDU (Graphic Device Units) Default for GPIB Interface	FULL	1.079	1.554	1.619	2.543
	3X:2Y	1.195	1.721	1.792	2.816
	4X:3Y	1.044	1.504	1.566	2.461
	1X:1Y	0.796	1.147	1.195	1.877
	3X:4Y	1.044	1.504	1.566	2.461
	2X:3Y	1.195	1.721	1.792	2.816
mm (millimeters)	FULL	4.248	6.119	6.372	10.014
	3X:2Y	4.248	6.119	6.372	10.014
	4X:3Y	4.111	5.922	6.167	9.691
	1X:1Y	3.136	4.517	4.703	7.391
	3X:4Y	3.136	4.517	4.703	7.391
	2X:3Y	3.136	4.517	4.703	7.391

[a] Character space and line space values only apply to default ALPHA RATIO selections.

Table 10-3

DEFAULT ALPHA SIZE

Device Units	Alpha Size[a]	
	Width	Height
ADU	(7/768)L or approx .009115L	(121/9216)L or approx .01313L
GDU or mm	(224/28125)L or approx .007964L	(968/84375)L or approx .01147L

[a] L is the longer axis length of the Initial Window or Viewport.

References

The SET ALPHA DIMENSION command may be used instead of this command.

SET ALPHA RATIO

Purpose: This command specifies the ratio of character space to character width and the ratio of line space to character height. These ratios (and the character width and character height) are used by subsequent PRINT and CHARACTER MOVE Commands.

SERIAL

	^{A}T ^{A}D **BI**	x-ratio,y-ratio
Sample:	^{E}C **A BI**	**2,1.7**

GPIB

BI	x-ratio,y-ratio

4050 BASIC (Data Byte Form)

	PRINT @ pa,32: **"BI"**;	x-ratio;y-ratio
Sample:	**PRINT @ 1,32: "BI";**	**2;l.7**

Parameters

x-ratio
Character space divided by character width

y-ratio
Line space divided by character height

Comments

The default x-ratio and y-ratio values are 3/2 and 18/11, respectively. The product of x-ratio (or y-ratio) and character width (or height) is used to determine the character space (or line space) distance. These distances are used by subsequent CHARACTER MOVE commands and PRINT commands for determining the next character position.

A negative x-ratio will cause a reversal in direction for PRINT command advance-to-next-character actions, and for ^{B}S and ^{H}T actions. For instance, an x-ratio of −1.5 (assuming default alpha spacing control) will cause printed characters to be positioned in a row from right to left; the printing of characters will NOT be reversed.

A negative y-ratio will reverse the direction of PRINT ^{L}F and ^{V}T actions.

Examples

The samples in the syntax boxes show how to specify a character space (the distance from the start of one typical uppercase letter to the start of the next) that is two times the character width, and a line space (the distance from the baseline for one line of printed characters to the next baseline above it) that is 1.7 times the character height.

SET ALPHA DIMENSION

Purpose: This command sets the width and height for a typical uppercase letter (such as A). The Plotter divides the specified character space and line space by the current alpha ratio values to determine character width and height.

SERIAL

	$^{A}_{T}$ $^{A}_{D}$ **I**	character-space,line-space
Sample:	$^{E}_{C}$ **A I**	**18,36**

GPIB

S	character-space,line-space

4050 BASIC (Data Byte Form)

	PRINT @ pa,32: **"S"**;	character-space;line-space
Sample:	**PRINT @1,32: "S";**	**18;36**

4050 BASIC (Secondary Address Form)

	PRINT @ pa,17:	character-space;line-space
Sample:	**PRINT @1,17:**	**18;36**

Parameters

character-space
Distance from the beginning of a typical uppercase letter to the beginning of the next, in Initial World Units

line-space
Distance from the baseline for one line of printed characters to the next baseline above it, in Initial World Units

Comments

This is an alternate form of the SET ALPHA SIZE command.

Examples

The samples in the syntax boxes above set arguments of 18 and 36 which, for default alpha ratio values 3/2 and 18/11, are equivalent to arguments of 12 and 22 in a SET ALPHA SIZE command.

SET ALPHA ROTATION

Purpose: This command (1) sets the alpha rotation angle specified in degrees and (2) sets the left margin (as a line that goes through the current pen carriage position and is perpendicular to the printing direction) used by subsequent alpha action commands.

SERIAL

	$^{A}_{T}$ $^{A}_{D}$ **J**	rotation-angle
Sample:	$^{E}_{C}$ **A J**	**30**

GPIB

R	rotation-angle

4050 BASIC (Data Byte Form)

	PRINT@pa,32: **"R"**;	rotation-angle
Sample:	**PRINT@1,32: "R";**	**30**

4050 BASIC (Secondary Address Form)

	PRINT@pa,25:	rotation-angle
Sample:	**PRINT@1,25:**	**30**

Parameters

rotation-angle
The angle of rotation counterclockwise, in degrees

Comments

The default rotation angle is 0°, and the default left margin is the left side of the Window.

NOTE

The SET WINDOW command also sets the left margin as a line that goes through the upper left corner of the Window and is perpendicular to the printing direction.

The left margin is used for (1) carriage return, (2) margin control (if active), and (3) as a tab spacing reference. For details on margin control and tab spacing, see the descriptions for the SET TAB SEPARATION and SET MARGIN SEPARATION commands.

Examples

The samples in the syntax boxes show how to specify a 30° counterclockwise rotation angle, and how to set the left margin at the current pen carriage position. Figure 10-8 shows the effect of a 30° alpha rotation on subsequent PRINT command actions.

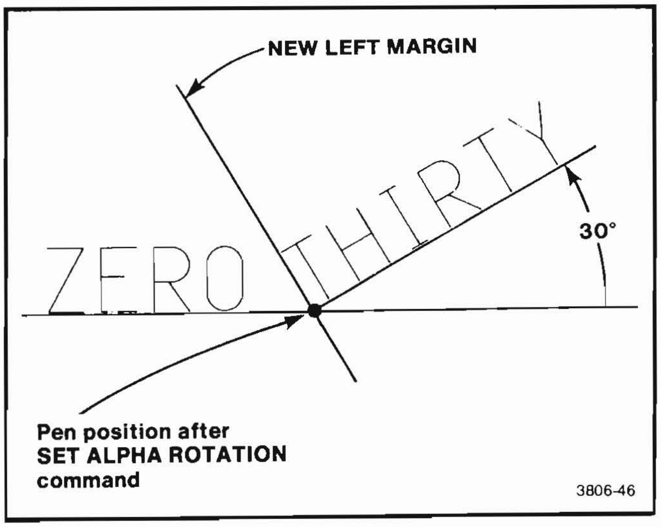


Figure 10-8. Alpha Rotation of 0° and 30°.

SET ALPHA SLANT

Purpose: This command sets the alpha slant angle in degrees used by subsequent PRINT CENTERED and PRINT commands (for noncontrol characters only).

SERIAL

	AT AD **BG**	slant-angle
Sample:	EC **A BG**	**20**

GPIB

	BG	slant-angle

4050 BASIC (Data Byte Form)

	PRINT @ pa,32: **"BG"**;	slant-angle
Sample:	**PRINT @ 1,32: "BG";**	**20**

Parameters

slant-angle
The clockwise slant angle, in degrees (default value is 0°)

Comments

Subsequent PRINT or PRINT CENTERED commands produce characters with all portions above and below the baseline slanted clockwise the specified angle from a vertical axis. Character height does not change.

The slant angle is not used for CHARACTER MOVE command actions or for control character (for example, LF and VT) actions of PRINT commands.

Angles from 89.9° to 90.1° and from 269.9° to 270.1° are converted to 89.9°.

Examples

The samples in the syntax boxes show how to specify a 20° clockwise slant angle for subsequent PRINT or PRINT CENTERED command actions. Figure 10-9 shows the effect of a 20° alpha slant on subsequent PRINT command actions.

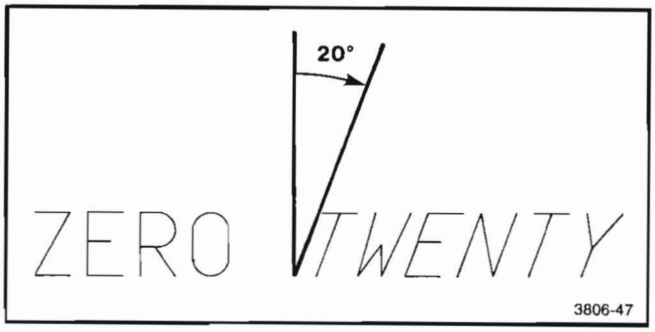


Figure 10-9. Alpha Slant of 0° and 20°.

SET ALPHA SCALE

Purpose: This command sets the X-axis and Y-axis magnification factors used by all subsequent alpha action commands.

SERIAL

	$^{A}{}_{T}$ $^{A}{}_{D}$ **BH**	x-mag-factor,y-mag-factor
Sample:	$^{E}{}_{C}$ **A BH**	**2,3**

GPIB

BH	x-mag-factor,y-mag-factor

4050 BASIC (Data Byte Form)

	PRINT@ pa,32: **"BH"**;	x-mag-factor;y-mag-factor
Sample:	**PRINT@ 1,32: "BH";**	**2;3**

Parameters

x-mag-factor
X-axis magnification factor

y-mag-factor
Y-axis magnification factor

Comments

Default values for magnification factors are 1,1. Negative values reverse the printing and spacing associated with alpha action commands.

Examples

The samples in the syntax boxes above cause all subsequent printing and the associated spacing between characters to be twice as wide and three times as high as produced with default parameters for this command.

SET TAB SEPARATION

Purpose: This command activates the horizontal tab function and sets the distance between tabs (and from the left margin to the first tab position) equal to the specified number of spaces of the current size. This information is used by subsequent PRINT commands containing the ᴴᴛ character. A value of zero (default) disables the tab function and causes the ᴴᴛ character to be printed as a ˢᴘ character.

SERIAL

	ᴬᴛ ᴬᴅ **BT**	tab-separation
Sample:	ᴱᴄ **A BT**	**8**

GPIB

	BT	tab-separation

4050 BASIC (Data Byte Form)

	PRINT@ pa,32: **"BT"**;	tab-separation
Sample:	**PRINT**@ **1,32: "BT";**	**8**

Parameters

tab-separation
Tab separation, as the number of character spaces, sign ignored

Comments

The horizontal tab separation defines a series of equally spaced tab positions (stops) starting one tab separation distance to the right of the left margin. For each ᴴᴛ character in subsequent PRINT commands, the Plotter moves along the current printing direction to the next tab position beyond the current pen carriage position.

The tab separation distance is set according to the current alpha size, alpha ratio, and alpha scale parameters. Subsequent changes in the alpha size, alpha dimension, alpha ratio, or alpha scale specifications do not modify tab separation. This means that character size and spacing parameters can be changed without affecting previously defined tab positions.

Examples

The samples in the syntax boxes show how to set a tab separation of eight spaces of the current size, which defines a series of tab positions that are 8, 16, 24, etc. spaces from the left margin.

SET ALPHA MARGIN SEPARATION

Purpose: This command activates the margin control function and sets the right margin a specified distance from the left margin. MARGIN CONTROL is used for subsequent PRINT and CHARACTER MOVE command actions. A margin separation of zero (default) disables margin control.

SERIAL

	$^{A}{}_{T}$ $^{A}{}_{D}$ **BR**	margin-separation
Sample:	$^{E}{}_{C}$ **A BR**	**50**

GPIB

BR	margin-separation

4050 BASIC (Data Byte Form)

	PRINT@ pa,32: **"BR"**;	margin-separation
Sample:	**PRINT@ 1,32: "BR"**;	**50**

Parameters

margin-separation
Distance between margins, as number of character spaces, sign ignored

Comments

The alpha margin separation distance defines a right margin that is parallel to and to the right of the left margin. Subsequent changes to the left margin will cause corresponding changes to the right margin. (For details on how to set the left margin, see the description of the SET ALPHA ROTATION command.)

The alpha margin separation distance is set according to the current alpha size, alpha ratio, and alpha scale parameters. Subsequent changes in the alpha size, alpha dimension, alpha ratio, or alpha scale specifications will not modify the margin separation. This means that character size and spacing parameters can be changed without affecting previously defined margins.

When the alpha margin separation is set to be non-zero, subsequent PRINT commands obey the following rules for each character:

1. For printable ASCII characters ($^{S}{}_{P}$ through $^{D}{}_{L}$) only, if the current pen position is to the right of the right margin, or if a typical uppercase letter would extend beyond the right margin, the alpha generator performs a PRINT $^{C}{}_{R}$ $^{L}{}_{F}$ command prior to printing the specified character. If the current pen position is to the left of the left margin, the alpha generator performs a PRINT $^{C}{}_{R}$ command prior to printing the specified character.
2. The alpha generator prints the specified character and moves to the origin of the next character.
3. If the new pen carriage position is to the right of the right margin, the alpha generator performs a PRINT $^{C}{}_{R}$ $^{L}{}_{F}$ command.

For the CHARACTER MOVE command, the alpha generator performs margin control according to the third step.

References

For information on how to set the left margin, see the description of the SET ALPHA ROTATION command.

SELECT ALPHA SPACING CONTROL

Purpose: This command selects the type of alpha spacing control used by subsequent PRINT commands. A selection of zero (the default) produces a constant start-to-start spacing of characters. A selection of one produces a constant end-to-start spacing of characters.

SERIAL

```
           AT AD AJ                       spacing-control-selector
  Sample:  EC A AJ                        1
```

GPIB

```
           AJ                             spacing-control-selector
```

4050 BASIC (Data Byte Form)

```
           PRINT@pa,32: "AJ";             spacing-control-selector
  Sample:  PRINT@1,32: "AJ";              1
```

Parameters

spacing-control-selector
Spacing control selector, 0-1:

- 0 = constant start-to-start
- 1 = constant end-to-start

Comments

Refer back to Figure 10-7. For the default selection of zero, subsequent print actions assume that all characters are of equal width (that of a typical uppercase letter, such as A). Narrow characters (for example, i) are printed near the center of the allocated character width. With ALPHA SPACING CONTROL set to one, the space allocated to each character varies such that the distance between characters is constant. This space is equal to the distance between uppercase H and M.

References

See *Spacing Control for the PRINT Command*, earlier in this section.

SELECT STANDARD ALPHA FONT

Purpose: This command selects the standard font that, when active, is used by subsequent PRINT or PRINT CENTERED commands.

SERIAL

	$^{A}{}_{T}$ $^{A}{}_{D}$ **T**	standard-font-selector
Sample:	$^{E}{}_{C}$ **A T**	**1**

GPIB

	F	standard-font-selector

4050 BASIC (Data Byte Form)

	PRINT@pa,32: **"F"**;	standard-font-selector
Sample:	**PRINT@1,32: "F";**	**1**

4050 BASIC (Secondary Address Form)

	PRINT@pa,18:	standard-font-selector
Sample:	**PRINT@I,18:**	**1**

Parameters

standard-font-selector
Standard font selector, 0-15

Comments

This command selects the standard font, which is activated by control characters within a PRINT command:

- When the Plotter receives an ASCII $^{S}{}_{O}$ (Shift Out) control character, the alternate font is activated.
- When the Plotter receives an ASCII $^{S}{}_{I}$ (Shift In) control character, the standard font is activated. The default active font is the standard font.

NOTE

Serial interface only: For Style-II command applications, use $^{A}{}_{T}$ $^{S}{}_{O}$ and $^{A}{}_{T}$ $^{S}{}_{I}$ sequences to active the alternate or standard font (or use Style-I PRINT commands). Any $^{S}{}_{O}$ or $^{S}{}_{I}$ characters received in Alpha mode will be ignored.

SELECT ALTERNATE ALPHA FONT

Purpose: This command selects the alternate font that, when active, is used by subsequent PRINT or PRINT CENTERED commands.

SERIAL

	$^{A}_{T}$ $^{A}_{D}$ **BQ**	alternate-font-selector
Sample:	$^{E}_{C}$ **A BQ**	**2**

GPIB

	BQ	alternate-font-selector

4050 Basic (Data Byte Form)

	PRINT @pa,32: **"BQ"**;	alternate-font-selector
Sample:	**PRINT** @1,32: **"BQ"**;	**2**

Parameters

alternate-font-selector
Alternate font selector, 0-15

Comments

This command selects the alternate font, which is activated by control characters within a PRINT command:

- When the Plotter receives an ASCII $^{S}_{O}$ (Shift Out) control character, the alternate font is activated.
- When the Plotter receives an ASCII $^{S}_{I}$ (Shift In) control character, the standard font is activated. The default active font is the standard font.

NOTE

Serial interface only: For Style-II command applications, use $^{A}_{T}$ $^{S}_{O}$ and $^{A}_{T}$ $^{S}_{I}$ sequences to active the alternate or standard font (or use Style-I PRINT commands). Any $^{S}_{O}$ or $^{S}_{I}$ characters received in Alpha mode will be ignored.

References

See the *Resident Alpha Fonts* discussion earlier in this section for descriptions of available fonts. See the SELECT STANDARD ALPHA FONT command for information on how to select the standard font.

RESET ALPHA PARAMETERS

Purpose: This command resets alpha parameters to their default (power-up) values.

SERIAL

$^{A}_{T}$ $^{A}_{D}$ **V**

Sample: $^{E}_{C}$ **A V**

GPIB

A

4050 Basic (Data Byte Form)

PRINT @pa,32: **"A"**

Sample: **PRINT @1,32: "A"**

4050 BASIC (Secondary Address Form)

PRINT @pa,7:

Sample: **PRINT @1,7:**

Comments

The RESET ALPHA PARAMETER command establishes default parameters for the alpha setup commands shown in Table 10-4.

This command does not affect values set by the SET ARC SMOOTHNESS command and does not affect the Parameter Entry Card selection for Alpha Character Quality.

Table 10-4

DEFAULT PARAMETERS FOR RESET ALPHA PARAMETER COMMAND[a]

Command	Default Parameter
SET ALPHA SIZE	(Set to the default size listed in Table 10-3.)
SET ALPHA RATIO	3/2, 18/11
SET ALPHA ROTATION	0°
SET ALPHA SLANT	0°
SET ALPHA SCALE	1, 1
SELECT STANDARD ALPHA FONT	0
SELECT ALTERNATE ALPHA FONT	0
SET ALPHA MARGIN SEPARATION	0 (Margin control inactive.)
SET TAB SEPARATION	0 ($^{H}_{T}$ treated as $^{S}_{P}$)
SELECT ALPHA SPACING CONTROL	0 (Constant spacing between character origins.)

[a] Left margin is set at the left edge of the Viewport. Standard alpha font is activated.

PRINT

Purpose: This command causes the specified ASCII characters to be printed, according to all alpha parameters currently defined. After each character is printed, the pen carriage moves to the origin for the next character.

SERIAL

	$^{A}{}_{T}$ $^{A}{}_{D}$ **P**	string
Sample:	$^{E}{}_{C}$ **A P**	**“ABC”/68′EFG′**

GPIB

	P	print-string

4050 Basic (Data Byte Form)

	PRINT @pa,32: **“P**	print-string” ;
Sample:	**PRINT @1,32: “P**	**ABCDEFG” ;**

4050 BASIC (Secondary Address Form)

	PRINT @pa,12:	“print-string” ;
Sample:	**PRINT @1,12:**	**“ABCDEFG” ;**

4050 BASIC (Special Keyword Form)

	PRINT @pa[,12]:	“print-string” ;
Sample:	**PRINT @1:**	**“ABCDEFG” ;**

Parameters

print-string or *string*

Represents one or more ASCII characters:

- $^{B}{}_{S}$ through $^{S}{}_{I}$ and
- $^{S}{}_{P}$ through $^{D}{}_{L}$

(Other control characters are ignored)

Comments

The PRINT command is used to print any of the 96 ASCII characters $^{S}{}_{P}$ through $^{D}{}_{L}$ and/or to process control characters $^{B}{}_{L}$ through $^{S}{}_{I}$. (For most fonts, $^{S}{}_{P}$ and $^{D}{}_{L}$ are non-printing and $^{D}{}_{L}$ is non-advancing.) The PRINT command prints each character according to current alpha setup command parameters. Control characters are processed as listed in Table 10-5 (other control characters are ignored). Control character processes are not affected by the SET ALPHA SLANT command or by character defining commands provided by Option 32.

Table 10-5

CONTROL CHARACTER RESPONSES

Character	Action
$^{B}{}_{L}$	Rings the bell.
$^{B}{}_{S}$	Moves the pen carriage left one character space.
$^{H}{}_{T}$	Moves the pen carriage to the next tab position to the right. If the tab separation is 0, $^{H}{}_{T}$ is treated as a $^{S}{}_{P}$.
$^{L}{}_{F}$	Moves the pen carriage down one line space.
$^{V}{}_{T}$	Moves the pen carriage up one line space.
$^{F}{}_{F}$[a]	Moves the pen carriage to the home position.
$^{C}{}_{R}$	Moves the pen carriage to the left margin (and generates a $^{L}{}_{F}$ if CR GENERATES LF is selected on the Parameter Entry Card).
$^{S}{}_{O}$[a]	Activates the Alternate Font. (See the SELECT ALTERNATE ALPHA FONT command.)
$^{S}{}_{I}$[a]	Activates the Standard Font. (See the SELECT STANDARD ALPHA FONT command.)

[a] **For Style-II command applications only: To obtain the equivalent action in Alpha mode, use the appropriate attention character sequence ($^{A}{}_{T}{}^{F}{}_{F}$, $^{A}{}_{T}{}^{S}{}_{O}$, or $^{A}{}_{T}{}^{S}{}_{I}$). Individual $^{F}{}_{F}$, $^{S}{}_{O}$, or $^{S}{}_{I}$ characters received in Alpha mode will be ignored.**

References

Special considerations are necessary to ensure that strings are properly delimited and/or terminated. For specific information, see information pertaining to strings in Sections 4 and 5.

Examples

The samples in the syntax boxes show how to print the first seven letters of the alphabet: ABCDEFG.

PRINT CENTERED

Purpose: This command causes ASCII character(s) to be printed and centered about the current pen position. After each character is printed, the pen carriage moves back to its original position.

SERIAL

	AT AD **AP**	string
Sample:	EC **A AP**	**"O"**

GPIB

AP	print-string

4050 Basic (Data Byte Form)

	PRINT @ pa,32: **"AP**	print-string" ;
Sample:	**PRINT @1,32: "AP**	**O"** ;

Parameters

print-string or string

Represents one (or more) ASCII characters:

- SP through DL (Control characters are ignored)

Comments

The PRINT CENTERED command is often used for labeling points on a graph. Characters should be limited to uppercase letters and other full-width symbols that have the same center as a typical uppercase letter. See the top portion of Figure 10-10. Two or more characters may be combined to form a new symbol (for example, the characters "N" and "Z" can be used to form an X with a box around it as shown in the bottom portion of Figure 10-10). As a recommendation, use the default selection for the ALPHA SPACING CONTROL command when using the PRINT CENTERED command.

References

Special considerations are necessary to ensure that print strings are properly delimited and terminated. For specific information, see the topics pertaining to strings in Sections 4 and 5.

Examples

The samples in the syntax boxes cause the uppercase letter "O" to be printed and centered about the current pen position.

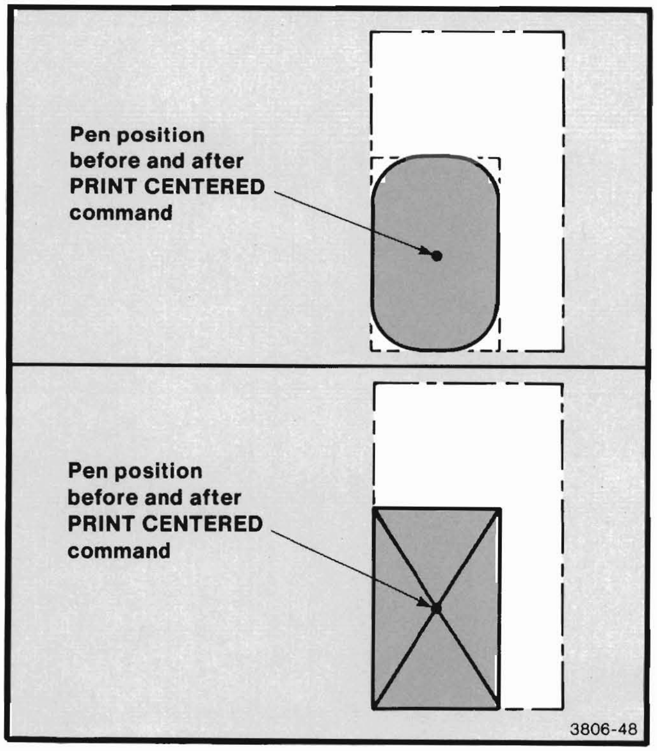

Pen position before and after PRINT CENTERED command
Pen position before and after PRINT CENTERED command
3806-48

Figure 10-10. PRINT CENTERED Command.

MOVE TO HOME

Purpose: This command moves the pen carriage to the home position (a point located one character below the upper-left corner of the Viewport along a line perpendicular to the current printing direction).

SERIAL

$^{A}{}_{T}$ $^{A}{}_{D}$ **AH**

Sample: $^{E}{}_{C}$ **A AH**

GPIB

H

4050 Basic (Data Byte Form)

PRINT @pa,32: "H"

Sample: **PRINT @1,32: "H"**

4050 BASIC (Secondary Address Form)

PRINT @pa,23:

Sample: **PRINT @I,23:**

4050 BASIC (Special Keyword Form)

HOME @pa[,23]:
or
PAGE @pa[,22]:

Samples: **HOME @1:**
or
PAGE @1:

Comments

This command does not cause a Plotter media change action.

Examples

The samples in the syntax boxes show how to move the pen carriage to the home position.

For a Page keyword shown in the last syntax box, the Plotter substitutes a Move-To-Home action.

CHARACTER MOVE

Purpose: This command causes the pen carriage to move the specified number of character spaces (right) and line spaces (up).

SERIAL

	AT AD **AM**	character-spaces,line-spaces
Sample:	EC **A AM**	**3,2**

GPIB

AM

4050 Basic (Data Byte Form)

	PRINT@ pa,32: **"AM";**	character-spaces;line-spaces
Sample:	**PRINT@ 1,32: "AM";**	**3;2**

Parameters

character-spaces
Number of character spaces to move right

line-spaces
Number of line spaces to move up

Comments

Negative parameter values reverse the direction of the move. When margin control is active and the CHARACTER MOVE command moves the pen carriage to the right of the right margin, the Plotter automatically follows this command with a PRINT CR LF command.

References

Margin control is discussed at the front of this section and under the SET MARGIN SEPARATION command.

Examples

The samples in the syntax boxes show how to move the pen carriage to the right three character spaces and up two line spaces from the current position.

Section 11

GRAPHICS COMMANDS

CONCEPTS

The Graphics commands described in this section fall into the following categories:

- Selection and setup commands.
- Line type selection commands.
- Commands that reposition the pen carriage or draw lines.
- A command for reporting a setup.
- Commands for plotting arcs and circles.

GRAPHICS SELECTION AND SETUP COMMANDS

While many set-up and selection options exist, the initial state of the Plotter when powered up makes the Plotter appear as a terminal screen when operating with a Serial interface and as a 4050 Series screen when operating with a GPIB interface.

Initial states were discussed in Section 1; Table 11-1 summarizes the initial states for the selection and set-up commands in this section. The table includes the command names, the choices available, and the initial power-up or default state.

Typically these setup and selection commands are modified to match the application. For example:

- SELECT DEVICE UNITS command can be changed to millimeters (mm).
- SET VIEWPORT command can be set to match a desired plot size.
- SET WINDOW command can be set to match the user's World Coordinates.

In addition, some applications require Graphic Units to be selected as Device Units or the SELECT CLIPPING CONTROL selection may need to be changed to Page. Other selections are more dependent on the subject matter of the plot and can be changed during a plot.

Table 11-1

GRAPHICS SELECTION AND SETUP COMMANDS

Command Name	Choices	Default State
SELECT GRAPHIC UNITS	World, Device	World
SELECT DEVICE UNITS	ADUs, GDUs, mm	ADUs for Serial interface, GDUs for GPIB interface
SET VIEWPORT	Available using front panel, command setting, or Parameter Entry Card	Initial Aspect Ratio selection on Parameter Entry Card
SET WINDOW	Available using command settings	Rules for ranges described in Section 1
SELECT PEN	Crosshair Cursor, Pen 1, Pen 2	Pen 1
SELECT COORDINATE TYPE	Absolute, Relative	Absolute
SELECT CLIPPING CONTROL	Viewport, Page	Viewport
SELECT LINE TYPE	Solid, fixed dash-pattern-length, variable dash-pattern-length point	Solid

LINE TYPE SELECTION COMMANDS

These commands select graphic line types:

- SELECT LINE TYPE
- SET DASH PATTERN
- SET DASH PATTERN LENGTH

Each command is discussed here.

SELECT LINE TYPE Command

Four graphic line types are available and are illustrated in Figure 11-1. They include solid, fixed dash-pattern-length, variable dash-pattern-length, and point.

Typically these set-up and selection commands are modified to match the application. For example:

- SELECT DEVICE UNITS command can be changed to millimeters (mm).
- SET VIEWPORT command can be set to match a desired plot size.
- SET WINDOW command can be set to match the user's World Coordinates.

In addition, some applications require Graphic Units to be selected as Device Units or the SELECT CLIPPING CONTROL selection may need to be changed to Page. Other selections are more dependent on the subject matter of the plot and can be changed during a plot.

Solid is the default line type selection that occurs at power-up and after the DEVICE RESET command. Certain operations always use the solid line type. These operations include: MARK VIEWPORT, OUTLINE VIEWPORT, AXIS, and PRINT alphanumeric characters.

The *point* line type selection causes a point (dot) at the end point of each line drawn.

The other two line type selections, *fixed* and *variable dash-pattern-length,* cause dashed and/or dotted lines when used with the SET DASH PATTERN and SET DASH PATTERN LENGTH commands. When fixed dash-pattern-length is selected, a given pattern begins at the start of the first line and is repeated as often as necessary to complete the line. Any portion of the pattern that is not used in the first line is used in the next connected line. This process continues for each successive connected line. If necessary, the pattern bends around corners. Unlike the variable dash-pattern-length line type, the fixed dash-pattern-length line type uses a constant (fixed) pattern length. The pattern is started or restarted again after a MOVE command or any SELECT LINE TYPE, SET DASH PATTERN, or SET DASH PATTERN LENGTH command.

The variable dash-pattern-length line type automatically reduces the pattern length (if necessary) so that the smallest whole number (one or more) of patterns fits along the line drawn. For each DRAW command, the reduced pattern length may be different.

If Option 31 is installed on your Plotter, you can use all four line types for drawing arcs and circles. The most useful line types are solid and fixed dash-pattern-length. The other two types create patterns using the actual chords of the arc or circle as a sequence of curved line segments.

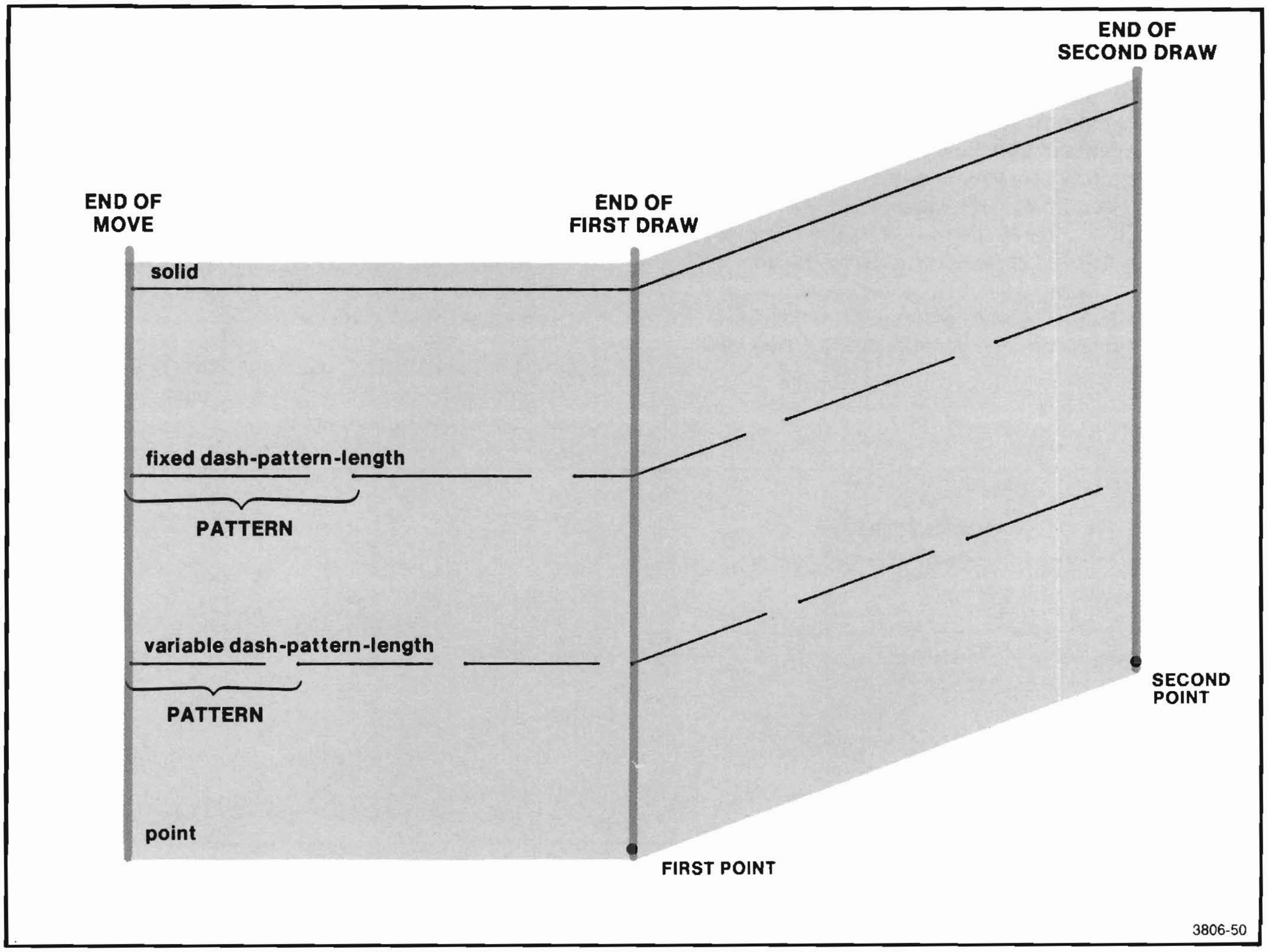


Figure 11-1. Four Line Types Shown Using Two Successive Draws.

SET DASH PATTERN Command

For both fixed and variable dash-pattern-length line types, the SET DASH PATTERN command establishes the dash/dot pattern. This pattern is specified by 1 to 20 pattern elements consisting of integers from 0 to 255. Each number represents a relative length of an element on the pattern. This means that the length of each element relative to the pattern is the numeric value of that number element divided by the sum of the numeric value of all of the number elements in the pattern. The actual pattern length is set using the SET DASH PATTERN LENGTH command described later.

The rules for setting pattern elements are as follows:

1. The first element is a DRAW.
2. Pattern elements are alternately DRAWs and MOVEs.
3. If the first entry is 0, the DRAW is ignored and the pattern begins with the next element, which is identified as a MOVE.
4. If 0 is assigned to a DRAW element (other than the first element), that element will be drawn as a point.
5. If 0 is assigned to a MOVE element, the two adjacent DRAWs will touch.

Figure 11-2 illustrates these rules. Note that the bottom pattern contains three successive 0 elements to produce a dot at the beginning.

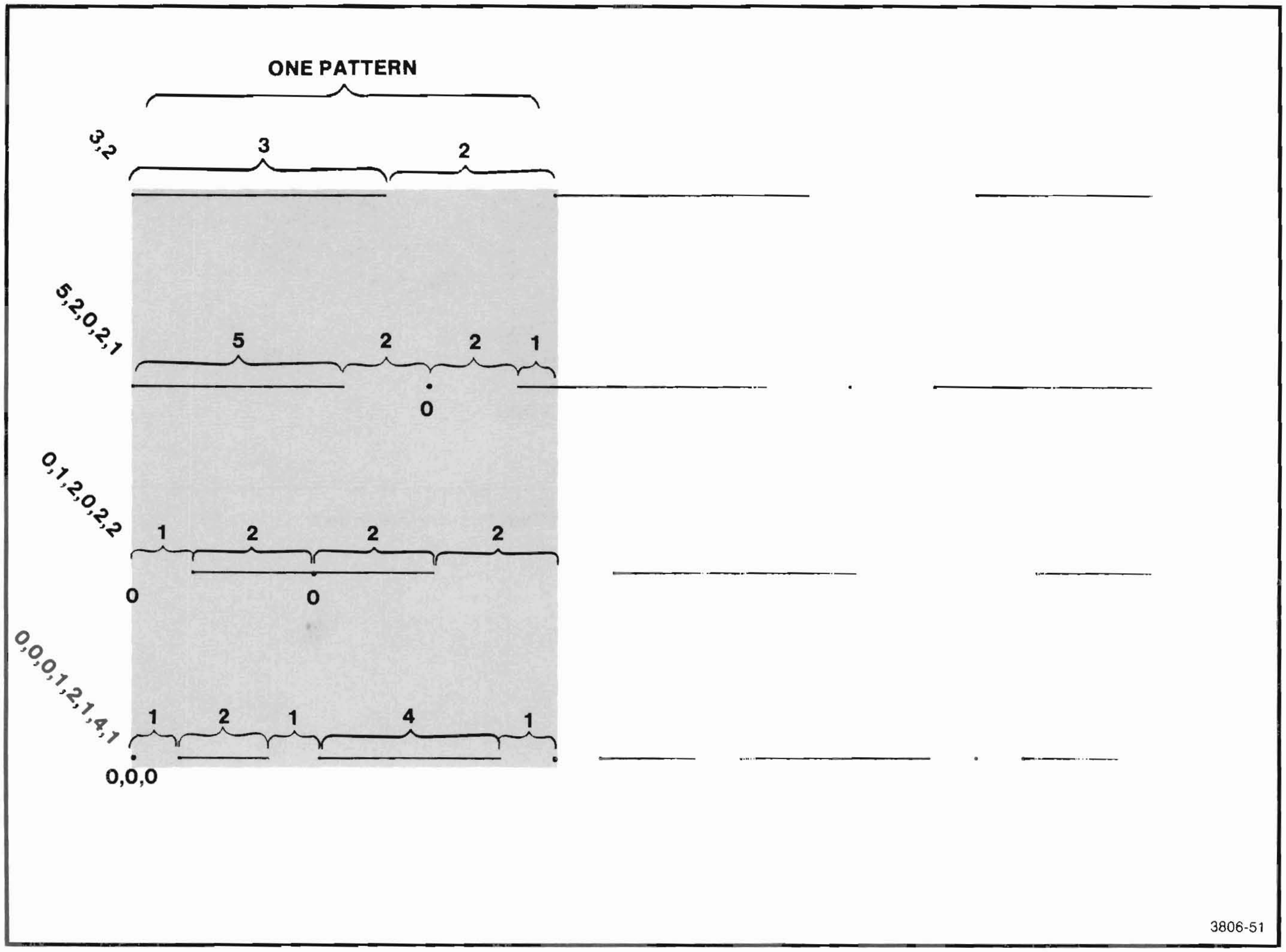


Figure 11-2. Sample Dash Patterns.

SET DASH PATTERN LENGTH Command

This command sets the fixed and the maximum dash pattern length for dash-pattern-length line types. The length is always specified in Initial World Units along the X-axis, even though Window settings may cause the actual World Units to be different. The actual physical pattern length tracks X-coordinate Page size and Viewport changes.

Extremely short pattern lengths (including the default length of 0) cause solid lines to be produced with DRAW commands. The default length of 0 is in effect at power-up and after the DEVICE RESET command.

PEN CARRIAGE POSITIONING AND LINE DRAWING COMMANDS

The MOVE and DRAW commands are commonly used to position the pen carriage and to draw lines. The commands listed below perform additional special operations:

- MOVE TO LOAD POINT
- OUTLINE VIEWPORT
- MARK VIEWPORT
- AXIS

A SETUP REPORTING COMMAND

The READ VIEWPORT command reports the coordinates of the Viewport and can be useful when setting a new Window to match the existing Viewport aspect ratio.

ARCS AND CIRCLES (REQUIRES OPTION 31)

With Option 31 installed, the following three additional Graphics commands are available:

- DRAW ARC
- DRAW CIRCLE
- SET ARC SMOOTHNESS

The main characteristics of these commands are discussed here.

The Plotter can draw approximations of arcs and circles using a series of straight lines that form chords. The accuracy of these approximations is adjustable using a programmable smoothness parameter in the SET ARC SMOOTHNESS command.

Keep in mind the following characteristics:

- DRAW ARC and DRAW CIRCLE commands are designed to work only with the Graphic Units selection set to World Units (and not Device Units).
- Transforms affect arcs and circles just as they affect characters and lines. Thus, arcs and circles can be translated, scaled, rotated, and skewed.
- You can use different line types (solid, fixed and variable dash-pattern-length, and point) with these commands.
- Arguments for arcs are interpreted as absolute coordinates even if relative coordinates are selected for MOVE and DRAW commands.

DRAW ARC Command

Three points specify the arc in the DRAW ARC command. The first point is the current position, while the intermediate and end points are provided by arguments for the command. If the end point and first point are the same, a circle is drawn that has a diameter defined by the first and intermediate points.

DRAW CIRCLE Command

The DRAW CIRCLE command draws a circle at the current position with a radius specified by a command argument. Upon completion, the pen carriage returns to the center of the drawn circle.

SET ARC SMOOTHNESS Command

The SET ARC SMOOTHNESS command allows you to adjust the visual appearance of arcs, circles, and alphanumeric characters with curved components. The smoothest approximations to curves have many chords, but take much longer to draw than the coarser approximations.

The minimum chord length is determined by internal parameters related to the physical precision of the Plotter. This occurs when the smoothness factor is set to 0. Use coarse approximations for previewing and prototyping purposes. However, no chord subtends an angle greater than 45°. This limitation occurs when the smoothness factor is set to 1, or occasionally a higher value for large curves. To retain the same visual appearance of smoothness, curves with larger radii must have more chords in their approximation.

The default smoothness of 0.5 provides acceptably smooth curves for most applications. However, the smoothness setting over the 0 to 1 range as related to visual smoothness is very nonlinear, especially from 0.5 to 1 and beyond. Figure 11-3 shows arcs drawn with smoothness settings from 0 to 1. The differences between 0 and 0.5 settings are more distinguishable when the radii are larger.

Because smoothness is related to physical lengths, the actual number of chords is a function of Page, Viewport, and Window selections. A first order adjustment is made when the physical World Unit lengths are different in the X and Y directions. However, adjustments are not made for transformations such as scaling and skewing.

Figure 11-4 shows how the arcs are approximated by a number of equal length chords. Note that the selected intermediate point does not necessarily lie on a chord endpoint.

When three points of an arc lie on a straight line (and the first and last are not the same as in a circle), a line is drawn from the first point to the last point that was specified. In some cases, however, internal numerical limitations of the Plotter may cause an arc with a very large radius to be plotted instead of a straight line.

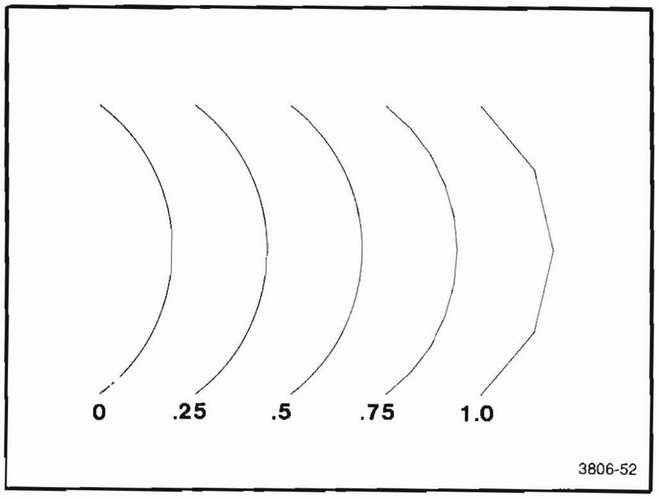


Figure 11-3. Arcs With Different Smoothness Settings.

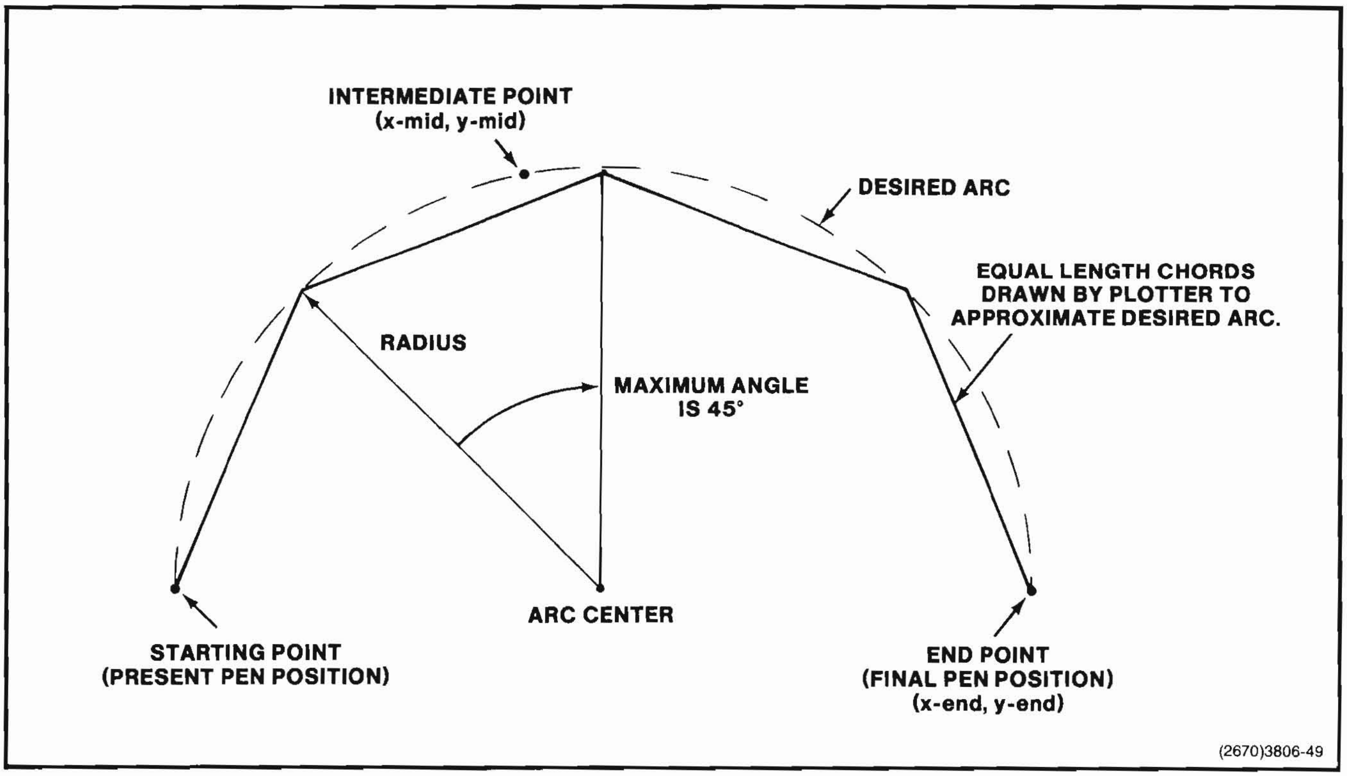


Figure 11-4. An Arc Approximated by Chords.

COMMAND DESCRIPTIONS

There are 20 commands described in this section:

- SELECT GRAPHIC UNITS
- SELECT DEVICE UNITS
- SET VIEWPORT
- SET WINDOW
- SELECT PEN
- SELECT COORDINATE TYPE
- SELECT CLIPPING CONTROL
- SELECT LINE TYPE
- SET DASH PATTERN
- SET DASH PATTERN LENGTH
- MOVE
- DRAW
- MOVE TO LOAD POINT
- OUTLINE VIEWPORT
- MARK VIEWPORT
- AXIS
- READ VIEWPORT
- SET ARC SMOOTHNESS
- DRAW ARC
- DRAW CIRCLE

These commands are described in the following pages.

SELECT GRAPHIC UNITS

Purpose: This command selects the Graphic Units (Device or World Units) for plotting (Moves and Draws) and for digitizing (sometimes referred to as graphic input).

SERIAL

	$^{A}_{T}$ $^{A}_{D}$ **BW**	input-selector[,output-selector]
Sample:	$^{E}_{C}$ **A BW**	**0,0**

GPIB

BW	input-selector[,output-selector]

4050 BASIC (Data Byte Form)

	PRINT @pa,32: **"BW"**;	input-selector[;output-selector]
Sample:	**PRINT @1,32: "BW";**	**1;1**

Parameters

input-selector
Graphic Units for Moves and Draws selector, 0-1:
- 0 = for World Units
- 1 = for Device Units

output-selector
Graphic Units for digitizing selector, 0-1:
- 0 = for World Units
- 1 = for Device Units

Omitted means same as input selector

Comments

Both the input-selector and output-selector parameters default to World Units under the following conditions:

- At Plotter power-up
- After a DEVICE RESET command

Examples

The samples shown in the syntax boxes illustrate how to set input and output selections to World Units for Serial applications and to Device Units for 4050 BASIC applications.

The SERIAL sample can be shortened to:

$^{E}_{C}$ **A BW 0**

The 4050 BASIC (Data Byte Form) sample can be shortened to:

PRINT @1,32: "BW"; 1

Input and output selectors can be different. For example:

$^{E}_{C}$ **A BW 0,1**
PRINT @1,32: "BW"; 1,0

SELECT DEVICE UNITS

Purpose: This command allows you to select three types of Device Units: ADUs, GDUs, or mm.

SERIAL

	$^{A}{}_{T}$ $^{A}{}_{D}$ **BV**	device-units-selector
Sample:	$^{E}{}_{C}$ **A BV**	**1**

GPIB

BV	device-units-selector

4050 BASIC (Data Byte Form)

	PRINT @pa,32: **"BV";**	device-units-selector
Sample:	**PRINT @1,32: "BV";**	**0**

Parameters

device-units-selector
Device Units selector, 0-2:

- 0 = ADUs
- 1 = GDUs
- 2 = mm

Comments

The default Device Unit selection depends upon the active interface. ADUs are the default for Serial applications, while GDUs are the default for GPIB applications. The Plotter automatically sets to the default under these conditions:

- Upon power-up
- After a DEVICE RESET command
- After the SELECT COMMAND/RESPONSE FORMAT command
- After the INTERFACE PARAMETER RESET command

Examples

The Serial applications sample sets Device Units to GDUs (from a possible ADU default setting). The 4050 BASIC (Data Byte Form) sample sets Device Units to ADUs (from a possible GDU default setting). Examples below set the Device Units to mm:

$^{E}{}_{C}$ **A BV 2**
PRINT @1,32: "BV"; 2

SET VIEWPORT

Purpose: This command sets the Viewport.

SERIAL

	${}^{A}{}_{T}$ ${}^{A}{}_{D}$ **AV**	llx,urx,lly,ury
Sample:	${}^{E}{}_{C}$ **A AV**	**1000,3000,500,2500**

GPIB

	AV	llx,urx,lly,ury

4050 BASIC (Data Byte Form)

	PRINT @pa,32: **"AV"**;	llx;urx;lly;ury
Sample:	**PRINT @1,32: "AV";**	**20;120;0;100**

Parameters

llx
X-coordinate of the lower-left corner of the Viewport, in Device Units

urx
X-coordinate of the upper-right corner of the Viewport, in Device Units

lly
Y-coordinate of the lower-left corner of the Viewport, in Device Units

ury
Y-coordinate of the upper-right corner of the Viewport, in Device Units

Comments

An Initial Viewport is established at power-up, after a DEVICE RESET command (0, 1, or 3), or by making a selection on any of the following Parameter Entry Card lines:

- Media Form
- Initial Page Size
- Initial Page Format
- Page Orientation
- Initial Aspect Ratio

The following restrictions apply to the SET VIEWPORT command:

- The command resets all transform parameters.
- The command does not affect Window coordinates, but may change the aspect ratio.
- Viewport lengths and widths must always be a nonzero value and both corners of the Viewport must lie on or in the current Page.

Errors

Error 4A
(Viewport of 0 length or outside Page boundary) For 0 length Viewports, the upper-right coordinate(s) is made slightly larger than the lower-left coordinate(s). Viewport coordinates outside Page boundaries are set to the closest boundary.

SET WINDOW

Purpose: This command sets the Window.

SERIAL

	$^{A}_{T}$ $^{A}_{D}$ **AW**	llx,urx,lly,ury
Sample:	$^{E}_{C}$ **A AW**	**0,1,0,1**

GPIB

AW	llx,urx,lly,ury

4050 BASIC (Data Byte Form)

	PRINT @pa,32: **"AW"**;	llx;urx;lly;ury
Sample:	**PRINT @1,32: "AW";**	**0;1;0;1**

Parameters

llx
X-coordinate of the lower-left corner of the Window, in World Units

urx
X-coordinate of the upper-right corner of the Window, in World Units

lly
Y-coordinate of the lower-left corner of the Window, in World Units.

ury
Y-coordinate of the upper-right corner of the Window, in World Units.

Comments

An Initial Window is established at power-up, after a DEVICE RESET command (0, 1, or 3), or by making a selection on any of the following lines of the Parameter Entry Card:

- Media Form
- Initial Page Size
- Initial Page Format
- Page Orientation
- Initial Aspect Ratio

An Initial Window is reset to mm after making a Device Units selection of millimeters. (The DEVICE RESET command establishes the default Device Units: ADUs when the Serial interface is active and GDUs when the GPIB interface is active.)

The following restrictions are associated with the SET WINDOW command:

- Changing Window coordinates does not affect the Viewport.
- All Transforms are reset.
- Zero-length and/or mirrored Windows are not allowed

Errors

Error 4B
(Window less than or equal to 0) The upper-right coordinate(s) is set to a value slightly larger than the lower-left coordinate(s).

SELECT PEN

Purpose: The SELECT PEN command selects Pen 1, Pen 2, or the crosshair cursor. Selection of the crosshair cursor disables the active pen while the crosshair cursor is active.

SERIAL

	A_T A_D **BP**	pen-selector
Sample:	E_C **A BP**	**2**

GPIB

	BP	pen-selector

4050 BASIC (Data Byte Form)

	PRINT @ pa,32: **"BP"**;	pen-selector
Sample:	**PRINT @ 1,32: "BP";**	**2**

4050 BASIC (Secondary Address Form)

	PRINT @ pa,8:	pen-selector
Sample:	**PRINT @ 1,8:**	**2**

Parameters

pen-selector

Pen-selector, 0 to 255:

- 0 = crosshair cursor
- 1 = Pen 1 (closest pen to the Plotter front panel)
- 2 = Pen 2
- 3-255 = no change in the current pen selection

Comments

A pen selection of "0" activates the crosshair cursor. (Normally, there can be a half-second time-out from the last Draw to when the crosshair cursor is selected.) A "0" selection temporarily disables the previous pen selection while the crosshair cursor is active. Subsequent Draws reactivate the selected pen and deactivate the crosshair cursor.

Default selection (Pen 1) is activated after any of the following operations:

- Powering up the Plotter.
- Doing a DEVICE RESET command.
- Performing a front panel Initialize function.

SELECT COORDINATE TYPE

Purpose: This command selects either absolute or relative coordinates for Moves and Draws.

SERIAL

	AT AD **BO**	coordinate-type-selector
Sample:	EC **A BO**	**1**

GPIB

BO	coordinate-type-selector

4050 BASIC (Data Byte Form)

	PRINT @ pa,32: **"BO"**;	coordinate-type-selector
Sample:	**PRINT @ 1,32: "BO";**	**1**

Parameters

coordinate-type-selector
Coordinate type selector, 0-1:

- 0 = absolute coordinates
- 1 = relative coordinates

Comments

The Plotter defaults to absolute coordinates under the following conditions:

- Upon Plotter power-up
- After a DEVICE RESET command

NOTE

Only MOVE and DRAW commands can be used with relative coordinates. All other graphic commands require absolute coordinates.

SELECT CLIPPING CONTROL

Purpose: This command selects the clipping boundary for subsequent plotting.

SERIAL

	A_T A_D **AK**	selector
Sample:	E_C **A AK**	**1**

GPIB

AK	selector

4050 BASIC (Data Byte Form)

	PRINT @pa,32: **"AK";**	selector
Sample:	**PRINT @1,32: "AK";**	**1**

Parameters

selector
Clipping selector, 0-1:

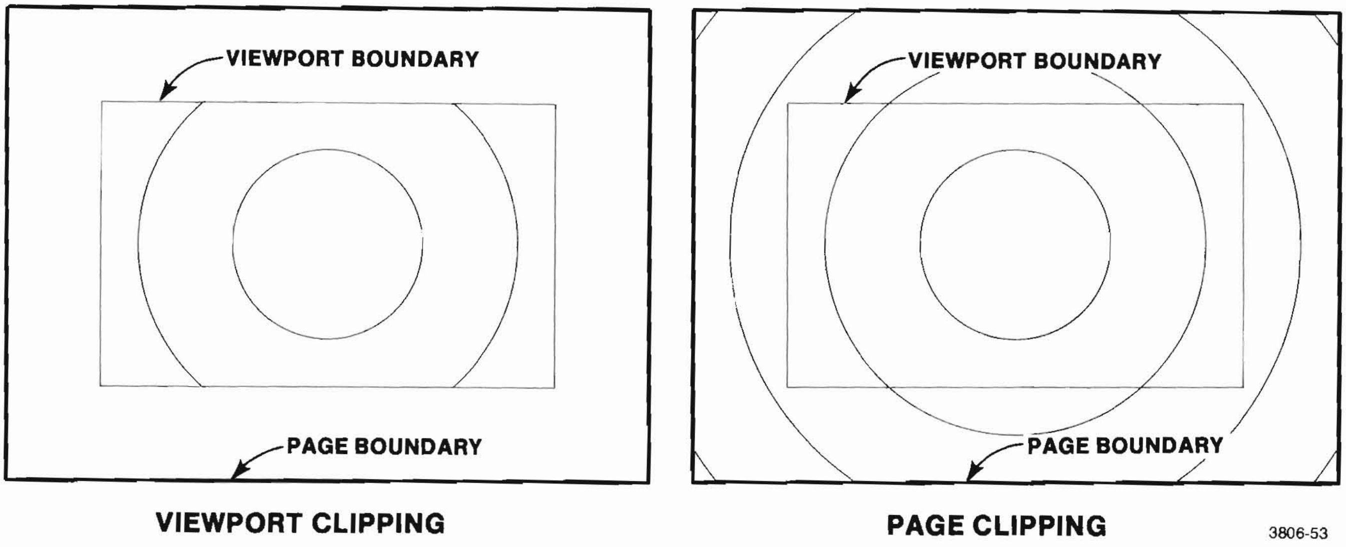
- 0 = Viewport boundary
- 1 = Page boundary

Comments

The default selection is 0. If a portion of the Viewport or Page lies outside the platen clipping boundary, clipping occurs at the platen clipping boundary.

Examples

Figure 11-5 shows clipping on the Viewport and Page boundaries.



VIEWPORT CLIPPING

PAGE CLIPPING

Figure 11-5. Clipping Viewport and Page Boundaries.

SELECT LINE TYPE

Purpose: The SELECT LINE TYPE command selects the type of line (solid, fixed dash-pattern-length, variable dash-pattern-length, or point) to be used when drawing lines, arcs, or circles. Note that the line type selection applies only when World Units is selected. For Device Units, the line type is always solid, regardless of the selection for this command.

SERIAL

	$^{A}{}_{T}$ $^{A}{}_{D}$ **BL**	line-type-selector
Sample:	$^{E}{}_{C}$ **A BL**	**1**

GPIB

BL	line-type-selector

4050 BASIS (Data Byte Form)

	PRINT @pa,32: **"BL"**;	line-type-selector
Sample:	**PRINT @1,32: "BL";**	**1**

Parameters

line-type-selector
Line type selector, 0-3:

- 0 = solid
- 1 = fixed dash-pattern-length
- 2 = variable dash-pattern-length
- 3 = point

Comments

Solid is the default line type selection (which occurs at power-up and after the DEVICE RESET command).

Reference

Refer to *Graphics Concepts*, earlier in this section, for additional information about line types.

SET DASH PATTERN

Purpose: The SET DASH PATTERN command establishes an alternating pattern consisting of up to 20 dots, dashes, and spaces to be used when the fixed or variable dash-pattern-length line types are selected.

SERIAL

	$^{A}{}_{T}$ $^{A}{}_{D}$ **BD**	pattern-element[,pattern-element] . . .
Sample:	$^{E}{}_{C}$ **A BD**	**2,3**

GPIB

	BD	pattern-element[,pattern-element] . . .

4050 BASIC (Data Byte Form)

	PRINT @pa,32: **"BD";**	pattern-element[;pattern-element] . . .
Sample:	**PRINT @1,32: "BD";**	**2;5;0;2;1**

Parameters

pattern-element
Pattern element, integer, sign ignored, 0-255

Comments

The dash pattern consists of 1 to 20 pattern elements ranging in value from 0 to 255.

References

Refer to *Concepts*, earlier in this section, for further discussion on dash patterns.

Errors

Error 44
(Integer Outside Legal Range) Pattern-elements are replaced by 255.

Error 45
(Too Many Entries or String Too Long) This error is reported when more than 20 entries are entered. All extra entries are ignored.

SET DASH PATTERN LENGTH

Purpose: This command sets the dash pattern length for fixed dash-pattern-length line types and sets the maximum dash pattern length for variable dash-pattern-length line types.

SERIAL

	$^{A}_{T}$ $^{A}_{D}$ **BS**	pattern-length
Sample:	$^{E}_{C}$ **A BS**	**100**

GPIB

BS	pattern-length

4050 BASIC (Data Byte Form)

	PRINT @pa,32: "**BS**";	pattern-length
Sample:	**PRINT @1,32: "BS";**	**10**

Parameters

pattern-length
Dash pattern length, sign ignored, in Initial World Units defined along the X-axis

Comments

The default length is 0 at power-up, and after DEVICE RESET commands. Extremely short pattern lengths (including 0) produce solid lines.

References

Refer to *Concepts*, earlier in this section, for further discussion of dash pattern length.

MOVE

Purpose: This command causes the pen carriage to move (with pen up) to the specified position.

SERIAL

	AT AD **X**	x-value,y-value
Sample:	EC **A X**	**1000,2000**

GPIB

	M	x-value,y-value

4050 BASIC (Data Byte Form)

	PRINT@ pa,32: **"M"**;	x-value,y-value
Sample:	**PRINT@ 1,32: "M";**	**50;100**

4050 BASIC (Secondary Address Form)

	PRINT@ pa,21:	x-value;y-value
Sample:	**PRINT@ 1,21:**	**50;100**

4050 BASIC (Special Keyword Form)

	MOVE @pa[,21]:	x-value,y-value
or		
	RMOVE @pa[,21]:	x-relative-value,y-relative-value
Samples:	**MOVE @1:**	**50,100**
or		
	RMOVE @1:	**10,−10**

Parameters

x-value
X-coordinate

y-value
Y-coordinate

x-relative-value
Relative X-coordinate

y-relative-value
Relative Y-coordinate

Comments

The Graphic Units (Device or World Units) selection, the Device Units (ADUs, GDUs, or mm), the coordinate type (absolute or relative), and current Window all affect the MOVE command.

All Moves to points outside the current clipping limits will be clipped and the pen will remain at the clipping limit intercept point. A command to move outside the clipping limits causes the appropriate off-scale status bits in the Device Status word to be set and the lower-left and upper-right front panel lights to turn on.

The 4050 BASIC Special Keyword forms MOVE and RMOVE are special cases. Moves generated by these commands are clipped at the 4050 Window. Coordinates sent by these commands are sent in GDUs and are absolute (even if generated by RMOVE). If the coordinate type is set to relative on the Plotter, then these absolute coordinates are interpreted as relative coordinates; for example, they might be interpreted as a displacement from the current position. So, care should be taken when using these commands.

All non-Keyword forms of the MOVE command have their data transmitted as entered and without transformations or clipping by the 4050 Series controller.

Examples

The Special Keyword Form of MOVE assumes that the 4050 Series Window and Viewport are numerically equal in range. The RMOVE sample causes a displacement of 10,—10 from the current 4050 Series position. The new 4050 Series coordinates are transmitted to the Plotter.

DRAW

Purpose: The DRAW command causes the Plotter to draw with the active pen from the current position to the specified position.

SERIAL

	$^{A}{}_{T}$ $^{A}{}_{D}$ **Y**	x-value,y-value[,x-value,y-value] . . .
Sample:	$^{E}{}_{C}$ **A Y**	**1000,2000**

GPIB

	D	x-value,y-value[,x-value,y-value] . . .

4050 BASIC (Data Byte Form)

	PRINT @pa,32: **"D"**;	x-value;y-value[;x-value;y-value] . . .
Sample:	**PRINT @1,32: "D";**	**50;100**

4050 BASIC (Secondary Address Form)

	PRINT @pa,20:	x-value;y-value[;x-value;y-value] . . .
Sample:	**PRINT @1,20:**	**50;100**

4050 BASIC (Special Keyword Form)

	DRAW @pa[,20]:	x-value,y-value
or		
	RDRAW @pa[,20]:	x-relative-value,y-relative-value
Samples:	**DRAW @1:**	**50,100**
or		
	RDRAW@1:	**10,−10**

Parameters

x-value
X-coordinate

y-value
Y-coordinate

x-relative-value
Relative X-coordinate

y-relative-value
Relative Y-coordinate

Comments

The Graphic Units selection (Device or World Units), the Device Units (ADUs, GDUs, or mms), the coordinate type (absolute or relative), and current Window all affect the DRAW command.

All draws to points outside of the current clipping limits are clipped and the pen remains at the clipping limit intercept point. A command to draw outside the clipping limits causes the appropriate off scale status bits in the Device Status word to be set and the lower left and upper right front panel lights to turn on.

The 4050 BASIC Keyword Forms (DRAW and RDRAW) are special cases. Draws generated by these commands are clipped at the 4050 Window. Coordinates sent by these commands are sent in GDUs and are absolute (even if generated by RDRAW). If the coordinate type is set to relative on the Plotter, these absolute coordinates are interpreted as relative coordinates; for example, they might be interpreted as a displacement from the current position. So, care should be taken when using these commands.

All non-Keyword forms of the DRAW command have their data transmitted as entered without transformations or clipping by the 4050 Series controller.

Examples

The Special Keyword Form of DRAW assumes that the 4050 Series Window and Viewport are numerically equal in range. The RDRAW form causes a displacement of 10,—10 from the current 4050 Series position, and the new 4050 Series coordinates are transmitted to the Plotter.

MOVE TO LOAD POINT

Purpose: This command causes the Plotter's pen carriage to move to the upper-right corner of the platen clipping boundary (the same point is used for media change operations).

SERIAL

$^{A}{}_{T}$ $^{A}{}_{D}$ **AI**

Sample: $^{E}{}_{C}$ **A AI**

GPIB

AI

4050 BASIC (Data Byte Form)

PRINT @pa,32: **"AI"**

Sample: **PRINT@1,32: "AI"**

Comments

The load point usually is outside the Page boundary. If so, the Lower Left and Upper Right lights on the front panel are on, and the appropropiate off-scale bits in the status word are set.

OUTLINE VIEWPORT

Purpose: This command causes the Plotter to draw the Viewport boundary.

SERIAL

$^{A}_{T}$ $^{A}_{D}$ **CB**

Sample: $^{E}_{C}$ **A CB**

GPIB

CB

4050 BASIC (Data Byte Form)

```
         PRINT @ pa,32: "CB"
Sample:  PRINT @ 1,32: "CB"
```

Comments

Upon completion of this function, the pen carriage returns to its original position. This command is identical to the front panel's Outline Viewport function. The outline is clipped for any portion of the Viewport that extends beyond the platen clipping boundary.

MARK VIEWPORT

Purpose: This command causes the Plotter to mark the four corners of the Viewport boundary with small 90° corner brackets.

SERIAL

${}^{A}{}_{T}$ ${}^{A}{}_{D}$ **CM**

Sample: ${}^{E}{}_{C}$ **A CM**

GPIB

CM

4050 BASIC (Data Byte Form)

PRINT@pa,32: **"CM"**

Sample: **PRINT@1,32: "CM"**

Comments

Upon completion of this command, the pen carriage returns to its original position. This command is identical to selecting the front panel Mark Viewport function. Corner marks that extend beyond the platen clipping boundary are clipped.

The actual size of the corner marks depends upon the size of the Viewport or Page, but not on the Window values.

With some Viewport reversal situations, the corner marks extend outward and, therefore, are not visible if the clipping control is set to the Viewport boundary. In this case, dots are plotted at the Viewport corners.

AXIS

Purpose: This command causes the Plotter to draw two lines parallel to the current axes with the specified tic marks on each line.

SERIAL

	$^{A}_{T}$ $^{A}_{D}$ **CA**	x-space,y-space[,x-intercept,y-intercept]
Sample:	$^{E}_{C}$ **A CA**	**100,100,1000,1000**

GPIB

	CA	x-space,y-space[,x-intercept,y-intercept]

4050 BASIC (Data Byte Form)

	PRINT @ pa,32: **"CA";**	x-space;y-space[;x-intercept;y-intercept]
Sample:	**PRINT @ 1,32: "CA";**	**10;10**

Parameters

x-space
Spacing between tic marks on the X-axis, non-negative, 0 means no tic marks

y-space
Spacing between tic marks on the Y-axis, non-negative, 0 means no tic marks

x-intercept
X-coordinate of axis intersection

y-intercept
Y-coordinate of axis intersection

Comments

The parameters are specified in World Units. The default intercept values are 0 if x-intercept and y-intercept are omitted. The actual size of the tic marks depends upon the size of the Viewport or Page and does not vary with Window values.

Errors

Error 44
(Integer Outside Legal Range) This error occurs if negative values are entered for x-space or y-space.

Examples

A command showing an axis at 0,0 with no tic marks on the Y-axis and spacings of 10 on the X-axis is:

PRINT @ 1,32: "CA"; 10;0

An axis with the intercept at 4000,3000 and no tic marks is:

$^{E}_{C}$ **A CA 0,0,4000,3000**

READ VIEWPORT

Purpose: This command causes the Plotter to report the locations of the lower-left and upper-right corners, respectively, of the Viewport. Coordinates are specified in Device Units.

SERIAL

(Command/Response Formats 1-2 Only)
(Two Output Message Blocks Produced)

AT AD **CV**

Sample: EC **A CV**

GPIB

(Two Output Message Blocks Produced)

CV

4050 BASIC (Data Byte Form)

```
         PRINT@pa,32: "CV"
         INPUT@pa,32:          first-value-1-target-variable,
                               first-value-2-target-variable,
                               first-tag-value-target-variable
         INPUT@pa,32:          last-value-1-target-variable,
                               last-value-2-target-variable,
                               last-tag-value-target-variable
Sample:  PRINT@1,32: "CV"
         INPUT@1,32:           V1,V2,T1
         INPUT@1,32:           V3,V4,T2
```

Outputs

First output message block:
value 1
X-coordinate of lower-left corner
value 2
X-coordinate of upper-right corner
tag value
See Table 11-2

Last output message block:
value 1
Y-coordinate of lower-left corner
value 2
Y-coordinate of upper-right corner
tag value
See Table 11-2

Table 11-2

TAG VALUES FOR READ VIEWPORT COMMAND

TAG Meaning	TAG Value By Format		
	Serial Formats		GPIB Formats
	1	2	1 Thru 6
First Block	2	0	0
Last Block	4	2	2

References

See the Serial Interface and GPIB Interface sections (Sections 4 and 5) for information about decoding output responses from the Plotter.

SET ARC SMOOTHNESS

Purpose: This command sets the smoothness (or granularity) of arcs and circles, including those used for enhanced character quality and Print actions.

SERIAL

	$^{A}_{T}$ $^{A}_{D}$ **BA**	smoothness
Sample:	$^{E}_{C}$ **A BA**	**.2**

GPIB

BA	smoothness

4050 BASIC (Data Byte Form)

	PRINT@ pa,32: **"BA"**;	smoothness
Sample:	**PRINT**@ **1,32: "BA";**	**.8**

Parameters

smoothness
smoothness value, 0-1, sign ignored

Comments

The smoothness value determines how many chords are used to approximate an arc or a circle.

The smoothness value is interpreted as follows:

0	The smoothest setting, which causes the Plotter to draw arcs and circles using the maximum number or chords.
0.5	A setting used for default setups.
1	A very coarse setting; usually the coarsest available setting.
>1	The same as a setting of "1" unless the limiting case in coarseness has not been reached.

The defaults setting of 0.5 occurs after power-up and with the DEVICE RESET commands.

References

Refer to *Arcs and Circles*, earlier in this section.

DRAW ARC

Purpose: This command causes the Plotter to draw an arc, which is defined by three points: the current pen carriage position and two specified points.

SERIAL

	$^{A}{}_{T}$ $^{A}{}_{D}$ **AA**	x-mid,y-mid,x-end,y-end
Sample:	$^{E}{}_{C}$ **A AA**	**500,500,1000,0**

GPIB

AA	x-mid,y-mid,x-end,y-end

4050 BASIC (Data Byte Form)

	PRINT @pa,32: **"AA"**;	x-mid;y-mid;x-end;y-end
Sample:	**PRINT** @**1,32:** **"AA"**;	**50;50;0;0**

Parameters

x-mid
X-coordinate of intermediate point, in World Units

y-mid
Y-coordinate of intermediate point, in World Units

x-end
X-coordinate of end point, in World Units

y-end
Y-coordinate of end point, in World Units

Comments

This command requires Option 31. The current pen carriage position provides the first X and Y coordinates (the starting point of the arc). The Graphic Units selection must be World Units.

The coarseness of the arc is dependent upon the SET ARC SMOOTHNESS command. For maximum precision, the intermediate point should be near the midpoint of the arc.

If the end-point is coincident with the first point (current pen carriage position), a circle with its diameter defined by the first and intermediate point is drawn.

If three points are in a line, or if the midpoint coincides with either of the end points, the DRAW ARC command draws a straight line from the first point to the end point. In some cases, due to numerical resolution, an arc with a very large radius is drawn between the points specified.

References

Refer to the SET ARC SMOOTHNESS command to change coarseness of the plotted arcs.

Examples

In the Serial sample, if the current pen carriage position is at 0,0, then an arc is drawn from that point clockwise through 500,500 and ending at 1000,0.

In the 4050 BASIC sample, if the current position is at 0,0, then a circle is drawn such that 0,0 and 50,50 lie on its diameter. If the current position is 0,100, then a straight line between 0,100 and 100,0 is drawn. If the current position is 100,100, then a counterclockwise arc is drawn.

DRAW CIRCLE

Purpose: This command causes the Plotter to draw a circle centered at the current pen position, and with the specified radius.

SERIAL

	^{A}T ^{A}D **AC**	radius[,radius] . . .
Sample:	^{E}C **A AC**	**1000**

GPIB

AC	radius[,radius] . . .

4050 BASIC (Data Byte Form)

	PRINT @pa,32: **"AC";**	radius[;radius] . . .
Sample:	**PRINT @1,32: "AC";**	**30**

Parameters

radius
Radius of circle, in World Units

Comments

This command requires Option 31. The Graphic Units selection must be World Units.

Both the DRAW CIRCLE command and the DRAW ARC command can draw circles. However, the DRAW CIRCLE command has the following advantages:

- It requires only one argument (radius).
- The pen carriage returns to the center of the circle after the circle is drawn (rather than remaining at the end-point).
- This command accepts multiple arguments, which thereby specify concentric circles.

Examples

Two "target" patterns are generated below:

^{E}C **A AC 100,200,300,400,500,600,700, 800,900,1000**

PRINT @1,32: "AC"; 10;20;30;40;50;60; 70;80;90;100

Section 12

TRANSFORM COMMANDS

CONCEPTS

Transform routines (actually, matrix operations) cause the Plotter to modify incoming data. Six transform commands are provided to perform specific transform operations (e.g., scaling). Transforms are combined with existing Viewport and Window settings and other transforms to produce a transform matrix designated as the current transform. (Page parameters are handled separately, so that front panel Set Page adjustments can be made without affecting the Viewport, Window, and transforms.) The current transform is used to convert incoming data into Plotter coordinates, and the inverse of this current transform is used for digitizing operations.

The current transform can be modified by subsequent transform commands to form a new current transform. At any stage, the current transform can be saved (on a transform stack) so that it can be restored for later use. Two commands to do this are discussed later.

Transform commands are applicable only when the Graphic Units selection is World Units. When using Device Units, all transforms and Viewport and Window settings are bypassed.

TRANSFORMATION OF DATA

The following transform commands cause incoming data to be modified by the specified operation. All of these commands assume that the Viewport and Window have been established.

SET SCALE Command

Incoming X- and Y-coordinates are independently multiplied by the dimensionless X- and Y-scale factors. Negative scale factors are allowed.

SET TRANSLATION Command

X- and Y-coordinate translation values are added to the incoming X- and Y-coordinates. This operation has the effect of shifting the location of the origin (0,0 location) by the translation values.

SET ROTATION Command

A rotation operation (multiplication by a rotation matrix) is performed on the incoming X- and Y-coordinates so that the data appears rotated by the specified angle. The positive direction of rotation is counterclockwise. Rotations occur about the data origin.

SET SKEW Command

The coordinate system can be skewed in the manner shown in Figure 12-1. The arrows indicate positive skew angles. While the general case allows X- and Y-skew angles to be entered together, most applications (e.g., to slant a picture) involve entering only one angle at a time. The SET SKEW command is defined for angles given in all four quadrants such that the picture will be reflected about the appropriate axis for angles in the 90° to 270° range.

SET ORIGIN TO CURRENT POSITION Command

The origin is set at the current pen carriage position by the process of identifying the current position and then doing a translation.

SET ROTATION TO LAST ANGLE Command

In this operation, a Set Origin To Current Position operation is performed followed by a rotation operation so that the new X-axis is in the same direction as the last MOVE or DRAW command (or, with Option 31, the last chord of the DRAW ARC command). The origin is at the current pen carriage position. This routine is useful for printing alphanumeric characters or other symbols along the same axis of a previously drawn line.

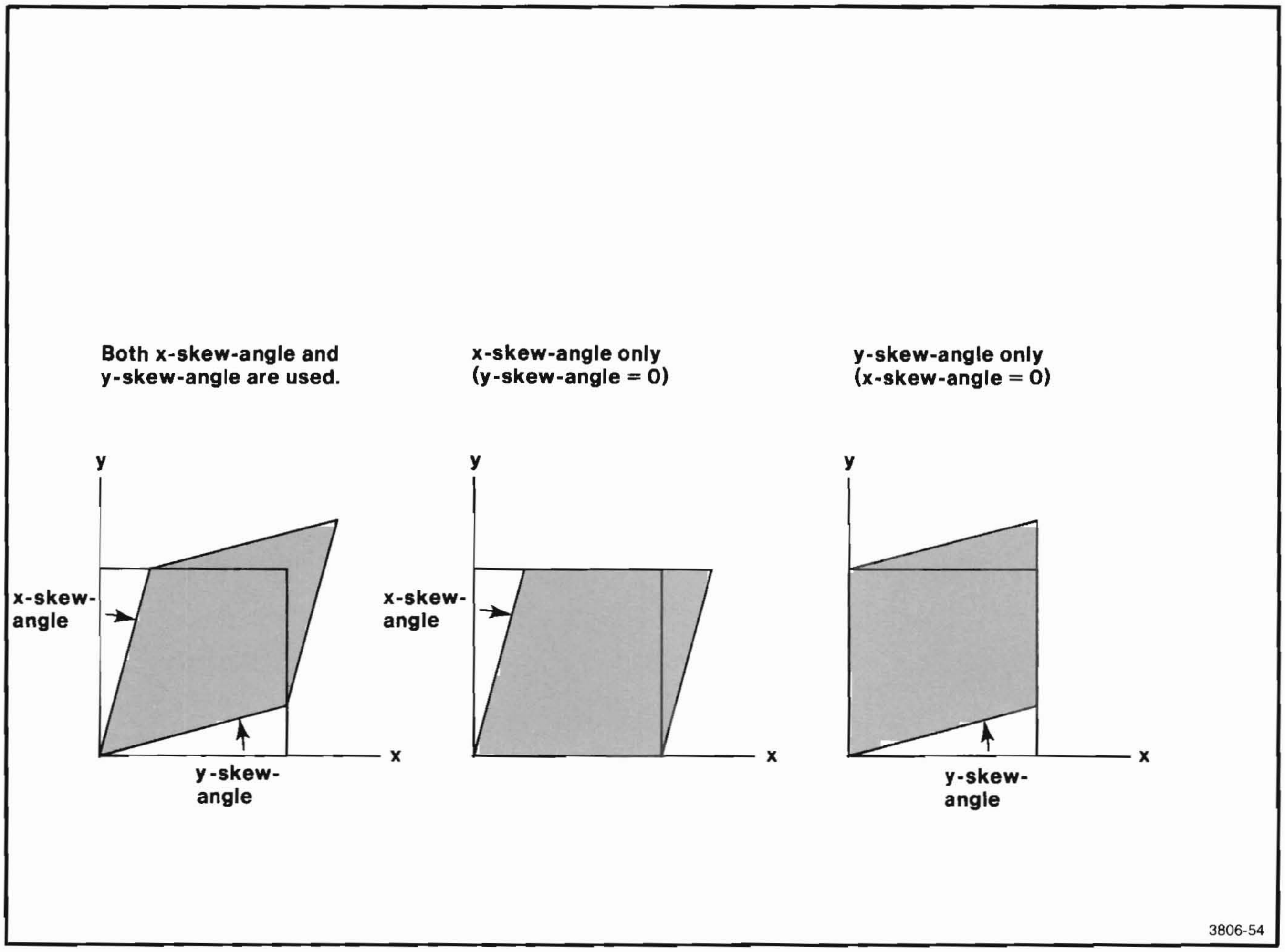

Both x-skew-angle and y-skew-angle are used.
x-skew-angle only (y-skew-angle = 0)
y-skew-angle only (x-skew-angle = 0)
y
x
x-skew-angle
y-skew-angle
3806-54

Figure 12-1. Conventions for the SET SKEW Command.

COMBINING TRANSFORMS

The order in which the transforms are entered is important. The convention adapted in the Plotter is one of reverse ordering — the last transform received is the first one to act on the incoming data. Figures 12-2 and 12-3 illustrate how the order that transforms are entered affect the picture. The steps in Figure 12-2 show how a square is transformed by commands entered in this order:

1. SET ROTATION (angle = 45°)
2. SET SCALE (x-scale = 2, y-scale = 1)

Figure 12-3 illustrates the results of entering the same commands, but in the opposite order:

1. SET SCALE (x-scale = 2, y-scale = 1)
2. SET ROTATION (angle = 45°)

Note that the Plotter applies commands in the opposite order in which you enter them.

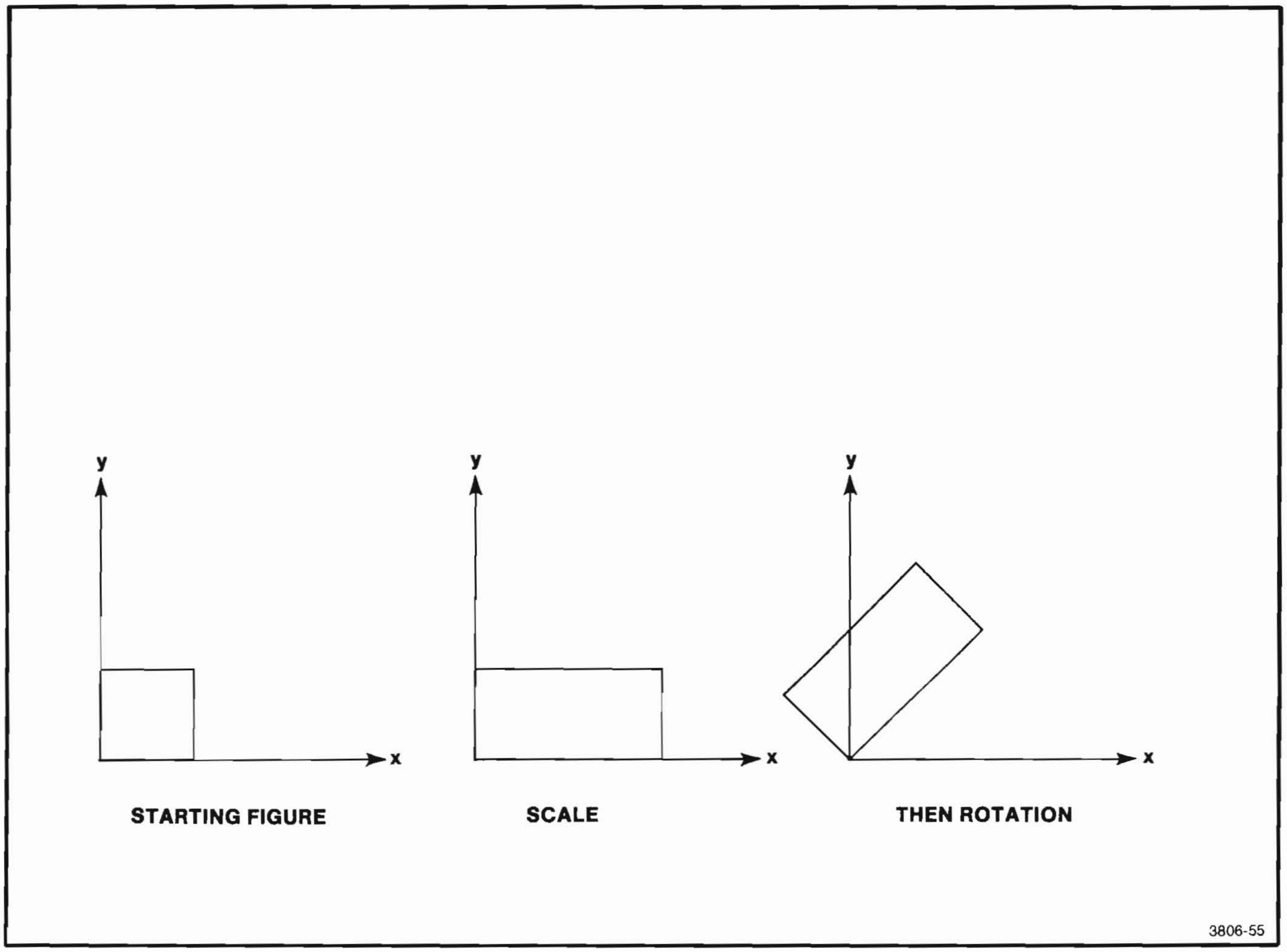


Figure 12-2. Scale and Rotation.

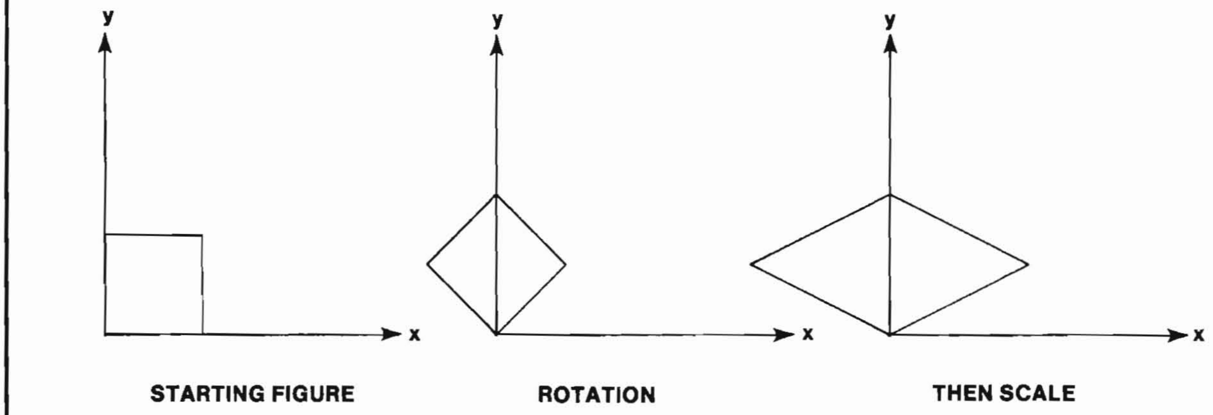


Figure 12-3. Rotation and Scale.

SAVING AND RESTORING TRANSFORMS

The SAVE CURRENT TRANSFORM and RESTORE PREVIOUS TRANSFORM commands allow current transforms to be saved on a transform stack and recalled later. The stack is cleared along with any transform associated with the current transforms when a new SET WINDOW or SET VIEWPORT operation is performed. These operations can be either commands, front panel operations, or a result of Paramater Entry Card operations. Also, the DEVICE RESET command clears all transforms associated with the current transform and clears the transform stack.

The SAVE CURRENT TRANSFORM command is used to save, on the transform stack, the current transform and any previously saved transforms.

The RESTORE PREVIOUS TRANSFORM command is used to remove the last transform saved on the transform stack and to restore it as the current transform. The next previously saved transform is available to be restored at the next RESTORE PREVIOUS TRANSFORM operation. So the stack operates on a last-in, first-out basis.

One application of the save and restore operations involves labeling vectors along the vector axis. The procedure involves these steps:

1. Draw the vector
2. Issue the SAVE CURRENT TRANSFORM command
3. Issue the SET ROTATION TO LAST ANGLE command
4. Label the vector
5. Issue the RESTORE PREVIOUS TRANSFORM command
6. Continue plotting

COMMAND DESCRIPTIONS

There are eight commands described in this section:

- SET SCALE
- SET TRANSLATION
- SET SKEW
- SET ROTATION
- SET ORIGIN TO CURRENT POSITION
- SET ROTATION TO LAST ANGLE
- SAVE CURRENT TRANSFORM
- RESTORE PREVIOUS TRANSFORM

Each command is described separately in the pages that follow.

SET SCALE

Purpose: This command multiplies the plot data by the specified scale factors.

SERIAL

	$^{A}{}_{T}$ $^{A}{}_{D}$ **AS**	x-scale,y-scale
Sample:	$^{E}{}_{C}$ **A AS**	**−1,2.5**

GPIB

	AS	x-scale,y-scale

4050 BASIC (Data Byte Form)

	PRINT @pa,32: **"AS"**;	x-scale;y-scale
Sample:	**PRINT@1,32: "AS";**	**−1;2.5**

Parameters

x-scale
X-coordinate scale factor (nonzero)

y-scale
Y-coordinate scale factor (nonzero)

Comments

Negative scale factors are allowed.

Errors

Error 49
(Scale = 0) The Plotter sets a very small positive scale factor.

SET TRANSLATION

Purpose: This command translates plot data by specified translation values.

SERIAL

	$^{A}{}_{T}$ $^{A}{}_{D}$ **AT**	x-translation,y-translation
Sample:	$^{E}{}_{C}$ **A AT**	**20,0**

GPIB

	AT	x-translation,y-translation

4050 BASIC (Data Byte Form)

	PRINT @pa,32: **"AT"**;	x-translation;y-translation
Sample:	**PRINT @1,32: "AT";**	**20;0**

Parameters

x-translation
Translation distance in X direction

y-translation
Translation distance in Y direction

SET SKEW

Purpose: This command skews plot data for either or both axes.

SERIAL

	^{A}T ^{A}D **AQ**	x-skew-angle,y-skew-angle
Sample:	^{E}C **A AQ**	**15,15**

GPIB

	AQ	x-skew-angle,y-skew-angle

4050 BASIC (Data Byte Form)

	PRINT @pa,32: **"AQ"**;	x-skew-angle;y-skew-angle
Sample:	**PRINT @1,32: "AQ";**	**15;15**

Parameters

x-skew-angle
The clockwise angle between the old and new Y-axes, in degrees

y-skew-angle
The counterclockwise angle between the old and new X-axes, in degrees

Comments

This command can change the plotted lengths of units on the skewed axes.

Angles from 89.9° to 90.1° and from 269.9° to 270.1° are converted to 89.9°.

This command is defined for angles given in all four quadrants such that the picture will be reflected about the appropriate axis for angles in the 90° to 270° range.

Reference

Angle direction conventions for the SET SKEW command are described under *Concepts* at the beginning of this section.

Errors

Combinations of x-skew-angle and y-skew-angle that add up to 90° or 270° are not allowed. Such combinations may produce errors in subsequent operations as a result of transforming the incoming data onto a line.

SET ROTATION

Purpose: This command rotates plot data about the origin.

SERIAL

	$^{A}_{T}$ $^{A}_{D}$ **AR**	angle
Sample:	$^{E}_{C}$ **A AR**	**30**

GPIB

AR	angle

4050 BASIC (Data Byte Form)

	PRINT @pa,32: **"AR"**	angle
Sample:	**PRINT @1,32: "AR"**	**30**

Parameters

angle
Rotation angle, in a counterclockwise direction, in degrees

SET ORIGIN TO CURRENT POSITION

Purpose: This command sets the origin (0,0) to the current pen carriage position.

SERIAL

$^{A}{}_{T}$ $^{A}{}_{D}$ **AO**

Sample: $^{E}{}_{C}$ **A AO**

GPIB

AO

4050 BASIC (Data Byte Form)

PRINT @pa,32: **"AO"**

Sample: **PRINT** @ **1,32: "AO"**

SET ROTATION TO LAST ANGLE

Purpose: This command sets the origin (0,0) to the current pen carriage position and then rotates the new system about its origin so that the new X-axis is in the direction of the most recent motion (MOVE or DRAW) command (or, with Option 31, the last chord of the DRAW ARC command).

SERIAL

```
         ^T ^D AL
Sample:  ^E_C A AL
```

GPIB

```
         AL
```

4050 BASIC (Data Byte Form)

```
         PRINT @pa,32: "AL"
Sample:  PRINT @1,32: "AL"
```

SAVE CURRENT TRANSFORM

Purpose: This command saves the current transform so that it can be restored for later use.

SERIAL

```
         A_T A_D AX
Sample:  E_C A AX
```

GPIB

```
         AX
```

4050 BASIC (Data Byte Form)

```
         PRINT @pa,32: "AX"
Sample:  PRINT @1,32: "AX"
```

Comments

This command saves the current transform by pushing it onto a last-in, first-out transform stack for retrieval by the RESTORE PREVIOUS TRANSFORM command.

Each transform saved requires 30 bytes of RAM storage. The current transform remains in effect.

References

See the RESTORE PREVIOUS TRANSFORM command.

Errors

Error 82
(Transform stack overflow or underflow) Memory is insufficient to store the current transform.

RESTORE PREVIOUS TRANSFORM

Purpose: This command restores a transform previously saved by the SAVE CURRENT TRANSFORM command.

SERIAL

A_T A_D **AY**

Sample: E_C **A AY**

GPIB

AY

4050 BASIC (Data Byte Form)

PRINT @pa,32: **"AY"**

Sample: **PRINT @1,32: "AY"**

Comments

This command replaces the current active transform with the transform most recently saved by the SAVE CURRENT TRANSFORM command.

Recalling the saved transform frees the Plotter memory used by the previous SAVE CURRENT TRANSFORM command.

References

See SAVE CURRENT TRANSFORM command.

Errors

Error 82

(Transform stack underflow or overflow) No transforms are currently stored in the transform stack, and the current transform is not changed.

Section 13

DIGITIZING COMMANDS

INTRODUCTION

The Plotter's digitizing commands permit the host computer to request and obtain coordinates of: (1) the current pen position, or, (2) of points designated by the operator by front panel POINT switch functions. Current pen position information is obtained using the DIGITIZE command.

Operator designated points are obtained using the Operator Digitize Set-up and Action commands discussed below for each of the two interfaces. Note that POINT switch responses depend on the POINT switch mode, which may be disabled, enabled, or permanently enabled.

CONCEPTS

Five operator digitizing procedures are discussed in the following text. Two procedures are for use with the Serial interface; three are for use with a GPIB interface. The procedures are:

- Operator Digitizing via Serial Interface (Formats 1 through 4)
- Operator Digitizing via Serial Interface (Formats 3 and 4)
- Operator Digitizing via GPIB Interface (Formats 1 through 6)
- Operator Digitizing via GPIB Interface (Formats 5 and 6)
- Operator Digitizing via GPIB Interface (without SRQ Messages)

OPERATOR DIGITIZING VIA SERIAL INTERFACE (FORMATS 1-4)

Operator digitizing consists of: (1) enabling or disabling the front panel POINT switch, (2) controlling the front panel Point light, and (3) processing the data received from the Plotter for each point.

The following summarizes a typical application:

1. The OPERATOR DIGITIZE ENABLE command is used to enable the Plotter's POINT switch for (in this example) twelve points and to turn the Point light to a steady on condition.
2. For each point, the operator uses the Joystick to position the crosshair cursor over a desired point and then selects either the front panel Move Point or Draw Point function. This instructs the Plotter to store the point.

   Alternatively, the operator can use the front panel Last Point function to indicate the last point (see Step 3).
3. For each point, the Plotter sends one output message block of information consisting of X and Y position values and a TAG value. The TAG value indicates the POINT switch function (Move Point or Draw Point) and, for the last point, it indicates the last block of data. (If Format 3 or 4 is used, the Plotter sends a status block, rather than position information for the last point.)

   If this is NOT the last point, go back to Step 2.

4. The Plotter turns off the steady POINT light indication, and disables the POINT switch.

This completes the operator digitizing process. Either the Plotter operator or the host computer can terminate the operator digitizing process after fewer than the specified number of points; the operator can use the POINT switch Last Point function or the host computer can send an OPERATOR DIGITIZE ENABLE command with an argument of 0 to disable the POINT switch. The Plotter responds in Formats 1 and 2 by sending a final one-block output message with a TAG value indicating the last block and in Formats 3 and 4 by sending a status block.

For Formats 3 and 4 only, the host computer must enable the POINT switch for one more point than required. The "last point" response contains status rather than position information.

For special applications, the host computer can also use the JOYSTICK AXIS DISABLE command to restrict Joystick motion to one axis only, or the PROMPT LIGHT ON and PROMPT LIGHT OFF commands to signal the operator to take a predefined action.

OPERATOR DIGITIZING VIA SERIAL INTERFACE (FORMATS 3 AND 4 ONLY)

Operator digitizing, when using Command/Response Formats 3 and 4, can be performed as described above for Formats 1 through 4 or as described below for the permanently enabled POINT switch mode.

For Command/Response Formats 3 and 4, the POINT switch is permanently enabled after power-up or after a DEVICE RESET, INTERFACE PARAMETER RESET, or SELECT COMMAND/RESPONSE FORMAT command (and remains so unless an OPERATOR DIGITIZE ENABLE command is received). Thus, the Plotter operator must be instructed as to when to use or not use the POINT switch.

A typical application is summarized below:

1. The PROMPT LIGHT ON command is used to flash the Point light, which indicates to the Plotter operator that digitizing is permitted.

2. For each point, the operator uses the Joystick to position the crosshair cursor over a desired point and then selects either the front panel's Move Point or Draw Point function, thus instructing the Plotter to store the point.

   Alternatively, the Plotter operator can use the Last Point function (which sends a status message) to indicate the end of digitizing. This action assumes an agreed-upon meaning for the status message. The Last Point function does NOT cancel the permanently enabled mode, and does not acquire point position data.

3. For each point, the Plotter sends one output message block of information consisting of X and Y position values and a TAG value. The TAG value indicates the POINT switch function (Move Point or Draw Point). If a Last Point function is used, the Plotter sends a status block rather than position information.

   If this is NOT the last point, go back to Step 2.

4. PROMPT LIGHT OFF command is used to stop the flashing Point Light indication, thus informing the Plotter operator to stop digitizing (the Point light remains lit and the POINT switch is still permanently enabled).

This completes the operator digitizing process.

OPERATOR DIGITIZING VIA GPIB INTERFACE (FORMATS 1-6)

Operator digitizing involves: (1) enabling or disabling the front panel POINT switch (and sometimes the Joystick), (2) controlling the front panel Point light, (3) requesting data for each point (via the CALL DIGITIZE command), and (4) processing the data received from the Plotter for each point.

The operator digitizing operations discussed below involve GPIB SRQ messages used to inform the GPIB controller that the Plotter has data available. An alternate method of operation without SRQ message generation is discussed later in this section.

A typical application for operator digitizing is summarized below:

1. The OPERATOR DIGITIZE ENABLE command is used to enable the POINT switch for (for example) twelve points and to turn the Point light to a steady on condition.
2. For each point, the operator uses the Joystick to position the crosshair cursor over a desired point and then selects either the front panel Move Point or Draw Point function, thus instructing the Plotter to store the point.

   Alternatively, the Plotter operator can use the front panel Last Point function to store a point and to terminate operator digitizing. (Refer to the discussion below.)
3. The Plotter sends an SRQ (Status Request) message.
4. For each point, the GPIB controller sends a CALL DIGITIZE command to request data.
5. For each point, the Plotter sends one output message block of information consisting of X and Y position values and a TAG value. The TAG value indicates the POINT switch function (Move Point or Draw Point) and indicates for the last point, the last block of data.

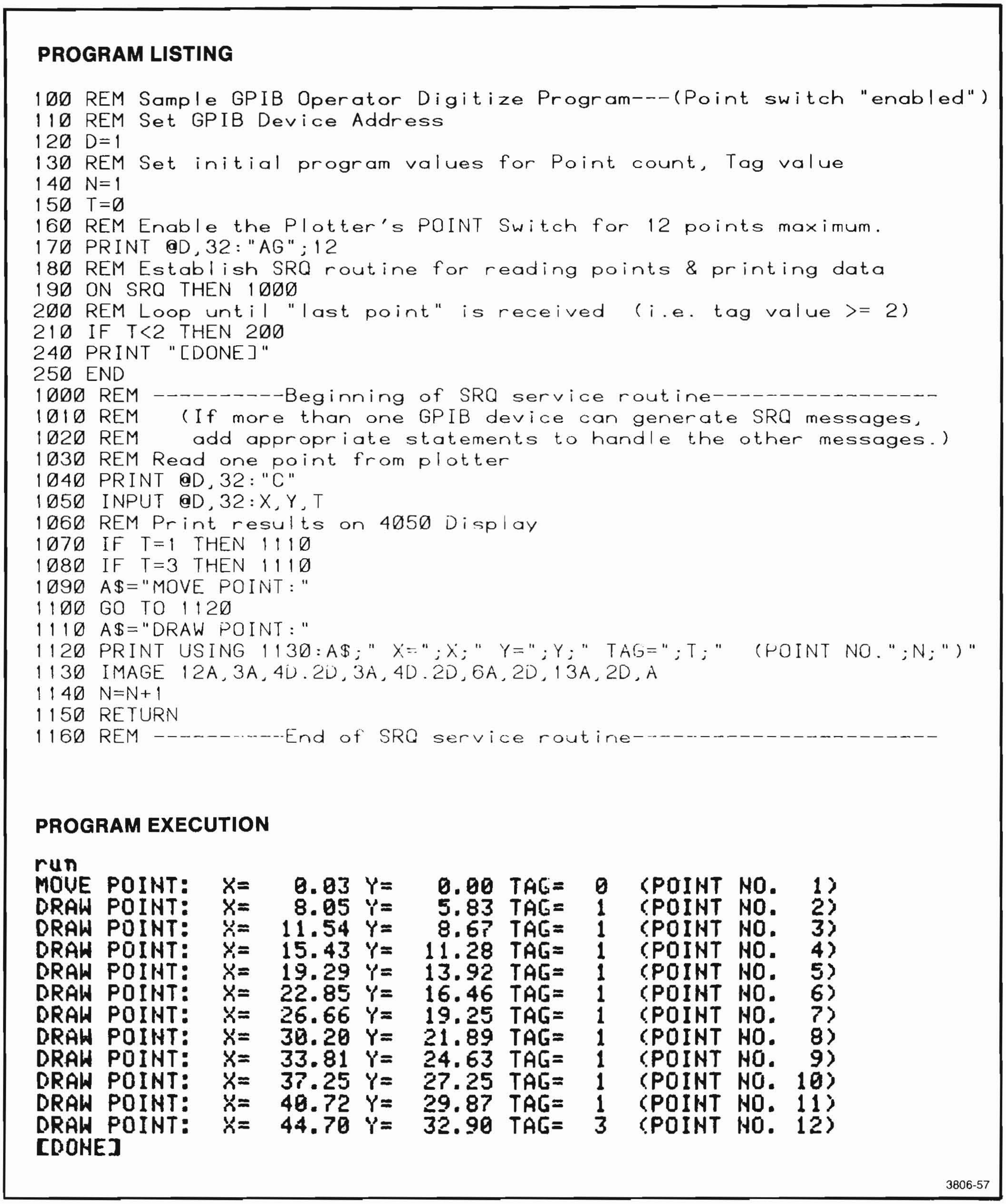
   If this is NOT the last point, go back to Step 2.
6. The Plotter turns off the Point light, and disables the POINT switch functions.

This completes the operator digitizing process. The sample 4050 program in Figure 13-1 allows the Plotter operator to store up to twelve points and displays the results on the 4050 display.

Either the Plotter operator or the GPIB controller can terminate the operator digitizing process after fewer than the specified number of points. The operator can use the POINT switch Last Point function, or the controller can send an OPERATOR DIGITIZE ENABLE command with an argument of 0 (meaning disable). The Plotter responds by sending a one block output message with a TAG indicating last block. (If the GPIB controller disables the enabled POINT switch, it must be able to handle the corresponding one-block output and SRQ messages.)

In general, if (1) either method of disabling the POINT switch after fewer than the specified number of points is allowed, and (2) Move vs. Draw Point TAG data is needed, then the GPIB controller should enable the POINT switch for one more than the required number of points, and treat the last block of data as control information, not point data. (This avoids the limitation that the TAG value for a front panel Last Point function cannot include Draw Point (vs. Move Point) information; the TAG value always indicates Move Point.)

For special applications, the host computer can use the JOYSTICK DISABLE command to restrict Joystick motion to one axis only, or the PROMPT LIGHT ON and PROMPT LIGHT OFF commands to signal the operator to take a predefined action.



PROGRAM LISTING

```
1ØØ REM Sample GPIB Operator Digitize Program---(Point switch "enabled")
11Ø REM Set GPIB Device Address
12Ø D=1
13Ø REM Set initial program values for Point count, Tag value
14Ø N=1
15Ø T=Ø
16Ø REM Enable the Plotter's POINT Switch for 12 points maximum.
17Ø PRINT @D,32:"AG";12
18Ø REM Establish SRQ routine for reading points & printing data
19Ø ON SRQ THEN 1ØØØ
2ØØ REM Loop until "last point" is received  (i.e. tag value >= 2)
21Ø IF T<2 THEN 2ØØ
24Ø PRINT "[DONE]"
25Ø END
1ØØØ REM ----------Beginning of SRQ service routine-----------------
1Ø1Ø REM   (If more than one GPIB device can generate SRQ messages,
1Ø2Ø REM    add appropriate statements to handle the other messages.)
1Ø3Ø REM Read one point from plotter
1Ø4Ø PRINT @D,32:"C"
1Ø5Ø INPUT @D,32:X,Y,T
1Ø6Ø REM Print results on 4Ø5Ø Display
1Ø7Ø IF T=1 THEN 111Ø
1Ø8Ø IF T=3 THEN 111Ø
1Ø9Ø A$="MOVE POINT:"
11ØØ GO TO 112Ø
111Ø A$="DRAW POINT:"
112Ø PRINT USING 113Ø:A$;" X=";X;" Y=";Y;" TAG=";T;"  (POINT NO.";N;")"
113Ø IMAGE 12A,3A,4D.2D,3A,4D.2D,6A,2D,13A,2D,A
114Ø N=N+1
115Ø RETURN
116Ø REM ----------End of SRQ service routine-----------------------
```

PROGRAM EXECUTION

```
run
MOVE POINT:   X=   0.03 Y=   0.00 TAG=  0  (POINT NO.  1)
DRAW POINT:   X=   8.05 Y=   5.83 TAG=  1  (POINT NO.  2)
DRAW POINT:   X=  11.54 Y=   8.67 TAG=  1  (POINT NO.  3)
DRAW POINT:   X=  15.43 Y=  11.28 TAG=  1  (POINT NO.  4)
DRAW POINT:   X=  19.29 Y=  13.92 TAG=  1  (POINT NO.  5)
DRAW POINT:   X=  22.85 Y=  16.46 TAG=  1  (POINT NO.  6)
DRAW POINT:   X=  26.66 Y=  19.25 TAG=  1  (POINT NO.  7)
DRAW POINT:   X=  30.20 Y=  21.89 TAG=  1  (POINT NO.  8)
DRAW POINT:   X=  33.81 Y=  24.63 TAG=  1  (POINT NO.  9)
DRAW POINT:   X=  37.25 Y=  27.25 TAG=  1  (POINT NO. 10)
DRAW POINT:   X=  40.72 Y=  29.87 TAG=  1  (POINT NO. 11)
DRAW POINT:   X=  44.70 Y=  32.90 TAG=  3  (POINT NO. 12)
[DONE]
```

3806-57

Figure 13-1. Sample GPIB Operator Digitize Program (Enabled).

OPERATOR DIGITIZING VIA GPIB INTERFACE (FORMATS 5 AND 6)

Operator digitizing, when using Command/Response Formats 5 and 6, can be performed as previously described for Formats 1-6 or as described in the following text for the permanently enabled mode.

The operator digitizing operations discussed here involve GPIB SRQ messages used to inform the GPIB Controller that the Plotter has data available. An alternate operation without SRQ (Status Request) message generation is discussed later in this section.

For Command/Response Formats 5 and 6, the POINT switch is permanently enabled after power-up or after a DEVICE RESET or SET COMMAND/RESPONSE FORMAT command or after a GPIB INTERFACE CLEAR message (and remains so unless an OPERATOR DIGITIZE ENABLE command is received). Thus, the Plotter operator must be instructed as to when to use or not use the POINT switch.

A typical application is summarized below:

1. The PROMPT LIGHT ON command is used to flash the Point light, indicating to the Plotter operator that digitizing is permitted.
2. For each point, the operator uses the Joystick to position the crosshair cursor over a desired point and then selects either the front panel's Move Point or Draw Point function, thus instructing the Plotter to store the point.

   Alternatively, the Plotter operator can use the Last Point function for instructing the Plotter to store a point (which produces a last block TAG value) and to indicate the end of digitizing. This action assumes an agreed-upon meaning for the last block TAG information. The Last Point function does NOT disable the permanently enabled POINT switch.
3. The Plotter then sends a SRQ message.
4. For each point, the GPIB controller sends a CALL DIGITIZE command to request data.
5. For each point, the Plotter sends one output block of information consisting of X and Y position values and a TAG value. The TAG value indicates the POINT switch function (Move Point, Draw Point, or Last Point).

   If this is NOT the last point, go back to Step 2.
6. The PROMPT LIGHT OFF command stops the flashing POINT light indication, thus informing the Plotter operator to stop digitizing. (The Point light remains on steady; the POINT switch is still permanently enabled.)

The sample 4050 program in Figure 13-2 allows the Plotter operator to store points and displays the results on the 4050 display.

This completes the operator digitizing process.

OPERATOR DIGITIZING VIA GPIB INTERFACE WITHOUT SRQ MESSAGES

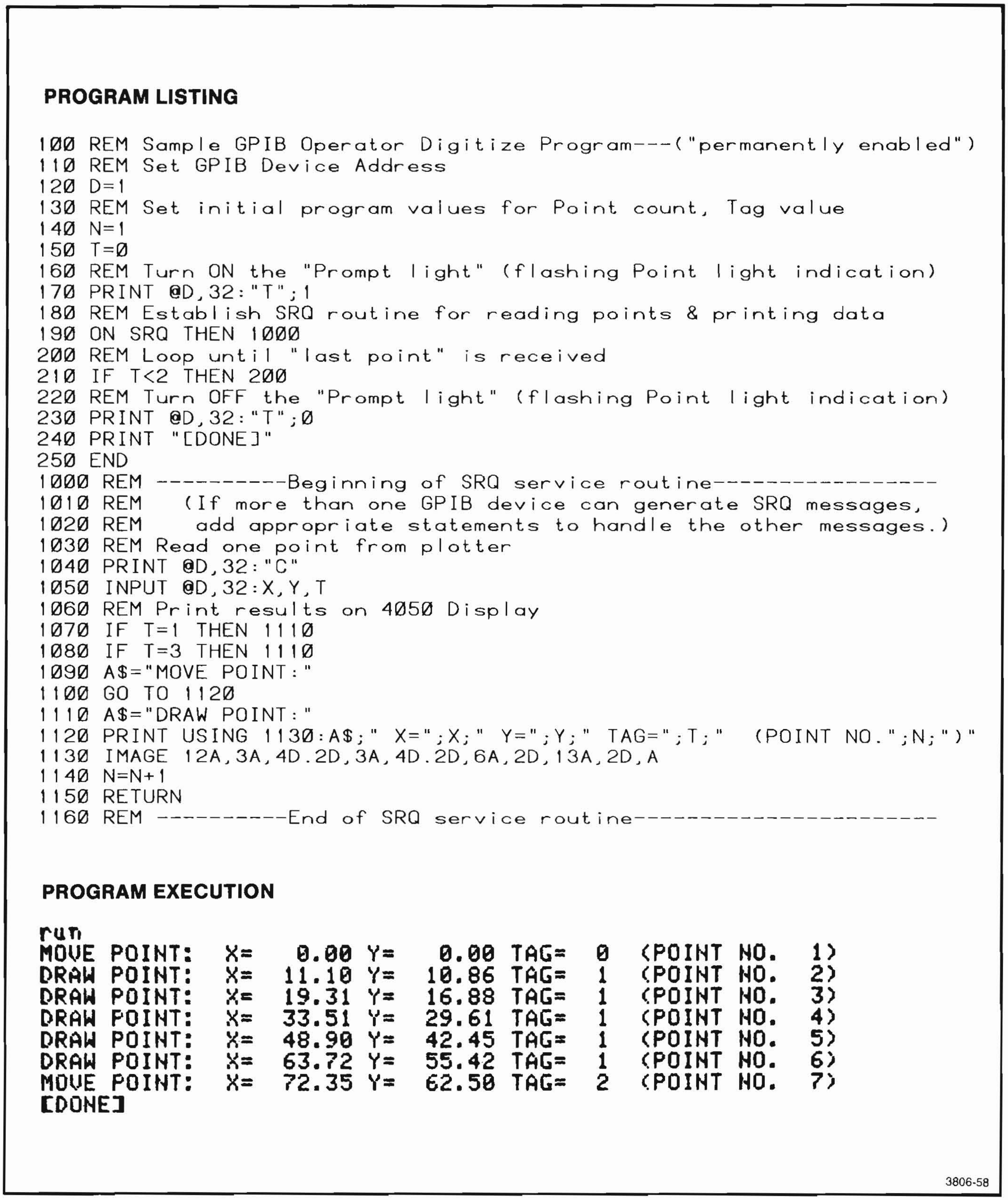
Operator digitizing via GPIB can be performed without the need for or generation of SRQ (Status Request) messages. If a CALL DIGITIZE command is sent before each POINT switch function is pushed, the Plotter responds with data without sending an SRQ message.

Use the last two operator digitizing procedures with two changes:

- Step 3 is ignored.
- Step 4 is performed immediately before Step 2 (instead of after Step 3).

NOTE

If the Plotter operator pushes the next POINT switch function before the GPIB controller has been able to send the next CALL DIGITIZE command, an SRQ message will result. If using a TEKTRONIX 4050 Series controller, or any controller that recognizes SRQ's, a "do nothing" SRQ service routine is required to satisfy the unexpected SRQ's.



PROGRAM LISTING

```
100 REM Sample GPIB Operator Digitize Program---("permanently enabled")
110 REM Set GPIB Device Address
120 D=1
130 REM Set initial program values for Point count, Tag value
140 N=1
150 T=Ø
160 REM Turn ON the "Prompt light" (flashing Point light indication)
170 PRINT @D,32:"T";1
180 REM Establish SRQ routine for reading points & printing data
190 ON SRQ THEN 1ØØØ
2ØØ REM Loop until "last point" is received
21Ø IF T<2 THEN 2ØØ
22Ø REM Turn OFF the "Prompt light" (flashing Point light indication)
23Ø PRINT @D,32:"T";Ø
24Ø PRINT "[DONE]"
25Ø END
1ØØØ REM ----------Beginning of SRQ service routine-----------------
1Ø1Ø REM   (If more than one GPIB device can generate SRQ messages,
1Ø2Ø REM    add appropriate statements to handle the other messages.)
1Ø3Ø REM Read one point from plotter
1Ø4Ø PRINT @D,32:"C"
1Ø5Ø INPUT @D,32:X,Y,T
1Ø6Ø REM Print results on 4Ø5Ø Display
1Ø7Ø IF T=1 THEN 111Ø
1Ø8Ø IF T=3 THEN 111Ø
1Ø9Ø A$="MOVE POINT:"
11ØØ GO TO 112Ø
111Ø A$="DRAW POINT:"
112Ø PRINT USING 113Ø:A$;" X=";X;" Y=";Y;" TAG=";T;"  (POINT NO.";N;")"
113Ø IMAGE 12A,3A,4D.2D,3A,4D.2D,6A,2D,13A,2D,A
114Ø N=N+1
115Ø RETURN
116Ø REM ----------End of SRQ service routine----------------------
```

PROGRAM EXECUTION

```
run
MOVE POINT:  X=   0.00 Y=   0.00 TAG= 0  (POINT NO.  1)
DRAW POINT:  X=  11.10 Y=  10.86 TAG= 1  (POINT NO.  2)
DRAW POINT:  X=  19.31 Y=  16.88 TAG= 1  (POINT NO.  3)
DRAW POINT:  X=  33.51 Y=  29.61 TAG= 1  (POINT NO.  4)
DRAW POINT:  X=  48.90 Y=  42.45 TAG= 1  (POINT NO.  5)
DRAW POINT:  X=  63.72 Y=  55.42 TAG= 1  (POINT NO.  6)
MOVE POINT:  X=  72.35 Y=  62.50 TAG= 2  (POINT NO.  7)
[DONE]
```

3806-58

Figure 13-2. Sample GPIB Operator Digitize Program (Permanently Enabled).

COMMAND DESCRIPTIONS

There are six commands described in this section:

- DIGITIZE
- OPERATOR DIGITIZE ENABLE
- CALL DIGITIZE
- PROMPT LIGHT ON
- PROMPT LIGHT OFF
- JOYSTICK AXIS DISABLE

Each command is described separately in the pages that follow.

DIGITIZE

Purpose: This command causes the Plotter to send one output message block containing current X and Y position values and a TAG value. Position values are expressed in current Graphic Units.

SERIAL

(One Output Message Block Produced)

${}^{A}{}_{T}\,{}^{A}{}_{D}$ **M**

Sample: ${}^{E}{}_{C}$ **A M**

GPIB

(One Output Message Block Produced)

G

4050 BASIC (Data Byte Form)

```
          PRINT @pa,32: "G"
          INPUT @pa,32:              value-1-target-variable,
                                     value-2-target-variable,
                                     tag-value-target-variable
Samples:  PRINT @1,32: "G"
          INPUT @1,32:               V1,V2,T
```

4050 BASIC (Secondary Address Form)

```
          PRINT @pa,24:
          INPUT @pa,32:              value-1-target-variable,
                                     value-2-target-variable,
                                     tag-value-target-variable
      or
          INPUT @pa,24:              value-1-target-variable,
                                     value-2-target-variable,
                                     tag-value-target-variable
Sample:   PRINT @1,24:
          INPUT @1,32:               V1,V2,T
      or
          INPUT @1,24:               V1,V2,T
```

4050 BASIC (Special Keyword Form)

```
          GIN @pa[,24]:              value-1-target-variable,
                                     value-2-target-variable
Sample:   GIN @1:                    V1,V2
```

Outputs

value 1
X- position

value 2
Y- position

tag value
See Table 13-1

Table 13-1

TAG VALUES FOR DIGITIZE COMMAND

TAG Meaning	TAG Value				
	Serial Formats:			GPIB Formats:	
	1	2	3-4	1-4	5-6
PEN UP	4	2	0	2	0
PEN DOWN	5	3	4	3	1

References

See either the Serial Interface or GPIB Interface section (Section 4 or 5) for information on decoding output responses from the Plotter.

Examples

The samples in the syntax boxes show how to obtain the current pen position values and TAG value. Note that for the GIN Keyword, only two target variables are allowed; the TAG value is ignored.

OPERATOR DIGITIZE ENABLE

Purpose: This command enables the front panel POINT switch for a specified number of points. (A value of 0 "disables" the POINT Switch.)

SERIAL

(One Output Message Block Per Point)

	$^{A}{}_{T}$ $^{A}{}_{D}$ **AG**	number-of-points
Sample:	$^{E}{}_{C}$ **A AG**	**12**

GPIB

AG	number-of-points

4050 BASIC (Data Byte Form)

	PRINT @pa,32: **"AG";**	number-of-points
Sample:	**PRINT** @**1**,**32**: **"AG";**	**12**

Parameters

number-of-points
Number of points, 0-65535. Special values:

- 65535 = infinity
- 0 = disabled

Table 13-2

TAG VALUES FOR OPERATOR DIGITIZE RESPONSES

Meaning Assigned to TAG		TAG Value for Serial Format:		
Point Switch Enabled	**POINT Switch Permanently Enabled**	**1**	**2**	**3-4**
Move Point	Move Point	2	0	0
Draw Point	Draw Point	3	1	4
Move Point, last block or Last Point or OPERATOR DIGITIZE DISABLE	Last Point	4	2	2[a]
Draw Point, last block		5	3	

[a] Status Block is sent

Outputs

Serial Interface (except Last Point in Command/Response Formats 3-4):

value 1
X- position

value 2
Y- position

tag value
Use Table 13-2

Serial Interface (Last Point in Command/Response Formats 3-4):

value 1
0

value 2
Device status (see Table 9-5 under the READ STATUS command in Section 9)

tag value
2

GPIB Interface:

This command does not request output. Use the CALL DIGITIZE command or the abbreviated command shown under COMMENTS to request output. Use one request per point.

Comments

This command enables the POINT switch for a specified number of points. The Plotter operator can then locate and store each point using the Joystick and the POINT switch, respectively.

The default POINT switch mode is permanently enabled for Serial Formats 3 and 4 and for GPIB Formats 5 and 6. Disabled is the default POINT switch mode for all other Formats.

If the POINT switch is enabled and an OPERATOR DIGITIZE ENABLE command with 0 argument (to disable the command) is received, the Plotter responds as if the front panel Last Point function had been selected. If the POINT switch is not enabled, the disable is ignored.

GPIB only: The following abbreviation of the 4050 BASIC form for the CALL DIGITIZE command (the command code is omitted) or the equivalent GPIB action can be used to request each output message (one per point) after each Point switch action:

```
INPUT @1,32:     value-1-target-variable,
                 value-2-target-variable,
                 tag-value-target-variable
```

Sample:

```
INPUT @1,32:     V1,V2,T
```

The output values are identical to those described for the CALL DIGITIZE command.

References

See either the Serial Interface or GPIB Interface sections (Section 4 or 5) for information on decoding output responses from the Plotter.

Examples

The samples in the syntax boxes enable the POINT switch for up to 12 points. Eleven points are enabled for Serial Formats 3 and 4 or for any other Format where the last block provides control rather than position information.

CALL DIGITIZE

Purpose: This command (GPIB only) is used to request position data stored by the front panel POINT switch function during operator digitize actions. The Plotter responds to the CALL DIGITIZE command by sending one output message block containing X and Y position values and a TAG value. Position values are expressed in current Graphic Units.

GPIB

```
            (One Output Message Block is Produced)
            C
```

4050 BASIC (Data Byte Form)

```
            PRINT @pa,32: "C"
            INPUT @pa,32:               value-1-target-variable,
                                        value-2-target-variable,
                                        tag-value-target-variable
   Sample:  PRINT @1,32: "C"
            INPUT @1,32:                V1,V2,T
```

4050 BASIC (Secondary Address Form)

```
            PRINT @pa,27:
            INPUT @pa,32:               value-1-target-variable,
                                        value-2-target-variable,
                                        tag-value-target-variable
       or
            INPUT @pa,27:               value-1-target-variable,
                                        value-2-target-variable,
                                        tag-value-target-variable
  Samples:  PRINT @1,27:
            INPUT @1,32:                V1,V2,T
       or
            INPUT @1,27:                V1,V2,T
```

Outputs

value 1
X- position

value 2
Y- position

tag value
See Table 13-3

Table 13-3

TAG VALUES FOR CALL DIGITIZE COMMAND

TAG Meaning		
POINT Switch Enabled	**POINT Switch Permanently Enabled**	**TAG Value**
Move Point	Move Point	0
Draw Point	Draw Point	1
Move Point last block or Last Point or OPERATOR DIGITIZE DISABLE	Last Point	2
Draw Point last block		3

Comments

The CALL DIGITIZE command can be used in two ways described below. The two methods differ in regard to generation and use of GPIB SRQ messages.

- The CALL DIGITIZE command is sent *after* the POINT switch is pushed; the Plotter generates an SRQ message when the POINT switch is pushea. The controller then sends a CALL DIGITIZE command to obtain the one-block output message.
- The CALL DIGITIZE command is sent *before* the POINT switch is pushed; the Plotter sends a one-block output message as soon as data is available. No SRQ message is sent. (Note: If the operator is too fast and generates another point before the next CALL DIGITIZE command, an SRQ message will be sent. For controllers that respond to SRQ, a "do nothing" SRQ service routine may be needed.)

Format 5 and 6 only: The front panel Last Point function generates a last block Move Point indication. This response is NOT available from the 4662 Plotter; the 4662 front panel CALL switch send status operation is the nearest function, but generates different output information.

References

See Section 5 for information on decoding output responses from the Plotter.

Examples

The samples in the syntax boxes show how to acquire data produced by a POINT switch function.

PROMPT LIGHT ON

Purpose: This command produces a flashing POINT light indication.

SERIAL

	$^{A}{}_{T}$ $^{A}{}_{D}$ **K**	
Sample:	$^{E}{}_{C}$ **A K**	

GPIB

	T	selector

4050 BASIC (Data Byte Form)

	PRINT @pa,32: **"T"**;	selector
Sample:	**PRINT @1,32: "T";**	1

4050 BASIC (Secondary Address Form)

	PRINT @pa,26:	selector
Sample:	**PRINT @1,26:**	1

Parameters

selector

Prompt light selector,0-1:

- 1 = ON
- 0 = OFF (see PROMPT LIGHT OFF command)

Comments

The flashing POINT light indication set by this command is overridden by any subsequent action that will set or clear a steady POINT light indication (for example, an OPERATOR DIGITIZE ENABLE command or a POINT switch function that terminates operator digitizing).

Serial Command/Response Formats 3 and 4 only: The Prompt light on state is one of several factors that enable prompt string scan action when a prompt string is defined. For details, refer to the *Communicating with Other Devices* discussion in Section 4.

Serial interface only: This command does not have an argument.

References

See the PROMPT LIGHT OFF command description.

Examples

The samples in the syntax boxes set the Prompt light on, as indicated by a flashing POINT light.

PROMPT LIGHT OFF

Purpose: This command cancels the flashing POINT light indication set by the PROMPT LIGHT ON command.

SERIAL

	$^{A}{}_{T}$ $^{A}{}_{D}$ **L**
Sample:	$^{E}{}_{C}$ **A L**

GPIB

T	selector

4050 BASIC (Data Byte Form)

	PRINT @pa,32: "T";	selector
Sample:	**PRINT @1,32: "T";**	**0**

4050 BASIC (Secondary Address Form)

	PRINT @pa,26:	selector
Sample:	**PRINT @1,26:**	**0**

Parameters

selector
Prompt light selector, 0-1:

- 0 = OFF
- 1 = ON (See PROMPT LIGHT ON command)

Comments

Refer to PROMPT LIGHT ON command.

Examples

The samples in the syntax boxes cancel the Prompt light on, and end the flashing POINT light indication.

JOYSTICK AXIS DISABLE

Purpose: This command disables Joystick motion along the axis (or axes) specified.

SERIAL

	^{A}T ^{A}D **BJ**	axis-selector
Sample:	^{E}C **A BJ**	**1**

GPIB

	BJ	axis-selector

4050 BASIC (Data Byte Form)

	PRINT @pa,32: **"BJ"**;	axis-selector
Sample:	**PRINT @1,32: "BJ";**	**1**

Parameters

axis-selector
Axis-to-be-disabled selector, 0-3:

- 0 = neither axis
- 1 = X-axis
- 2 = Y-axis
- 3 = both axes

Comments

The disabled axis (or axes) is automatically enabled by the next command or front panel action that causes pen carriage motion.

Errors

Error 44
(Integer Outside Legal Range) The Plotter substitutes a value of "0".

Section 14

MACRO COMMANDS

INTRODUCTION

The programmable macros (available only with Option 31) allow the user to store groups of frequently used Plotter commands. Each of these groups of commands can later be executed by sending an EXPAND MACRO command. This capability can be used in many applications to reduce the number of commands transmitted to the Plotter, thereby reducing plotting time.

CONCEPTS

The following general concepts are presented:

- Macro fundamentals
- Restricted commands
- Macro storage requirements

MACRO FUNDAMENTALS

Macro fundamentals include both an overview of programmable macros and a discussion of the graphics environment during macro use.

Overview of Programmable Macros

The programmable macro feature allows a group of Plotter commands preceded by a BEGIN MACRO DEFINITION command and followed by an END MACRO command to be identified by a macro number and saved in the Plotter memory. When the Plotter receives an EXPAND MACRO command for a specified macro, the Plotter executes the commands defined within the macro. A macro can contain EXPAND MACRO commands, allowing macros to expand other (nested) macros. With this capability, complex graphic patterns can be generated using only a few simple patterns.

NOTE

A macro definition cannot include an EXPAND MACRO command that references itself either directly or indirectly through any other EXPAND MACRO commands.

The SET AUTO MACRO command designates a specified macro to be automatically expanded at the end of each DRAW, DRAW ARC, and DRAW CIRCLE command. The macro expansion begins at the end of the line or arc, or at the center of the circle.

The Graphics Environment and Macros

The commands stored in programmable macros are processed, before being stored, according to the current Graphic Units, Device Units, and coordinate type. If these parameters are subsequently changed before the macro is expanded, the macro will retain the original parameters. For example, if a macro is defined while the coordinate type is set to relative, the Moves and Draws within the macro specification will be stored as relative Moves and Draws. If the coordinate type is changed to absolute and the macro is subsequently "expanded," the Moves and Draws within the macro are still executed as relative Moves and Draws.

Selections on certain Parameter Entry Card lines must be the same when the macro is defined and when it is expanded. These lines are: Media Form, Initial Page Size, Initial Page Format, Page Orientation, and Initial Aspect Ratio.

RESTRICTED COMMANDS

Commands from the following command groups will not work as expected if included in macro definitions. Do not use these commands.

- Any of the commands listed in either the Serial or GPIB Interface sections (Sections 7 and 8).
- Any commands that generate output, including the OPERATOR DIGITIZE ENABLE command.
- A DEVICE RESET command.
- A PROMPT LIGHT ON command or a PROMPT LIGHT OFF command.
- A SELECT GRAPHIC UNITS command.
- A SELECT DEVICE UNITS command.
- Any of the commands listed in the Downloadable Characters Commands section (Section 15).
- A DELETE MACRO command.
- A SET AUTO MACRO command.
- A READ MACRO STATUS command.

MACRO STORAGE REQUIREMENTS

This section describes how to determine the number of memory bytes required to store any macros transmitted to the Plotter. There are two main topics:

- How to determine the number of bytes required to store the internal form of the individual commands within the macros.
- How to determine the number of bytes required to store a group of macros (which includes control information and the individual macro definition commands).

Macro Size

The following describes how to determine the size of a specific macro. The macro size is the number of bytes required to express the internal form of the individual commands within the macros. The total storage required by a macro is larger than the macro size value and will be described in the following discussion, *Macro Storage*.

For macros already transmitted to the Plotter, the READ MACRO STATUS command returns the macro size value for each current macro. For this case, skip to the following discussion (*Macro Storage*) to determine the storage required. Calculate the size of a specific macro by adding the bytes required to store the internal form of each command in the macro. The bytes required for a command can be determined as follows:

- Each command requires two bytes plus the bytes required to store the command arguments.
- A numeric argument requires five bytes.
- A string argument consisting of a single character (a character argument) requires one byte.
- Commands with string arguments of more than one character are divided into multiple commands, each with a character argument.
- Commands which allow multiple groups of arguments are divided into one command for each argument group.

Now, after the commands have been divided (if required), the size of each command is as follows:

```
Command Size = 2
             + (1)  (number of character
                    arguments)
             + (5)  (number of numeric
                    arguments)
```

The macro size is then the sum of the command size values for each command in the macro.

The BEGIN MACRO DEFINITION and END MACRO commands are not included in the macro size calculation. Note that the macro size value may be greater than or less than the number of characters required to transmit the macro to the Plotter.(Refer to *Input Buffering* in either Section 4 or 5 of this manual for additional information about the external and internal form of commands.) Some example commands and their associated sizes are shown in Table 14-1.

Table 14-1

MACRO SIZE EXAMPLES

Command	Argument	Size
OUTLINE VIEWPORT	No Argument	2
PRINT	A	3
PRINT	ABC	9
DRAW ARC	50,50,50,0	22
DRAW CIRCLE	100	7
DRAW CIRCLE	100,200,300	21
DRAW	5,10,10,20	24

Macro Storage

This section describes how to determine the total memory required to store a specific group of macros using either the READ STATUS command or a prescribed procedure.

If the macros are already programmed, the actual storage used can be determined using the READ STATUS command as follows:

1. Read the Current Available Memory value from the Plotter (via the READ STATUS command).
2. Download the macros.
3. Repeat Step 1.
4. The difference between the value in Step 1 and the value in Step 3 is the memory used to store the group of macros.

NOTE

The amount of memory reported as allocated for macro storage using this procedure will be in 64-byte increments as the macros are referenced.

A second method to determine macro storage requirements is as follows: The total storage required for a group of macros is the sum of the storage required for each macro plus the storage required for a directory of each of the current macros. (The directory is automatically maintained.) The method used to determine macro size was discussed previously.

To determine the storage bytes required for each macro, an adjusted macro size value is computed by adding 19 bytes to the macro size value to account for control information storage. If this information is an exact multiple of 62, then the adjusted macro size value is increased by 62 bytes. Next, the adjusted macro size value is multiplied by 1.04 and rounded up to the next higher multiple of 64 bytes. The resultant value is the number of storage bytes allocated to the storage of this macro. (This computation accounts for storage in blocks of 64 bytes.)

Each defined macro has a corresponding entry in the macro directory that requires eight bytes of storage. To determine the storage bytes required for the directory, multiply the number of macros to be defined by eight. If this value is an exact multiple of 62, then increase the value by 62 bytes. The resultant value is then multiplied by 1.04 and rounded up to the next higher multiple of 64 bytes. The resultant value is the number of storage bytes allocated to the storage of a macro directory of this size.

The total storage required for a given group of macros, then, is the sum of the storage required for each macro plus the storage required for the macro directory.

A standard 4663 with 8K bytes of memory installed has about 3700 bytes of memory available for macro storage if the Serial interface is active; about 3900 bytes are available with the GPIB interface active. This amount of memory will accomodate approximately 50 small macros.

The available memory is also used on a first-come, first-served basis for the storage of:

- Transforms saved by a SAVE CURRENT TRANSFORM command.
- The buffering of output responses.
- The storage of downloadable characters.
- The storage of commands in the input buffer.

COMMAND DESCRIPTIONS

There are six commands described in this section:

- BEGIN MACRO DEFINITION
- END MACRO DEFINITION
- EXPAND MACRO
- DELETE MACRO
- READ MACRO
- SET AUTO-MACRO

Each command is described in turn in the following pages.

BEGIN MACRO DEFINITION

Purpose: This command begins a macro definition.

SERIAL

	$^{A}_{T}$ $^{A}_{D}$ **BB**	macro-number
Sample:	$^{E}_{C}$ **A BB**	**2**

GPIB

BB	macro-number

4050 BASIC (Data Byte Form)

	PRINT @1,32: **"BB";**	macro-number
Sample:	**PRINT @1,32: "BB";**	**2**

Parameters

macro-number
Macro number, 0-255

Comments

This command begins the definition of the specified macro. Any previously defined macro with the same number is deleted. The Plotter commands that follow are stored in the specified macro.

Errors

The following errors may occur during the macro definition process:

Error 85
(Insufficient memory for buffer allocation) Insufficient memory exists to begin or continue storage of the macro. The macro definition is ignored.

Error 42
(Illegal macro command usage) A restricted command is encountered and the macro definition is ignored.

END MACRO DEFINITION

Purpose: This command ends the macro definition.

SERIAL

$^{A}{}_{T}$ $^{A}{}_{D}$ **BE**

Sample: $^{E}{}_{C}$ **A BE**

GPIB

BE

4050 BASIC (Data Byte Form)

PRINT @pa,32: **"BE"**

Sample: **PRINT @1,32: "BE"**

Errors

Error 85
(Insufficient memory for buffer allocation)
Insufficient memory exists and the macro definition is deleted.

EXPAND MACRO

Purpose: This command causes the Plotter to perform the instructions defined in the specified macro.

SERIAL

	$^{A}{}_{T}$ $^{A}{}_{D}$ **AE**	macro-number[,macro-number] . . .
Sample:	$^{E}{}_{C}$ **A AE**	**0,1,14,22,10,0**

GPIB

	AE	macro-number[,macro-number] . . .

4050 BASIC (Data Byte Form)

	PRINT @pa,32: **"AE"**;	macro-number[;macro-number] . . .
Sample:	**PRINT @1,32: "AE";**	**0;1;14;22;10;0**

Parameters

macro-number
Macro number, 0-255

Errors

Error 86
(Cannot find specific macro) The command is ignored.

Error 43
(Macro called itself) The command to call itself is ignored, even if called from another macro definition within the first macro definition.

DELETE MACRO

Purpose: This command deletes the specified macro or macros from Plotter memory.

SERIAL

$\mathrm{A_T}\ \mathrm{A_D}$ **BK** [macro-number[,macro-number] . . .]

Sample: $\mathrm{E_C}$ **A BK**

GPIB

BK [macro-number[,macro-number] . . .]

4050 BASIC (Data Byte Form)

PRINT @pa,32: **"BK"** [;macro-number[;macro-number] . . .]

Sample: **PRINT @1,32: "BK";**

Parameters

macro-number

Macro number, 0-255, omitted macro number means all macros

Comments

If the specified macro is not defined, there is no action. When a macro is deleted, the storage used by the macro is released for other purposes.

READ MACRO STATUS

Purpose: This command causes the Plotter to report the status of defined macros.

SERIAL

(Command/Response Formats 1-2 Only)
(One Output Message Block Per Macro)

$^{A}{}_{T}$ $^{A}{}_{D}$ **CS**

Sample: $^{E}{}_{C}$ **A CS**

GPIB

(One Output Message Block Per Macro)

CS

4050 BASIC (Data Byte Form)

```
PRINT @pa,32: "CS"
INPUT @pa,32:               value-1-target-variable,
                            value-2-target-variable,
                            tag-value-target-variable
```

NOTE

Execute one INPUT statement for each output message block.

```
Sample:  PRINT @1,32: "CS"
         INPUT @1,32:       V1,V2,T1
         INPUT @1,32:       V3,V4,T2
         INPUT @1,32:       V5,V6,T3
```

Outputs

There is one output message block for each macro and one when no macro is defined. The values for each message block are as follows:

value 1
Macro number, range 0-255

value 2
Macro size (see COMMENTS)

tag value
See Table 14-2

Table 14-2

TAG VALUE BY FORMAT FOR READ MACRO STATUS COMMAND

	TAG Value by Format		
TAG Meaning	Serial Format:		GPIB Formats:
	1	2	1-6
Not the Last Block	2	0	0
Last (or Only) Block	4	2	2

Comments

The Plotter responds to this command with one output block for each macro defined. Execute one INPUT statement for each output message block until a last block TAG is read. The macro size value reported indicates the number of bytes required to store the internal form of the individual commands included in the macro. However, the complete macro requires a larger amount of storage, which can be computed using the macro size value. Refer to the discussion of macro storage requirements earlier in this section for a description of how to determine the number of memory bytes required to store the complete macro.

If no macros are currently defined, the following information is sent:

• value 1	0
• value 2	0
• tag value	For last or only block (see Table 14-1).

References

See either Section 4 or 5 for information on decoding output responses from the Plotter.

SET AUTO MACRO Command

Purpose: This command designates a specified macro to be automatically expanded at the end of each DRAW, DRAW ARC, and DRAW CIRCLE command.

SERIAL

	$^{A}{}_{T}$ $^{A}{}_{D}$ **BN**	[macro-number]
Sample:	$^{E}{}_{C}$ **A BN**	**0**

GPIB

BN	[macro-number]

4050 BASIC (Data Byte Form)

	PRINT@pa,32: **"BN"**	[;macro-number]
Sample:	**PRINT@1,32: "BN";**	**0**

Parameters

macro-number
Macro number, 0-255, omitted macro number means no auto macro

Comments

This macro function is cleared by either the DEVICE RESET command or by sending the SET AUTO MACRO command with no macro number.

Errors

Error 86
(Cannot find specific macro) Auto macro is canceled.

Error 43
(Macro called itself) The command to call itself is ignored, even if called from another macro definition within the first definition.

Section 15

DOWNLOADABLE CHARACTER COMMANDS

INTRODUCTION

Downloadable characters (available only with Option 32) permit special characters and/or symbols to be defined (stored in Plotter memory). Once these characters have been defined, they can be printed by using all the Alpha features available for the resident alpha character definitions. These definitions can supercede resident character definitions or add definitions not currently available. Only the 96 printing ASCII characters ($^{S}{}_{P}$ through $^{D}{}_{L}$) may be defined. A maximum of 16 fonts (0 through 15) can be used.

Along with the ability to define new characters, Option 32 also provides the user with three predefined fonts (Fonts 10, 11, and 15). (These fonts were shown in Figures 10-4, 10-5, and 10-6 earlier in this manual.) Fonts 10 and 11 contain special mathematical symbols and Greek letters. Font 15 contains a single character $^{D}{}_{T}$ (DEL) defined as a "low resolution" coordinate grid usable as a design aid when defining downloadable characters. The following paragraphs define the use of low resolution coordinates.

CONCEPTS

The following concepts are discussed in this section:

- Defining a Character
- Storage Requirements
- Printing Downloadable Characters
- Option 32 Characters

SETUP REQUIREMENTS

The SET GRAPHIC UNITS command and the SET COORDINATE TYPE command must be set to default arguments while downloading character definitions.

DEFINING A CHARACTER

Listed below are the commands to use in defining a downloadable character. If a command is required in the definition of a specific character, then it must be sent in the order indicated here.

Required:

- [BEGIN CHARACTER DEFINITION]
- [SET CHARACTER X-EXTENT]

These can be used in any order or combination to describe the character:

- [MOVE] . . .
- [DRAW] . . .
- [DRAW ARC] . . .
- [DRAW CIRCLE] . . .
- [SPECIFY NONADVANCING CHARACTER]

Optional (if next command is BEGIN COMMAND DEFINITION):

- [END CHARACTER DEFINITION]

It is legal to terminate one character definition by beginning another.

Data Ranges for Commands

The legal range of arguments for the character defining commands (MOVE, DRAW, DRAW ARC, and SET CHARACTER X-EXTENT) is xmin — 64 through xmin + 175 in the X direction, and ymin — 64 through ymin + 175 in the Y direction. For the DRAW CIRCLE command, radius values may be as large as 175.

Downloaded Character Size

The shaded rectangle in Figure 15-1 illustrates the data ranges used to define a typical uppercase character. Alpha size and spacing actions assume that characters are defined within the shaded rectangle. (Characters can be defined as smaller or larger than the shaded rectangle, and will be printed accordingly.)

For most applications, use the default data ranges, shown in parentheses in Figure 15-1. Typical uppercase character data ranges are between 0 and 48 for the X-axis and between 0 and 88 for the Y-axis. Lowercase letters descend to —24 for characters used with other resident font characters. Special symbols may be larger or smaller, as desired.

For other applications, use the SET DOWNLOADED CHARACTER SIZE command (1) to select different data ranges, (2) to utilize an existing character data base, (3) to allow more character definition resolution (for example, 175 by 175 vs. 48 by 88), or (4) to avoid the use of negative values when defining characters with descenders. The SET DOWNLOADED CHARACTER SIZE command sets the ranges for X and Y data. The four specified values correspond to the extremes of the shaded rectangle in Figure 15-1. Two examples illustrate use of this command:

- For character data on a 0 to 6, 0 to 11 grid (for typical uppercase characters), use SET DOWNLOADED CHARACTER SIZE values of 0,6,0,11.
- To avoid sending negative values for the same character data that includes descenders that reach —3, add 3 to all Y data after entering SET DOWNLOADED CHARACTER SIZE values of 0,6,3,14.

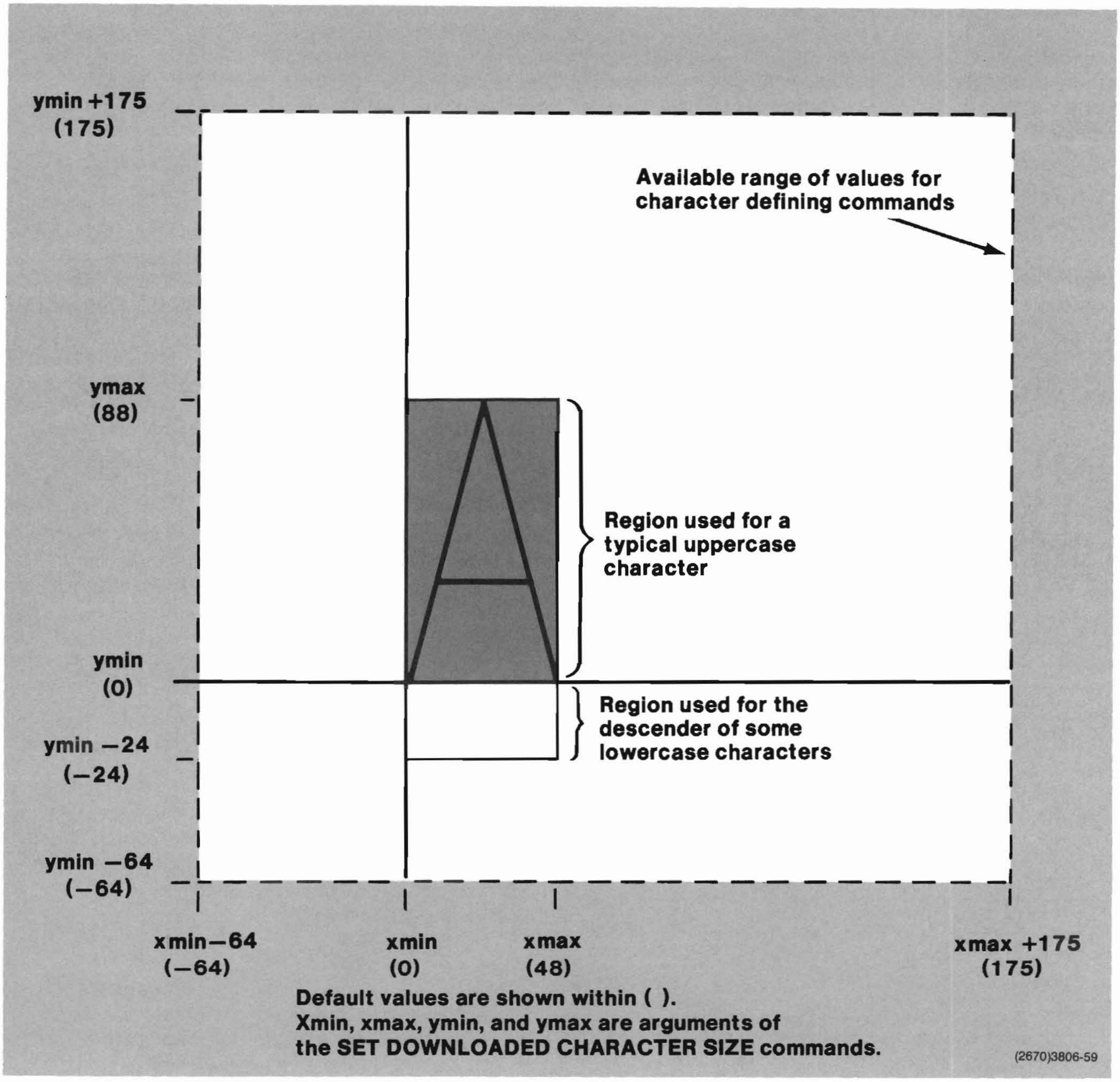

ymin +175
(175)
Available range of values for character defining commands
ymax
(88)
Region used for a typical uppercase character
ymin
(0)
Region used for the descender of some lowercase characters
ymin −24
(−24)
ymin −64
(−64)
xmin−64
(−64)
xmin
(0)
xmax
(48)
xmax +175
(175)
Default values are shown within ( ).
Xmin, xmax, ymin, and ymax are arguments of the SET DOWNLOADED CHARACTER SIZE commands.
(2670)3806-59

Figure 15-1. Numerical Value Ranges for Character Definition Commands.

STORAGE REQUIREMENTS

Described here are ways to determine the number of memory bytes required to store downloadable characters transmitted to the Plotter. There are two topics discussed:

- How to determine the number of bytes required to store the internal form of the individual commands within the downloadable characters.
- How to determine the number of storage bytes required to store a group of downloadable characters.

Storage Bytes for Each Command Within a Character Definition

The following describes how to determine the number of bytes of internal storage for each specific command within a downloadable character definition. The total storage required for a downloadable character is larger (see *Determining Total Character Storage*). The bytes required to store each of the commands used in a downloadable character definition are shown in Table 15-1.

Table 15-1

STORAGE BYTES REQUIRED FOR COMMANDS IN A CHARACTER DEFINITION

Command	High Resolution	Low Resolution
SET CHARACTER X-EXTENT	2 (if right= 48+ xmin) 4 (if right≠48+ xmin)	none
MOVE	3	1
DRAW	3	1
DRAW ARC	5	3
DRAW CIRCLE	2	none
SPECIFY NONADVANCING CHARACTER	1	none

If a MOVE, DRAW, or ARC command (within a downloaded character definition) has arguments that meet both of the following conditions, the command will be stored in a low resolution form (commands not meeting these conditions are stored in the high resolution form):

- Arguments are expressed in integer multiples of eight from xmin or ymin.
- Arguments fall within an X range of xmin to xmin + 48 and a Y range from ymin — 24 to ymin + 96. (The xmin and ymin values are the default values of 0 and 0, or are specified by the SET DOWNLOADED CHARACTER SIZE command.)

If the command is a SET CHARACTER EXTENT command and the right edge is equal to xmin + 48, the internal storage requirement is two bytes, otherwise four bytes.

NOTE

Printing the $^{\mathrm{D}}\mathrm{T}$ character from Font 15 will draw a low resolution coordinate grid, which may be used as an aid when defining low resolution characters.

Determining Total Character Storage

Described here are two ways to determine the total memory required to store a specific group of downloadable characters:

Measure the amount of current available memory using the READ STATUS (1) command, both before and after downloading characters. The difference between the two readings is the actual storage used for the downloaded characters.

NOTE

The amount of memory reported as allocated for the storage of a downloaded character set using this procedure will be in 64-byte increments from the value determined by computation.

Alternatively, the downloadable character storage requirements may be computed as described in the following paragraphs.

The total storage required for a given group of downloaded characters is the sum of the storage required for each of the fonts containing characters plus the storage required for the font directory.

First determine the storage bytes required for each font. Calculate the sum of the storage bytes needed for the individual character-defining commands plus two bytes for each defined character plus 21 bytes. If this sum is an exact multiple of 62, add 62 (otherwise, use the original sum).

Then multiply the sum by 1.04 and round the result to the next higher multiple of 64. (This computation accounts for the fact that memory is allocated for storage in blocks of 64 bytes.) Repeat this procedure for each additional font.

NOTE

Even if downloadable characters are deleted from a font, the deleted character must still be included in the font storage calculation. The deleted character space is not released until the font is deleted.

The storage bytes required for the internal font is determined by the number of downloaded (complete or partial) fonts in use, as follows:

64 bytes for 0 through 7 fonts
128 bytes for 8 through 15 fonts
192 bytes for the 16th font

Total storage is the sum of the storage requirements for the fonts and the font directory. For two examples of these procedures, refer to Table 15-2. In the table, Case A is based on 46 characters of 100 high resolution strokes each. Case B is based on 12 characters of 100 high resolution strokes each.

Table 15-2

DETERMINING TOTAL CHARACTER STORAGE

Procedure	Case A[a]		Case B[b]	
1. Strokes	96 x 100 x 3	= 2880	12 x 100 x 3	= 3600
Add (2) x (chars)	96 x 2	= 192	12 x 2	= 24
Add 21		+ 21		+ 21
Subtotal		3093		3645
2. If not exact multiple of 62, add 62	(3093÷62= 49.88 ≠ integer; thus add 62)	+ 62	(3645÷58.79 ≠ integer; thus add 62)	+ 62
Subtotal		3155		3707
3. Multiply by 1.04	3155 x 1.04	= 3281.2	3707 x 1.04	= 3855.28
4. Round up to multiple of 64	3281.2÷64= 51.27 64 x 52	= 3328	3855.28÷64= 60.24 64 x 61	= 3904
5. Add font direct (assume < 8 fonts)		+ 64		+ 64
Total		3392		3968

[a] 96 characters of 100 high resolution strokes each
[b] 12 characters of 100 high resolution strokes each

There are about 3700 bytes of memory available on a standard 4663 with the Serial interface; with the GPIB interface active, there are about 3900 bytes available.

The available memory is used on a first-come, first-served basis in the following order:

1. The storage of commands in the input buffer.
2. The storage of downloadable characters.
3. The storage of programmable macros.
4. The buffering of output responses.
5. Transforms saved by a SAVE CURRENT TRANSFORM command.

This amount of memory accommodates, for example, the storage of about:

- 96 coarse characters (with an average of 10 high resolution strokes per character in a single font), or
- about 12 very complex characters (with an average of 100 high resolution strokes per character in a single font).

PRINTING DOWNLOADED CHARACTERS

The following text presents information on printing downloadable characters.

Printed Character Quality

Downloaded characters and predefined characters in Fonts 10, 11, and 15 are not affected by the Alpha Character Quality line selection on the Parameter Entry Card. Their circles and arcs are, however, subject to the control of the SET ARC SMOOTHNESS command.

Fonts 10 and 11 Greek and Math character definitions exhibit ENHANCED (vs. STANDARD) character quality. Downloaded character definitions, when used with resident characters, should be defined to match the quality of the resident characters that will be selected.

Priority of Character Definitions

The plotter identifies each character contained in a PRINT command as (1) a downloadable character, (2) a character defined in an optional character table, or (3) a character defined in one of the resident character tables. When the Plotter receives the character to be printed, the entries in these tables, which define the specified character in the current active font, are searched in a specific order. This order is shown in Table 15-3 for a given font n.

According to Table 15-3, the first definition the Plotter encounters is used. This search order establishes a priority between the character tables. If a definition for the character is not found in any of these tables for Font n, then the Plotter uses the resident Font 0 definition. The redefinition of a Font 0 downloadable character does not affect the characters obtained in Fonts 1 to 15, which default to the resident Font 0 table for their definition.

Table 15-3

PRINTING PRIORITIES

Priority	Character Set
1 (highest)	Downloaded Font n character
2	Resident Option 32 Font n character
3	Resident Font n character
4 (lowest)	Resident Font 0 character

For example, if the character "A" is defined in Font 14 via a downloadable character definition (and no other downloadable character definitions are present), then printing a Font 14 "A" causes the downloaded definition to be used. Printing a Font 14 "B" causes the resident Font 0 "B" definition to be used. Printing a Font 8 "A" causes the resident Font 0 "A" to be used.

OPTION 32 CHARACTERS

The Option 32 character Fonts 10, 11, and 15 were shown previously in Figures 10-4, 10-5, and 10-6.

NOTE

When using Font 10 and 11 characters, select ENHANCED on the Alpha Character Quality line of the Parameter Entry Card. This selection will cause the Plotter to draw Font 0 default characters with the same quality as that of the Option 32 characters.

COMMAND DESCRIPTIONS

There are seven commands described in this section:

- SET DOWNLOADED CHARACTER SIZE
- BEGIN CHARACTER DEFINITION
- END CHARACTER DEFINITION
- SET CHARACTER X-EXTENT
- SELECT NONADVANCING CHARACTER
- DELETE CHARACTER DEFINITION
- DELETE FONT DEFINTION

Each command is described separately in the pages that follow.

SET DOWNLOADED CHARACTER SIZE

Purpose: This command specifies the coordinate ranges to be used when defining a typical uppercase character within the specified font. Default ranges are 0 to 48 (X-axis) and 0 to 88 (Y-axis).

SERIAL

	AT AD **CZ**	font-selector,xmin,xmax,ymin,ymax
Sample:	EC **A CZ**	**1,0,6,0,11**

GPIB

	CZ	font-selector,xmin,xmax,ymin,ymax

4050 BASIC (Data Byte Form)

	PRINT @pa,32: **"CZ"**;	font-selector;xmin;xmax;ymin;ymax
Sample:	**PRINT @1,32: "CZ"**;	**1;0;6;0;11**

Parameters

font-selector
Font selector, 0-15

xmin
X-min coordinate, integer

xmax
X-max coordinate, integer

ymin
Y-min coordinate, integer

ymax
Y-max coordinate, integer

Comments

This command must be sent before attempting any definition of the first character in the font. The specifications apply to all character definitions in the font. To change the specifications, the font must be deleted and redefined.

The default values of 0,48,0,88 are used for xmin, xmax, ymin, and ymax, respectively.

The arguments of this command describe two aspects of following downloadable character definitions for this font (refer back to Figure 15-1).

First, the xmin and ymin values locate the range of integer argument values to be used in subsequent definitions. Second, the xmax and ymax values complete the specifications of the nominal uppercase character width (xmax — xmin) and the nominal uppercase character height (ymax — ymin).

The specification of these two aspects allow existing character set specifications having unique coordinate ranges (for example, 0 to 6, 0 to 11) to be used directly as downloadable character specifications.

If the transmission of negative arguments (as may be required for the specification of character descenders) to the Plotter is difficult, the character size specifications, ymin and ymax, and the character specification data may be offset in the positive direction. The offset can be the value of the maximum descender coordinate, thus avoiding the need to specify negative values. For example, use 0 xmin to 6 xmax and 3 ymin to 14 ymax for the SET DOWNLOADED CHARACTER command and add 3 to all Y values for MOVE, DRAW, and ARC commands. Descenders in Y will be specified from 0 to 3 rather than from —3 to 0.

BEGIN CHARACTER DEFINITION

Purpose: This command begins a downloadable character definition for the specified character in the chosen font.

SERIAL

	AT AD **CN**	font-selector,single-character-string
Sample:	EC **A CN**	**13,'a'**

GPIB

	CN	font-selector,single-character-string

4050 BASIC (Data Byte Form)

	PRINT@pa,32: **"CN";**	font-selector;",single-character-string"
Sample:	**PRINT@1,32: "CN";**	**13;",'a' "**

Parameters

font-selector
Font selector, 0-15

single-character-string
The character to be defined, SP through DL, noncontrol characters only

Comments

This command terminates any current downloadable character definition before beginning a new definition. If the specified character is defined as a downloadable character, the current definition is deleted before the new definition is begun.

Be sure to include the argument separator, shown as a comma (,) in the samples. If the argument separator is omitted, the command is ignored and no error message is produced.

Errors

Error 45
(Too many entries or string too long) First character of string is used.

END CHARACTER DEFINITION

Purpose: This command ends the downloadable character definition.

SERIAL

$^{A}{}_{T}$ $^{A}{}_{D}$ **CO**

Sample: $^{E}{}_{C}$ **A CO**

GPIB

CO

4050 BASIC (Data Byte Form)

PRINT@pa,32: **"CO"**

Sample: **PRINT@1,32: "CO"**

Comments

This command is not required if the next command is the BEGIN CHARACTER DEFINITION command.

Errors

Error 51
(Illegal downloadable character command) A downloadable character definition was not in process.

SET CHARACTER X-EXTENT

Purpose: This command specifies the left and right edges of the character. These are used for subsequent printing actions whenever the selection for the SELECT ALPHA SPACING CONTROL command is 1 (constant end-to-start).

SERIAL

	$^{A}_{T}$ $^{A}_{D}$ **CT**	left,right
Sample:	$^{E}_{C}$ **A CT**	**10,20**

GPIB

	CT	left,right

4050 BASIC (Data Byte Form)

	PRINT@pa,32: **"CT"**;	left;right
Sample:	**PRINT@1,32: "CT"**;	**10;20**

Parameters

left
Left edge X-coordinate

right
Right edge X-coordinate

Comments

The left and right edge coordinates define the actual X-axis left and right edges (the actual character extent) of a downloadable character. The range of these coordinates must be between xmin —64 and xmin + 175, where xmin has a default value of 0 or a value specified by the SET DOWNLOADABLE CHARACTER SIZE command.

References

See the SELECT ALPHA SPACING CONTROL command for a discussion of the use of the character extent information in the Alpha Spacing Control function.

SELECT NONADVANCING CHARACTER

Purpose: This command causes the downloadable character being defined to be designated as nonadvancing. Subsequent PRINT commands will, after printing a nonadvancing character, move the pen carriage back to the original start point for the character.

SERIAL

$^{A}{}_{T}$ $^{A}{}_{D}$ **CU**

Sample: $^{E}{}_{C}$ **A CU**

GPIB

CU

4050 BASIC (Data Byte Form)

PRINT @ pa,32: **"CU"**

Sample: **PRINT @ 1,32: "CU"**

Comments

When used, this command must be placed at the end of the downloadable character definition.

In a typical application the composite character ñ can be produced as follows:

- First, define ~ as a nonadvancing character.
- Second, include the nonadvancing ~ character as the first of the two-character sequence ~n within a PRINT command.

Errors

Error 51
(Illegal downloadable character command) A downloadable character definition was not in process.

DELETE CHARACTER DEFINITION

Purpose: This command deletes a specified character or characters from a specified user-defined font. The corresponding memory is not released for other uses.

SERIAL

	$^{A}{}_{T}$ $^{A}{}_{D}$ **CP**	font-selector,character-string
Sample:	$^{E}{}_{C}$ **A CP**	**14,'ABC'**

GPIB

CP	font-selector,character-string

4050 BASIC (Data Byte Form)

	PRINT @pa,32: **"CP";**	font-selector;",character-string"
Sample:	**PRINT @1,32: "CP";**	**14;",'ABC' "**

Parameters

font-selector
Font selector, 0-15

character-string
One or more characters

Comments

Be sure to include the argument separator, shown as a comma (,) in the samples. If the argument separator is omitted, the command is ignored, but no error message is produced.

DELETE FONT DEFINITION

Purpose: This command deletes all downloadable character definitions for the specified font and releases the corresponding memory for other uses. If no font number is specified, *all* fonts are deleted.

SERIAL

	$^{A}{}_{T}$ $^{A}{}_{D}$ **CQ**	[font-selector [,font-selector] . . .]
Sample:	$^{E}{}_{C}$ **A CQ**	**14**

GPIB

CQ	[font-selector[,font-selector] . . .]

4050 BASIC (Data Byte Form)

	PRINT @ pa,32: **"CQ"**	[;font-selector[;font-selector] . . .]
Sample:	**PRINT @ 1,32: "CQ";**	**14**

Parameters

font-selector
Font selector, 0-15

NOTE

If font-selector is omitted, all fonts are deleted.

Section 16

MEDIA CHANGE COMMANDS

INTRODUCTION

Media change operations are discussed in the *4663 Operator's Manual* in the description of the front panel MEDIA switch. The concepts and commands discussed here relate to programmable media change operations. However, some of the commands set parameters or report settings associated with the front panel media change functions.

CONCEPTS

The media change function works either of two ways, depending on the Media Form line setting on the Parameter Entry Card. If this line is set to SHEET (designated Sheet mode), then commands are processed for media secured by the electrostatic media hold-down. If Option 36 (Media Advance) is installed and the Media Form line is set to ROLL (designated Roll mode), the commands support media change using the media drive mechanism. Media changes activated by commands for both Sheet and Roll modes do the same operations as the MEDIA switch on the front panel.

In Roll mode, the length of media advance is initially set to extend 1.25 inches (31.37 mm) beyond the left edge of the page. However, the length may also be selected by commands in 1/64-inch increments (.397 mm). Thus, an advance of 640 will advance the media 10 inches (254 mm). However, advances less than one inch (25.4 mm) may produce an out-of-media indication, causing the front panel Plot Control light to blink.

COMMAND DESCRIPTION

There are four commands described in this section:

- PAGE CHANGE
- SET FORM LENGTH
- READ FORM LENGTH
- ADVANCE MEDIA

Each command is described separately in the pages that follow.

PAGE CHANGE

Purpose: This command causes a media change operation (in the same manner as the front panel Media Change switch) for either Sheet or Roll mode.

SERIAL

$^{A}{}_{T}$ $^{A}{}_{D}$ **BC**

Sample: $^{E}{}_{C}$ **A BC**

GPIB

BC

4050 BASIC (Data Byte Form)

PRINT @pa,32: **"BC"**

Sample: **PRINT @1,32: "BC"**

Comments

To continue operation when in Sheet mode, press the Media Change switch after the media has been changed.

References

See the Pens and Media section in the *4663 Operator's Manual*.

Examples

The samples in the syntax boxes show commands that initiate the media change operations. In Sheet mode, press Media Change on the front panel to continue operation after the media has been changed.

In Roll mode, the complete media advance operation will be performed without further intervention.

SET FORM LENGTH

Purpose: This command sets the form length for use either by the PAGE CHANGE command or by the front panel Media Change function.

SERIAL

	${}^{A}_{T}$ ${}^{A}_{D}$ **BF**	form-length
Sample:	${}^{E}_{C}$ **A BF**	**768**

GPIB

BF	form-length

4050 BASIC (Data Byte Form)

	PRINT @pa,32: **"BF"**;	form-length
Sample:	**PRINT @1,32: "BF";**	**768**

Parameters

form-length
Number of 1/64 inch (.397 mm) increments, nonzero, positive values only

Comments

The form length is established in positive 1/64-inch (.397 mm) increments. In Roll mode, the form length specifies the distance that the media is advanced when doing a PAGE CHANGE command or when doing a front panel Media Change.

A DEVICE RESET command (0, 1, or 3) or a Parameter Entry Card setting on any line from Media Form through Initial Aspect Ratio cause the form length to be reset to a default value. This default value is the distance from the arrow on the media guide to a point 1.25 inches (31.37mm) beyond the left edge of the Page.

Examples

The samples show how to set a form length of 12 inches (304.80 mm).

READ FORM LENGTH

Purpose: This command causes the Plotter to respond with a one-block message containing the form length.

SERIAL

(Command/Response Formats 1 and 2 Only)
(One Output Message Block Produced)

A_T A_D **CF**

Sample: E_C **A CF**

GPIB

(One Output Message Block Produced)

CF

4050 BASIC (Data Byte Form)

PRINT @ pa,32: **"CF"**
INPUT @ pa,32: value-1-target-variable,
value-2-target-variable,
tag-value-target-variable

Sample: **PRINT @1,32: "CF"**
INPUT @1,32: V1,V2,T

Outputs

value 1
Form length in 1/64 inch (.397 mm) increments

value 2
0

tag value
See Table 16-1

Table 16-1

TAG VALUES FOR READ FORM LENGTH

Formats	TAG Values
Serial Format 1	4
Serial Format 2	2
GPIB Format 1 Thru 6	2

Errors

Error 48
(Invalid Command For Selected Output Format) This error occurs if Formats 3 or 4 are used with the Serial interface.

Example

For a form length of 12 inches (304.8 mm), output Value-1 is 768.

References

Refer to Section 4 or 5 for information on decoding output responses from the Plotter.

ADVANCE MEDIA

Purpose: This command (used for Roll mode only) causes the media to advance a specified number of 1/64-inch (.397 mm) increments.

SERIAL

	^{A}T ^{A}D **AU**	increments
Sample:	^{E}C **A AU**	**128**

GPIB

	AU	increments

4050 BASIC (Data Byte Form)

	PRINT @pa,32: **"AU"**;	increments
Sample:	**PRINT @1,32: "AU";**	**128**

Parameters

increments
Number of 1/64-inch (.397 mm) increments, sign ignored

Comments

This command requires Option 36. The advance is independent of the current form length setting.

Errors

Error 41
(Illegal Paper Advance Command) Plotter is in Sheet mode.

Examples

The samples show how to advance the media two inches (50.8 mm).

Command	Format	Example
Transform Commands		
RESTORE PREVIOUS TRANSFORM	A_T A_D **AY**	$^{E}_{C}$**AAY**
SAVE CURRENT TRANSFORM	A_T A_D **AX**	$^{E}_{C}$**AAX**
SET ORIGIN TO CURRENT POS	A_T A_D **AO**	$^{E}_{C}$**AAO**
SET ROTATION	A_T A_D **AR** angle	$^{E}_{C}$**AAR30**
SET ROTATION TO LAST ANGLE	A_T A_D **AL**	$^{E}_{C}$**AAL**
SET SCALE	A_T A_D **AS** x-scale,y-scale	$^{E}_{C}$**AAS2,4**
SET SKEW	A_T A_D **AQ** x-skew-angle,y-skew-angle	$^{E}_{C}$**AAQ20,10**
SET TRANSLATION	A_T A_D **AT** x-translation,y-translation	$^{E}_{C}$**AAT512,1024**
Digitizing Commands		
DIGITIZE[a]	A_T A_D **M**	$^{E}_{C}$**AM**
JOYSTICK AXIS DISABLE	A_T A_D **BJ** axis-selector 1: x 0: none 2: y 3: both	$^{E}_{C}$**ABJ2**
OPERATOR DIGITIZE ENABLE	A_T A_D **AG** number-of-points	$^{E}_{C}$**AAG20**
PROMPT LIGHT OFF	A_T A_D **L**	$^{E}_{C}$**AL**
PROMPT LIGHT ON	A_T A_D **K**	$^{E}_{C}$**AK**
Macro Commands		
BEGIN MACRO DEFINITION	A_T A_D **BB** macro-number (0 to 255)	$^{E}_{C}$**ABB12**
DELETE MACRO	A_T A_D **BK** [macro-number[,macro-number]. . .] no argument: all macros	$^{E}_{C}$**ABK7,9** or $^{E}_{C}$**ABK**
END MACRO DEFINITION	A_T A_D **BE**	$^{E}_{C}$**ABE**
EXPAND MACRO	A_T A_D **AE** macro-number[,macro-number]. . .	$^{E}_{C}$**AAE12** or $^{E}_{C}$**AAE3,9,11**
READ MACRO STATUS[c]	A_T A_D **CS**	$^{E}_{C}$**ACS**
SET AUTO MACRO	A_T A_D **BN** [macro-number] no argument: auto macro "off"	$^{E}_{C}$**ABN12** or $^{E}_{C}$**ABN**
Downloadable Character Commands		
BEGIN CHARACTER DEFINITION	A_T A_D **CN** font-selector,single-character-string	$^{E}_{C}$**ACN4,'a'**
DELETE CHARACTER DEFINITION	A_T A_D **CP** font-selector,string	$^{E}_{C}$**ACP1,'ABC'**
DELETE FONT DEFINITION	A_T A_D **CQ** [font-selector[,font-selector]. . .] no argument: all fonts	$^{E}_{C}$**ACQ2** or $^{E}_{C}$**ACQ**
END CHARACTER DEFINITION	A_T A_D **CO**	$^{E}_{C}$**ACO**
SELECT NONADVANCING CHAR.	A_T A_D **CU**	$^{E}_{C}$**ACU**
SET CHARACTER X-EXTENT	A_T A_D **CT** left,right	$^{E}_{C}$**ACT22,26**
SET DOWNLOADED CHAR. SIZE	A_T A_D **CZ** font-selector,x-min,x-max,y-min,y-max	$^{E}_{C}$**ACZ1,0,6,0,11**
Media Advance Commands		
ADVANCE MEDIA	A_T A_D **AU** increments (1/64 inch per increment)	$^{E}_{C}$**AAU128**
PAGE CHANGE	A_T A_D **BC**	$^{E}_{C}$**ABC**
READ FORM LENGTH[a]	A_T A_D **CF**	$^{E}_{C}$**ACF**
SET FORM LENGTH	A_T A_D **BF** increments (1/64 inch per increment)	$^{E}_{C}$**ABF768**

[a] One output message block produced.
[b] Two output message blocks produced.
[c] One output message block per macro.

Appendix A

COMMAND SUMMARY (SERIAL)

Command	Format	Example
Interface Commands		
BLOCK END	A_T A_D) checksum-value	E_CA)373
BLOCK START	A_T A_D (	E_CA(
DATA RESET	A_T A_D **CD**	E_C**ACD**
DEVICE OFF	A_T A_D **F**	E_C**AF**
DEVICE ON	A_T A_D **E**	E_C**AE**
INTERFACE PARAMETER RESET	A_T A_D **CR**	E_C**ACR**
SELECT COMMAND/RESPONSE FMT	A_T A_D **CC** format-selector	E_C**ACC2**
SET BLOCK SIZE	A_T A_D **H** maximum-block-size	E_C**AH132**
SET BYPASS CANCEL CHARACTER	A_T A_D **U** bypass-cancel-char-string	E_C**AU**$^L{}_F$
SET PROMPT STRING	A_T A_D **R** prompt-string	E_C**AR"?**$^S{}_P$**"**
SET SIGNATURE CHARACTER	A_T A_D **S** signature-character-string	E_C**ASA**
SET TURNAROUND DELAY	A_T A_D **G** delay-time (in milliseconds)	E_C**AG200**
Device Commands		
DEVICE RESET	A_T A_D **N** [selector]	E_C**AN0**
IDENTIFY or SIZE[a]	A_T A_D **Q**	E_C**AQ**
READ ERROR[b]	A_T A_D **CE**	E_C**ACE**
READ STATUS[a]	A_T A_D **O** status-register-selector	E_C**AO2**
Alpha Commands		
CHARACTER MOVE	A_T A_D **AM** character-spaces, line-spaces	E_C**AM3,2**
MOVE TO HOME	A_T A_D **AH**	E_C**AAH**
PRINT	A_T A_D **P** string	E_C**AP"ABC"/68'EFG'**
PRINT CENTERED	A_T A_D **AP** string	E_C**AAP"O"**
RESET ALPHA PARAMETERS	A_T A_D **V**	E_C**AV**
SELECT ALPHA SPACING CONTROL	A_T A_D **AJ** spacing-control-selector 0: constant origin-to origin 1: constant space between characters	E_C**AJ1**
SELECT ALTERNATE ALPHA FONT	A_T A_D **BQ** alternate-font-selector	E_C**ABQ8**
SELECT STANDARD ALPHA FONT	A_T A_D **T** standard-font-selector	E_C**AT1**
SET ALPHA DIMENSION	A_T A_D **I** character-space,line-space	E_C**AI90,180**
SET ALPHA MARGIN SEPARATION	A_T A_D **BR** margin-separation (number of spaces)	E_C**ABR80**
SET ALPHA RATIO	A_T A_D **BI** x-ratio,y-ratio	E_C**ABI1.5,1.63636**
SET ALPHA ROTATION	A_T A_D **J** rotation-angle	E_C**AJ30**
SET ALPHA SCALE	A_T A_D **BH** x-mag-factor,y-mag-factor	E_C**ABH2,3**
SET ALPHA SIZE	A_T A_D **BZ** character-width,character-height	E_C**ABZ60,110**
SET ALPHA SLANT	A_T A_D **BG** slant-angle	E_C**ABG20**
SET TAB SEPARATION	A_T A_D **BT** tab-separation (number of spaces)	E_C**ABT8**

[a] One output message block produced.
[b] Two output message blocks produced.
[c] One output message block per macro.

Command	Format	Example
Graphics Commands		
AXIS	A_T A_D **CA** x-space,y-space[,x-intercept, y-intercept]	E_C**ACA100,100,1000,1000**
DRAW	A_T A_D **Y** x-value,y-value[,x-value,y-value]. . .	E_C**AY1000,2000**
DRAW ARC	A_T A_D **AA** x-mid,y-mid,x-end,y-end[,x-mid,y-mid, x-end,y-end]. . .	E_C**AAA500,500,1000,0**
DRAW CIRCLE	A_T A_D **AC** radius[,radius]. . .	E_C**AAC1000**
MARK VIEWPORT	A_T A_D **CM**	E_C**ACM**
MOVE	A_T A_D **X** x-value,y-value	E_C**AX32,40**
MOVE TO LOAD POINT	A_T A_D **AI**	E_C**AAI**
OUTLINE VIEWPORT	A_T A_D **CB**	E_C**ACB**
READ VIEWPORT[b]	A_T A_D **CV**	E_C**ACV**
SELECT CLIPPING CONTROL	A_T A_D **AK** selector 0: Viewport boundary 1: Page boundary	E_C**AAK1**
SELECT COORDINATE TYPE	A_T A_D **BO** coordinate-type-selector 0: absolute 1: relative	E_C**ABO1**
SELECT DEVICE UNITS	A_T A_D **BV** device-units-selector 0: ADU 1: GDU 2: mm	E_C**ABV1**
SELECT GRAPHIC UNITS	A_T A_D **BW** input-selector[,output-selector] 0: World Units 1: Device Units (single selector: sets both to same)	E_C**ABW1,1** or E_C**ABW1**
SELECT LINE TYPE	A_T A_D **BL** line-type-selector 0: solid 2: variable dash 1: fixed dash 3: end points only	E_C**ABL2**
SELECT PEN	A_T A_D **BP** pen-selector 1: Pen 1 0: crosshair cursor (temporarily) 2: Pen 2 3-255: no action	E_C**ABP2** or E_C**ABP0**
SET ARC SMOOTHNESS	A_T A_D **BA** smoothness (0 to 1, 0 is very smooth, .5 default)	E_C**ABA0.2**
SET DASH PATTERN	A_T A_D **BD** pattern-element[,pattern-element]. . . up to 20 pattern elements: 1st, 3rd, etc. elements: "Draw" 2nd, 4th, etc. elements: "Move" 0: "zero Move" if 1st element, "point" if 3rd, 5th, etc., "zero Move" if 2nd, 4th, etc.	E_C**ABD2,3** or E_C**ABD0,5,10** or E_C**ABD0,0,0,5,10**
SET DASH PATTERN LENGTH	A_T A_D **BS** pattern-length	E_C**ABS100**
SET VIEWPORT	A_T A_D **AV** llx,urx,lly,ury	E_C**AAV1000,3000,500,2500**
SET WINDOW	A_T A_D **AW** llx,urx,lly,ury	E_C**AAW0,1,0,1**

[a] One output message block produced.
[b] Two output message blocks produced.
[c] One output message block per macro.

(continued)

Appendix B

ASCII CODE CHART

NOTE

Two other code charts are presented in this manual: an ASCII encoding chart in the Serial Interface section (Section 4), and a GPIB code chart in the GPIB Interface section (Section 5). These charts contain additional, interface-dependent information.

ASCII CODE CHART

B7 B6 B5	Ø Ø Ø	Ø Ø 1	Ø 1 Ø	Ø 1 1	1 Ø Ø	1 Ø 1	1 1 Ø	1 1 1
BITS B4 B3 B2 B1	CONTROL		NUMBERS SYMBOLS		UPPERCASE SYMBOLS		LOWERCASE SYMBOLS	
Ø Ø Ø Ø	0 NU NUL 0 0	20 DL DLE 10 16	40 Sp 20 32	60 0 30 48	100 @ 40 64	120 P 50 80	140 ` 60 96	160 p 70 112
Ø Ø Ø 1	1 SH SOH 1 1	21 D1 DC1 11 17	41 ! 21 33	61 1 31 49	101 A 41 65	121 Q 51 81	141 a 61 97	161 q 71 113
Ø Ø 1 Ø	2 SX STX 2 2	22 D2 DC2 12 18	42 " 22 34	62 2 32 50	102 B 42 66	122 R 52 82	142 b 62 98	162 r 72 114
Ø Ø 1 1	3 EX ETX 3 3	23 D3 DC3 13 19	43 # 23 35	63 3 33 51	103 C 43 67	123 S 53 83	143 c 63 99	163 s 73 115
Ø 1 Ø Ø	4 ET EOT 4 4	24 D4 DC4 14 20	44 $ 24 36	64 4 34 52	104 D 44 68	124 T 54 84	144 d 64 100	164 t 74 116
Ø 1 Ø 1	5 EQ ENQ 5 5	25 NK NAK 15 21	45 % 25 37	65 5 35 53	105 E 45 69	125 U 55 85	145 e 65 101	165 u 75 117
Ø 1 1 Ø	6 AK ACK 6 6	26 SY SYN 16 22	46 & 26 38	66 6 36 54	106 F 46 70	126 V 56 86	146 f 66 102	166 v 76 118
Ø 1 1 1	7 BL BEL 7 7	27 EB ETB 17 23	47 ' 27 39	67 7 37 55	107 G 47 71	127 W 57 87	147 g 67 103	167 w 77 119
1 Ø Ø Ø	10 BS 8 8	30 CN CAN 18 24	50 (28 40	70 8 38 56	110 H 48 72	130 X 58 88	150 h 68 104	170 x 78 120
1 Ø Ø 1	11 HT 9 9	31 EM 19 25	51) 29 41	71 9 39 57	111 I 49 73	131 Y 59 89	151 i 69 105	171 y 79 121
1 Ø 1 Ø	12 LF A 10	32 SB SUB 1A 26	52 * 2A 42	72 : 3A 58	112 J 4A 74	132 Z 5A 90	152 j 6A 106	172 z 7A 122
1 Ø 1 1	13 VT B 11	33 EC ESC 1B 27	53 + 2B 43	73 ; 3B 59	113 K 4B 75	133 [5B 91	153 k 6B 107	173 { 7B 123
1 1 Ø Ø	14 FF C 12	34 FS 1C 28	54 , 2C 44	74 < 3C 60	114 L 4C 76	134 \ 5C 92	154 l 6C 108	174 |* 7C 124
1 1 Ø 1	15 CR D 13	35 GS 1D 29	55 – 2D 45	75 = 3D 61	115 M 4D 77	135] 5D 93	155 m 6D 109	175 } 7D 125
1 1 1 Ø	16 SO E 14	36 RS 1E 30	56 . 2E 46	76 > 3E 62	116 N 4E 78	136 ^ 5E 4	156 n 6E 110	176 ~ 7E 126
1 1 1 1	17 SI F 15	37 US 1F 31	57 / 2F 47	77 UNL ? 3F 63	117 O 4F 79	137 UNT — 5F 95	157 o 6F 111	177 DT DEL RUBOUT 7F 127

* ¦ on some keyboards or systems

KEY

octal	25	
	NK	graphic representation
hex	15 NAK 21	decimal
	mnemonic	

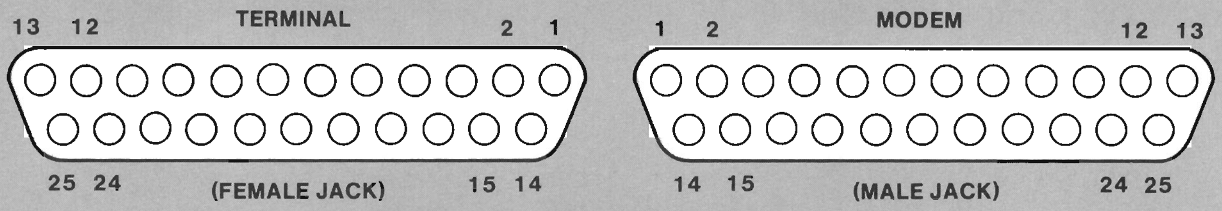
3806-66

Figure B-1. ASCII Code Chart.

Appendix C

SERIAL INTERFACE CONNECTOR

Refer to Figure C-1 for the pin/signal name relationships of the RS-232-C Interface connector.



PIN/SIGNAL DESCRIPTIONS

PIN NUMBER (EITHER JACK)	DESCRIPTION
1	PROTECTIVE GROUND (SHIELD)
2	TRANSMITTED DATA
3	RECEIVED DATA
4	REQUEST TO SEND
5	CLEAR TO SEND
6	DATA SET READY
7	SIGNAL GROUND (COMMON RETURN)
8	RECEIVED LINE SIGNAL DETECTOR (DATA CARRIER DETECT)
12	SECONDARY RECEIVED LINE SIGNAL DETECTOR
13	SECONDARY CLEAR TO SEND
14	SECONDARY TRANSMITTED DATA
15	TRANSMISSION SIGNAL ELEMENT TIMING (EXTERNAL TRANSMIT CLOCK)
16	SECONDARY RECEIVED DATA
17	RECEIVED SIGNAL ELEMENT TIMING (EXTERNAL RECEIVE CLOCK)
19	SECONDARY REQUEST TO SEND
20	DATA TERMINAL READY
21	SIGNAL QUALITY DETECTOR
22	RING INDICATOR
23	DATA SIGNAL RATE SELECTOR
24	TRANSMIT SIGNAL ELEMENT TIMING

ALL OTHER PINS ARE UNASSIGNED

3806-61

Figure C-1. RS-232-C Interface Connector.

Command	Format	Example
Digitizing Commands		
CALL DIGITIZE[a]	**C**	**PRINT@1,32: “C”** **INPUT@1,32: V1,V2,T**
DIGITIZE[a]	**G**	**PRINT@1,32: “G”** **INPUT@1,32: V1,V2,T**
JOYSTICK AXIS DISABLE	**BJ** axis-selector 1: x 0: none 2: y 3: both	**PRINT@1,32: “BJ”; 2**
OPERATOR DIGITIZE ENABLE	**AG** number-of-points	**PRINT@1,32: “AG”; 20**
PROMPT LIGHT OFF	**T 0**	**PRINT@1,32: “T0”**
PROMPT LIGHT ON	**T 1**	**PRINT@1,32: “T1”**
Macro Commands		
BEGIN MACRO DEFINITION	**BB** macro-number	**PRINT@1,32: “BB”; 12**
DELETE MACRO	**BK** macro-number[,macro-number]. . . no argument: all macros	**PRINT@1,32: “BK”; 7** **PRINT@1,32: “BK”; 3;5;18** or **PRINT@1,32: “BK”**
END MACRO DEFINITION	**BE**	**PRINT@1,32: “BE”**
EXPAND MACRO	**AE** macro-number[,macro-number]. . .	**PRINT@1,32: “AE”; 12** or **PRINT@1,32: “AE”; 3;9;11**
READ MACRO STATUS[c]	**CS**	**PRINT@1,32: “CS”** **INPUT@1,32: V1,V2,T1** **INPUT@1,32: V3,V4,T2** **INPUT@1,32: V5,V6,T3**
SET AUTO MACRO	**BN** macro-number no argument: auto macro “off”	**PRINT@1,32: “BN”; 12** or **PRINT@1,32: “BN”**
Downloadable Character Commands		
BEGIN CHARACTER DEFINITION	**CN** font-selector,single-character-string	**PRINT@1,32: “CN”; 4;“,’a’”**
DELETE CHARACTER DEFINITION	**CP** font-selector,character-string	**PRINT@1,32: “CP”; 1;“,’a’”** or **PRINT@1,32: “CP”; 1;** **’,’abc’”**
DELETE FONT DEFINITION	**CQ** [font-selector[,font-selector]. . .] no argument: all fonts	**PRINT@1,32: “CQ”; 2** or **PRINT@1,32: “CQ”; 2;4;5** or **PRINT@1,32: “CQ”**
END CHARACTER DEFINITION	**CO**	**PRINT@1,32: “CO”**
SELECT NONADVANCING CHAR.	**CU**	**PRINT@1,32: “CU”**
SET CHARACTER X-EXTENT	**CT** left,right	**PRINT@1,32: “CN”; 0;6**
SET DOWNLOADED CHAR. SIZE	**CZ** font-selector,x-min,x-max,y-min,y-max	**PRINT@1,32: “CZ”; 1;0;6;** **0;11**
Media Advance Commands		
ADVANCE MEDIA	**AU** increments	**PRINT@1,32: “AU”; 128**
PAGE CHANGE	**BC**	**PRINT@1,32: “BC”**
READ FORM LENGTH[a]	**CF**	**PRINT@1,32: “CF”** **INPUT@1,32: V1,V2,T**
SET FORM LENGTH	**BF** formlength	**PRINT@1,32: “BF”; 768**

[a] One output message block produced.
[b] Two output message blocks produced.
[c] One output message block per macro.

Appendix D

COMMAND SUMMARY (GPIB)

The general GPIB form is shown under the Format column while the 4050 BASIC (Data Byte) form is shown under the Examples column. The Secondary Address and Special Keyword forms of 4050 BASIC are not shown.

Command	Format	Example
Interface Commands		
DATA RESET	**CD**	**PRINT@1,32. “CD”**
INTERFACE PARAMETER RESET	**CR**	**PRINT@1,32: “CR”**
SELECT COMMAND/RESPONSE FMT	**CC** format-selector	**PRINT@1,32: “CC”; 2**
Device Commands		
DEVICE RESET	**N** [selector] no argument: same as 0	**PRINT@1,32: “N”; 0** or **PRINT@1,32: “N”**
IDENTIFY[a]	**I**	**PRINT@1,32: “I”** **INPUT@1,32: V1,V2,T**
READ ERROR[b]	**CE**	**PRINT@1,32: “CE”** **INPUT@1,32: V1,V2,T1** **INPUT@1,32: V3,V4,T2**
READ STATUS[a]	**V** status-register-selector	**PRINT@1,32: “V”; 2** **INPUT@1,32: V1,V2,T**
Alpha Commands		
CHARACTER MOVE	**AM** character-spaces,line-spaces	**PRINT@1,32: “AM”; 2;-3**
MOVE TO HOME	**H**	**PRINT@1,32: “H”**
PRINT	**P** print-string	**PRINT@1,32: “PAbcd”**
PRINT CENTERED	**AP** print-string	**PRINT@1,32: “APX”**
RESET ALPHA PARAMETERS	**A**	**PRINT@1,32: “A”**
SELECT ALPHA SPACING CONTROL	**AJ** spacing-control-selector 0: constant origin-to-origin 1: constant space between chars	**PRINT@1,32: “AJ”; 1**
SELECT ALTERNATE ALPHA FONT	**BQ** alternate-font-selector	**PRINT@1,32: “BQ”; 3**
SELECT STANDARD ALPHA FONT	**F** standard-font-selector	**PRINT@1,32: “F”; 2**
SET ALPHA DIMENSION	**S** character-space,line-space	**PRINT@1,32: “S”; 90;180**
SET ALPHA MARGIN SEPARATION	**BR** margin-separation	**PRINT@1,32: “BR”; 79.7**
SET ALPHA RATIO	**BI** x-ratio,y-ratio	**PRINT@1,32: “BI”; 1.5;1.63636**
SET ALPHA ROTATION	**R** rotation-angle	**PRINT@1,32: “R”; 45**
SET ALPHA SCALE	**BH** x-mag-factor,y-mag-factor	**PRINT@1,32: “BH”; 2;1**
SET ALPHA SIZE	**BZ** character-width,character-height	**PRINT@1,32: “BZ”; 60;110**
SET ALPHA SLANT	**BG** slant-angle	**PRINT@1,32: “BG”; 20**
SET TAB SEPARATION	**BT** tab-separation	**PRINT@1,32: “BT”; 6.5**
Graphics Commands		
AXIS	**CA** x-space,y-space[,x-intercept,y-intercept]	**PRINT@1,32: “CA”; 5;10; 50;50**
DRAW	**D** x-value,y-value[,x-value,y-value]. . .	**PRINT@1,32: “D”; 2;40** or **PRINT@1,32: “D”; 2;40;7;9**
DRAW ARC	**AA** x-mid,y-mid,x-end,y-end[,x-mid,y-mid,x-end, y-end]. . .	**PRINT@1,32: “AA”; 20;30; 49;62**
DRAW CIRCLE	**AC** radius[,radius].	**PRINT@1,32: “AC”; 256** or **PRINT@1,32: “AC”; 5;10; 20**
MARK VIEWPORT	**CM**	**PRINT@1,32: “CM”**
MOVE	**M** x-value,y-value	**PRINT@1,32: “M”; 32;40**

[a] One output message block produced.
[b] Two output message blocks produced.
[c] One output message block per macro.

Command	Format	Example
Graphics Commands (cont)		
MOVE TO LOAD POINT	**AI**	**PRINT@1,32: “AI”**
OUTLINE VIEWPORT	**CB**	**PRINT@1,32: “CB”**
READ VIEWPORT[b]	**CV**	**PRINT@1,32: “CV”** **INPUT@1,32: V1,V2,T1** **INPUT@1,32: V3,V4,T2**
SELECT CLIPPING CONTROL	**AK** selector 0: viewport boundary 1: page boundary	**PRINT@1,32: “AK”; 1**
SELECT COORDINATE TYPE	**BO** coordinate-type-selector 0: absolute 1: relative	**PRINT@1,32: “BO”; 1**
SELECT DEVICE UNITS	**BV** device-units-selector 0: ADU 1: GDU 2: mm	**PRINT@1,32: “BV”; 1**
SELECT GRAPHIC UNITS	**BW** input-selector[,output-selector] 0: world 1: device single argument: use for input and output	**PRINT@1,32: “BW”; 1;1** or **PRINT@1,32: “BW”; 1**
SELECT LINE TYPE	**BL** line-type-selector 0: solid 2: variable dash 1: fixed dash 3: end points only	**PRINT@1,32: “BL”; 2**
SELECT PEN	**BP** pen-selector 1: Pen 1 0: quick time-out 2: Pen 2 3-255: no change in pen selection	**PRINT@1,32: “BP”; 2**
SET ARC SMOOTHNESS	**BA** smoothness (0 to 1, 0 is very smooth, .5 default)	**PRINT@1,32: “BA”; 0.5**
SET DASH PATTERN	**BD** pattern-element[,pattern-element]. Up to 20 pattern elements 1st,3rd,etc. elements: “Draw” 2nd,4th,etc. elements: “Move” 0: “zero Move” if 1st element, “point” if 3rd,5th,etc., “zero Move” if 2nd,4th,etc.)	**PRINT@1,32: “BD”; 1;2**
SET DASH PATTERN LENGTH	**BS** pattern-length	**PRINT@1,32: “BS”; 256**
SET VIEWPORT	**AV** llx,urx,lly,ury	**PRINT@1,32: “AV”; 0;1024; 0;755.8**
SET WINDOW	**AW** llx,urx,lly,ury	**PRINT@1,32: “AW”; 0;360; −1;1**
Transformation Commands		
RESTORE PREVIOUS TRANSFORM	**AY**	**PRINT@1,32: “AY”**
SAVE CURRENT TRANSFORM	**AX**	**PRINT@1,32: “AX”**
SET ORIGIN TO CURRENT POS	**AO**	**PRINT@1,32: “AO”**
SET ROTATION	**AR** angle	**PRINT@1,32: “AR”; 30**
SET ROTATION TO LAST ANGLE	**AL**	**PRINT@1,32: “AL”**
SET SCALE	**AS** x-scale,y-scale	**PRINT@1,32: “AS”; 2;4**
SET SKEW	**AQ** x-skew-angle,y-skew-angle	**PRINT@1,32: “AQ”; 20;10**
SET TRANSLATION	**AT** x-translation,y-translation	**PRINT@1,32: “AT”; 512; 1024**

[a] One output message block produced.
[b] Two output message blocks produced.
[c] One output message block per macro.

(continued)

Appendix E

GPIB DESCRIPTION

THE GENERAL PURPOSE INTERFACE BUS (GPIB)

The GPIB is a standard interface for programmable instrumentation defined in IEEE Standard 488-1978. The following descriptions summarize the parts of that standard that apply to the Plotter.

THE GPIB CONNECTOR

The GPIB connector (located on the Plotter's rear panel) permits Plotter access to GPIB devices. These devices must conform to IEEE Standard 488-1978 and describes a byte-serial, bit-parallel interface system for programmable instruments. The GPIB connector is a standard 24-pin connector with sixteen active signal lines and eight interlaced grounds. The cable attached to the GPIB connector must be no longer than 65.6 ft (20 m) with no more than fifteen peripheral devices connected at one time. Refer to Figure E-1 for the connector pin arrangement and the respective signal line nomenclature.

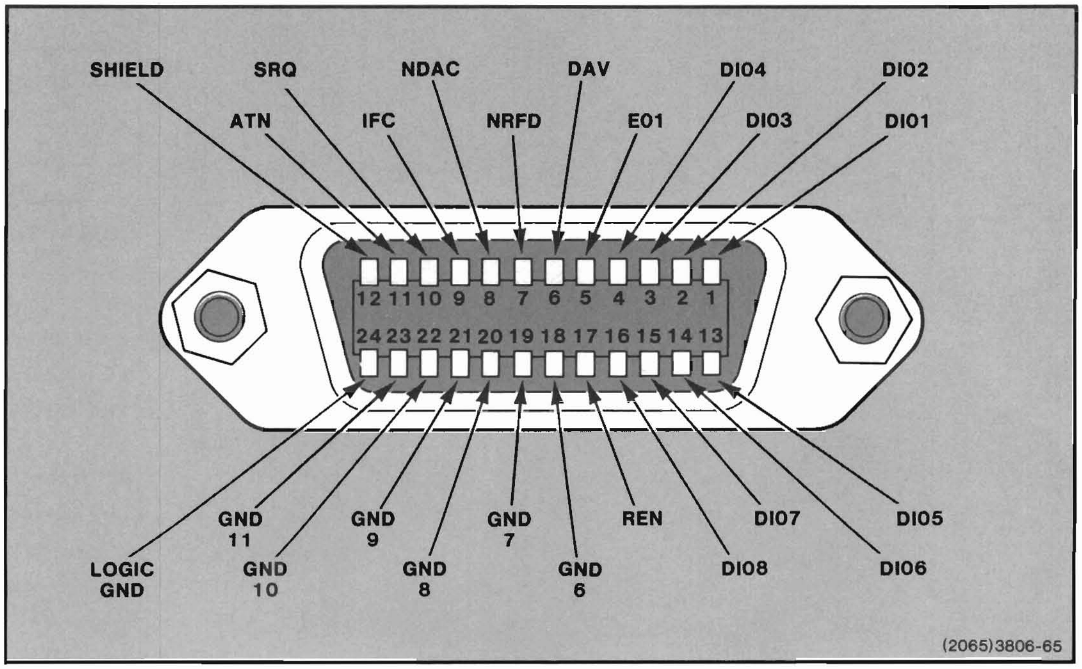


Figure E-1. Connector Pin Arrangement.

THE GPIB INTERFACING CONCEPT

The GPIB is functionally divided into three component busses: an eight-line Data bus, a three-line Transfer bus, and a five-line Management bus for a total of sixteen active signal lines. Figure E-2 shows the bus structure.

The transfer rate over the Data bus is a function of the slowest peripheral device taking part in a transfer at any one time. The bus operates asynchronously with a maximum transfer rate of 250K bytes/second (one megabyte/second with tri-state drivers). Both peripheral addresses and data are sent sequentially over the Data Bus. Once peripheral addresses are established for a particular transfer, successive data bytes can be transmitted in a group for higher effective data rates.

Peripheral devices on the GPIB are designated as talkers or listeners either manually by the Plotter's front panel functions, or under program control by a controller device on the GPIB bus.

A *talker* is a device capable of transmitting information on the Data bus. Only one talker can exist at a time.

A *listener* is a device capable of receiving information transmitted over the Data bus. A maximum of fourteen listeners can receive directions in an I/O (input/output) operation at any one time.

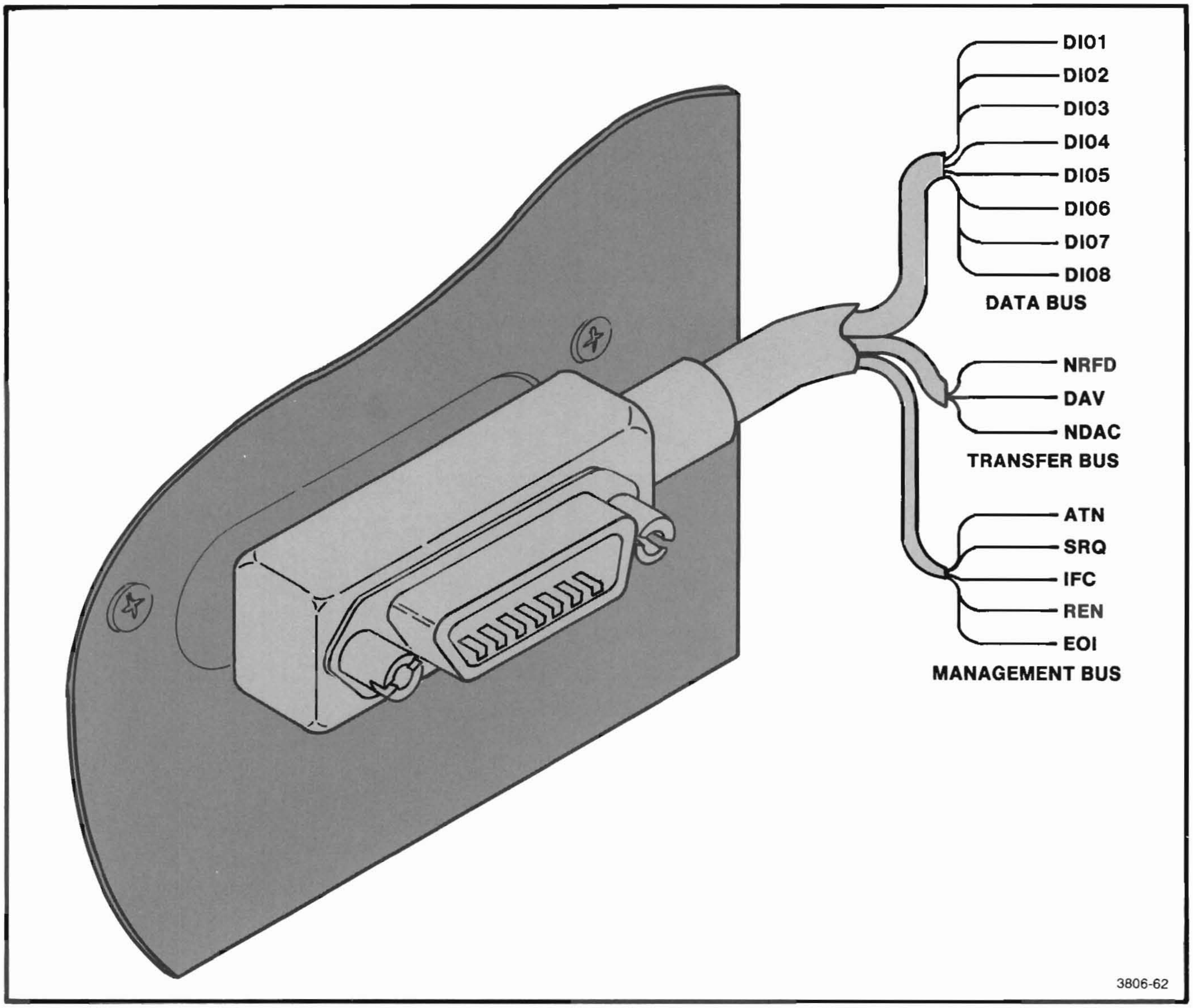


Figure E-2. Bus Structure.

GPIB SIGNAL DEFINITIONS

The GPIB signal lines are described in three groups in the following discussions.

DATA BUS

The Data bus contains eight bidirectional active-low signal lines, DIO1 through DIO8. One byte of information (eight bits) is transferred over the bus at a time. DIO1 represents the least significant bit in the byte; DIO8 represents the most significant bit in the byte.

Each byte represents a peripheral address (either primary or secondary), a control message, or a data byte. Data bytes sent to the Plotter are formatted in ASCII code.

MANAGEMENT BUS

The Management bus is a group of five signal lines that control data transfers over the Data bus. The five signals and their definitions appear in Table E-1.

Table E-1

MANAGEMENT BUS SIGNALS

Signal Line	Definition
Attention (ATN)	This signal line is asserted by the controller while peripheral devices are assigned as listeners and talkers. Only peripheral addresses and control messages are transferred over the Data bus when ATN is active low. Once ATN goes high, only peripheral devices assigned as listeners and talkers can take part in the transfer of data bytes. The bus controller is the only source of this signál.
Service Request (SRQ)	Any peripheral device on the GPIB can request the attention of the controller by asserting SRQ as active low. The controller typically responds by asserting ATN active low and executing a serial poll to see which device is requesting service. If the controller is a 4050 Series Graphic Computer System, this response is enabled by an "ON SRQ THEN" statement, which is executed in the BASIC program. The serial poll is taken when a "POLL" statement is executed in the BASIC program. After the peripheral device requesting service is found, BASIC program control is transferred to a service routine for that device. When the service routine is finished executing, program control returns to the main program. The SRQ signal line is reset to an inactive state when the device requesting service is polled.
Interface Clear (IFC)	The IFC signal line is asserted by the controller when all interface circuitry must be placed in a predetermined quiescent state. The bus controller is the only source of this signal. In a 4050 Series Graphic Computer System, IFC is activated each time the "INIT" statement is executed in a BASIC program or the 4050 BREAK key is hit twice in succession.
Remote Enable (REN)	The REN signal line is asserted whenever the system is operating under program control. REN causes all peripheral devices on GPIB to ignore their front panel controls and operate under remote control by signals and/or control messages received over the GPIB. The 4663 Plotter has no REN capability.
End or Identify (EOI)	The EOI signal is used by the talker to indicate the end of a data transfer sequence. The talker activates EOI as the last byte of transmitted data. A listening bus controller (with EOI asserted) assumes a data byte being received is the last byte in the transmission. When the 4050 Series Graphic Computer System is talking, it always asserts EOI along with the last byte transferred.

THE TRANSFER BUS

A handshake sequence is executed by the talker and the listeners over the Transfer bus each time a data byte is transferred over the Data bus. Signal lines and definitions for the Transfer bus appear in Table E-2.

Table E-2

TRANSFER BUS SIGNALS

Signal Line	Definition
Not Ready For Data (NRFD)	An active low NRFD signal line indicates that one or more assigned listeners are not ready to receive the next data byte. When all assigned listeners for a particular data transfer have released NRFD, the NRFD line goes inactive high. This tells the talker to place the next data byte on the Data bus.
Data Valid (DAV)	The DAV signal line is asserted by the talker shortly after a valid data byte is placed on the Data bus. An active low DAV signal tells each listener to capture the data byte currently on the Data bus. The talker is inhibited from asserting DAV when NRFD is active low.
No Data Accepted (NDAC)	The NDAC signal line is held active low by each listener until that listener captures the data byte currently being transmitted over the Data bus. Once all listeners have captured the data byte, NDAC goes inactive high. This tells the talker to take the byte off the Data bus.

GPIB DATA FORMATS

Any series of 8-bit bytes can be transmitted over the GPIB. For the Plotter, both numeric data and alphanumeric data are transmitted in ASCII code.

TRANSFERRING ASCII DATA

ASCII numeric data can be transferred in either integer, floating point format, or scientific notation format, as long as the most significant digit is transferred first. Valid ASCII numeric characters are digits 0 through 9, E, e, + , —, and the decimal point. ASCII character strings can be transmitted as any sequence of valid ASCII characters. All ASCII data transfers, both numeric and alphanumeric, are terminated with a carriage return character and/or by activating the EOI signal line on the Management bus.

When using a TEKTRONIX 4050 Series Graphic System, ASCII data is transferred from the Graphic System to a Plotter or other peripheral device on the GPIB using the PRINT statement. ASCII transfers in the opposite direction are executed by using the INPUT statement. (Refer to the INPUT and PRINT statements in the Input/Output Operations section of the *4051 Graphic System Reference Manual* for detailed information on ASCII data transfers over the GPIB.)

TRANSFERRING ONE DATA BYTE AT A TIME

When using a TEKTRONIX 4050 Series Graphic System, direct access to the GPIB is available through the "WBYTE" (Write Byte) statement and the "RBYTE" (Read Byte) statement. These two statements permit sending any eight-bit bytes over the GPIB. Also, the "WBYTE" statement can assert the ATN signal line (telling peripheral devices that the byte you are sending is a peripheral address or a control message) and gives you complete control over the activation of the EOI signal line (except when a binary 0 is transferred). Refer to the WBYTE and RBYTE statements in the I/O Operations section of the *4051 Graphic System Reference Manual.*

GPIB TO IEEE COMPATIBILITY

INTRODUCTION

The following text describes the interfacing compatibility of the Plotter's GPIB interface with the IEEE Standard 488-1978. The Plotter can function as a standard talker or listener.

4663 GPIB SUBSET

The IEEE Standard 488-1978 requires each interface to contain a group of allowable GPIB function subsets. Table E-3 provides a list of subsets which the Plotter's GPIB interface implements and gives a description of the GPIB interface capability. If needed, refer to the IEEE Standard 488-1978 for a detailed description of these subsets.

Table E-3

GPIB INTERFACE SUBSETS FOR THE PLOTTER

Function	Implemented Subset	Comments
SH (Source Handshake)	SH1	Complete Capability
AH (Acceptor Handshake)	AH1	Complete Capability
T (Talker)	T5	Serial Poll, Talk Only, Unaddress if MLA
T (With Address Extension)	TEO	No Capability
L (Listener)	L3	Listen Only, Unaddress if MTA
L (With Address Extension)	LEO	No Capability
SR (Service Request)	SR1	Complete Capability
RL (Remote/Local)	RLO	No Capability
PP (Parallel Poll)	PPO	No Capability
DC (Device Clear)	DC1	Complete Capability
DT (Device Trigger)	DTO	No Capability
C (Controller)	CO	No Capability

NOTE

Secondary addresses can be used to send commands to the Plotter. The use of secondary addresses as commands is not the same as the use of secondary addresses for extended addressing functions. The Plotter accepts secondary addresses as commands to maintain compatiblity with the TEKTRONIX 4050 Series Graphic Computer Systems.

EXCEPTIONS TO GPIB STANDARD

The Plotter conforms to the electrical, mechanical, and procedural standards contained in IEEE Standard 488-1978 with the following exceptions:

- The Plotter might not (if it is busy processing commands) recognize an IFC (Interface Character) bus message if the message is asserted for the minimum time specified (100 μsec). The message should be asserted for at least 300 μsec to ensure a response under all conditions.
- The Plotter will not respond to an IFC bus message within the maximum time specified in the IEEE Standard (100 μsec). The actual response time is 165 μsec and the controller should delay about 200 μsec following an IFC bus message before using the GPIB.
- Excessive noise on the NRFD line of the GPIB can cause the Plotter (when addressed as a talker) to erroneously transmit data bytes when the listener is not ready to receive.
- If the Plotter is addressed as a listener by being placed in the Listen Only mode, the listen function is not inhibited while the IFC bus message is asserted. This permits response to bus messages other than DCL (Device Clear) while IFC is asserted.

Appendix F

COORDINATE CONVERSION CHART

INTRODUCTION

Tables F-1 through F-4 are designed for use with 12-, 14-, 16-, and 10-bit graphing, respectively. Use the appropriate procedure described for your desired coordinate resolution.

10-BIT GRAPHING

10-bit graphing is the appropriate resolution for the TEKTRONIX 4010, 4012, or 4013 Terminals. Simply use Table F-4 (Coordinate Conversion Chart). Find the desired X or Y coordinate in the body of the chart; follow that column to the bottom of the chart to find the decimal value or the ASCII character that represents the HIY or HIX byte; go to the far right in the row containing the coordinate value to find the LOY byte, or go to the left to find the LOX byte. With 10-bit graphing, it is not possible to use coordinate values between those shown on the chart. Arrange the characters in the sequence:

HIY,LOY,HIX,LOX

Example: 480X,100Y would be SP,y,#,X in ASCII code.

12-BIT GRAPHING

12-bit graphing is the appropriate resolution for the TEKTRONIX 4014 or 4015 Terminals. You must use Table F-1 (Interpolation Insert Chart) and Table F-4 (Coordinate Conversion Chart).

Simply find the largest coordinate value in Table F-4 that is equal to or less than the desired coordinate value; follow that column to the bottom of the chart to find the decimal value or the ASCII character that represents the HIY or HIX byte; go to the right in the row containing that largest coordinate value equal to or less than the desired coordinate value to find the LOY byte, or go to the left to find the LOX byte. Then subtract this largest coordinate value in the chart that is equal to or less than the desired coordinate value. Repeat for the other coordinate value. Note both of these remainders and their respective axes (for the example below, X remainder = 1, Y remainder = 3). Notice that these values will always range from 0 to 3. Apply both of these remainders to Table F-1 (using the X remainder across the top and the Y remainder down the left side) to determine the extra byte (EB) character in the sequence below. (Figure F-1 shows the increased resolution gained by the extra byte.)

HIY,EB,LOY,HIX,LOX

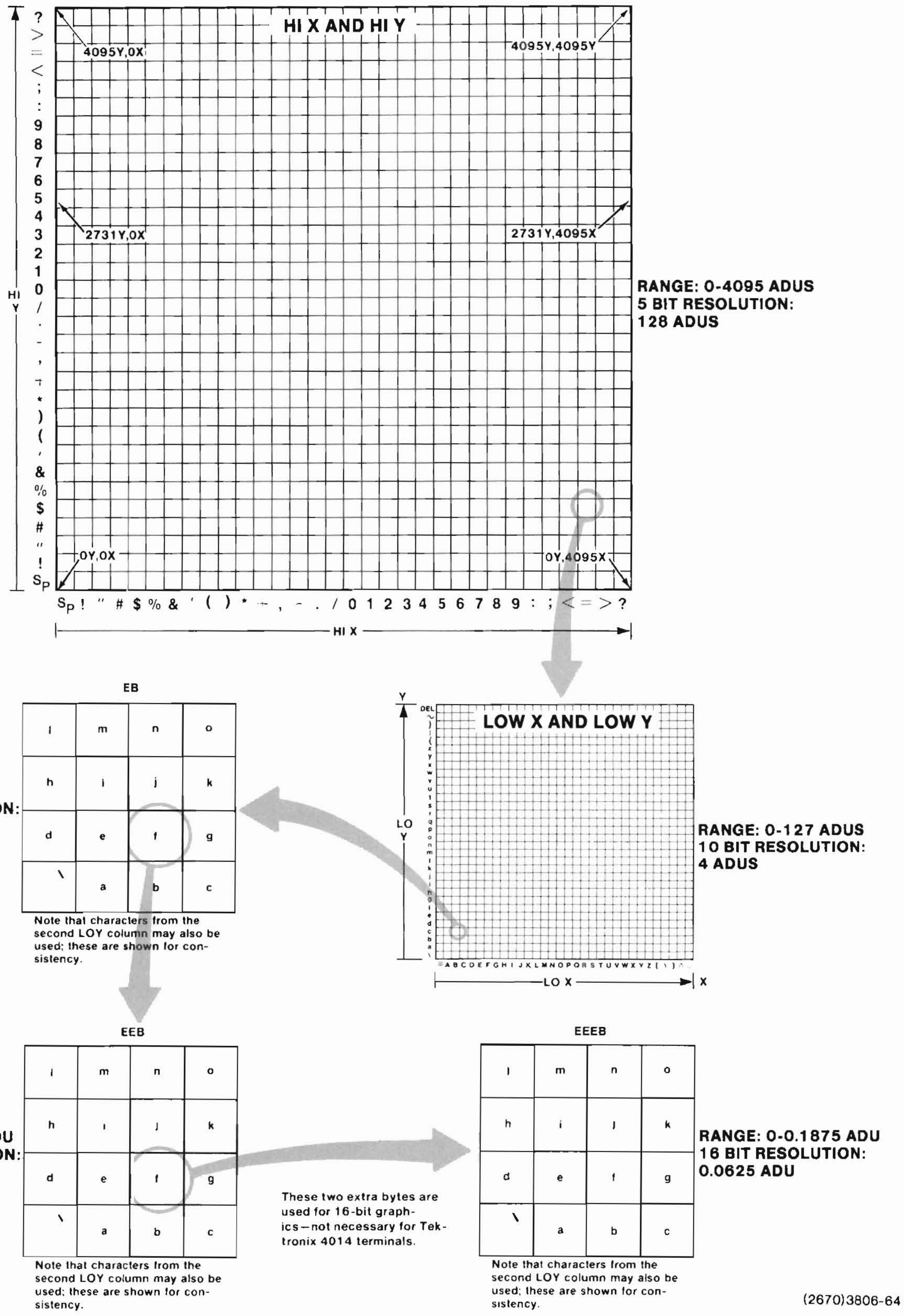
Example: 841X,31Y would be SP,m,g,&,R in ASCII code.

Table F-1

INTERPOLATION INSERT CHART (12-BIT RESOLUTION USING THE EB BYTE)

Y Remainder	X Remainder			
	0	1	2	3
3	l	m	n	o
2	h	i	j	k
1	d	e	f	g
0	`	a	b	c

COORDINATE CONVERSION CHART



RANGE: 0-4095 ADUS
5 BIT RESOLUTION:
128 ADUS

RANGE: 0-3 ADUS
12 BIT RESOLUTION:
1 ADU

Note that characters from the second LOY column may also be used; these are shown for consistency.

RANGE: 0-127 ADUS
10 BIT RESOLUTION:
4 ADUS

RANGE: 0-0.75 ADU
14 BIT RESOLUTION:
0.25 ADU

Note that characters from the second LOY column may also be used; these are shown for consistency.

These two extra bytes are used for 16-bit graphics—not necessary for Tektronix 4014 terminals.

RANGE: 0-0.1875 ADU
16 BIT RESOLUTION:
0.0625 ADU

Note that characters from the second LOY column may also be used; these are shown for consistency.

(2670)3806-64

Figure F-1. X-Y Coordinates (Showing the Increased Resolution Gained by the Extra Byte).

14-BIT AND 16-BIT GRAPHING

For graphics coordinates with additional resolution (14-bit or 16-bit), repeat the 12-bit graphing process previously described to determine the EEB or EEEB bytes by obtaining a remainder using Tables F-2 or F-3. For 16-bit resolution, arrange the characters in the following sequence:

HIY,EEEB,EEB,EB,LOY,HIX,LOX

Examples: 0X,0Y would be $^{S}_{P}$````$^{S}_{P}$@ in ASCII code.

480.375X,100.875Y would be $^{S}_{P}$jm`y#x in ASCII code.

Table F-2

INTERPOLATION INSERT CHART (14-BIT RESOLUTION USING THE EEB BYTE)

Y Remainder	X Remainder			
	0	0.25	0.50	0.75
0.75	l	m	n	o
0.50	h	i	j	k
0.25	d	e	f	g
0	`	a	b	c

Table F-3

INTERPOLATION INSERT CHART (16-BIT RESOLUTION USING THE EEEB BYTE)

Y Remainder	X Remainder			
	0	0.0625	0.125	0.1875
0.1875	l	m	n	o
0.125	h	i	j	k
0.0625	d	e	f	g
0	`	a	b	c

Refer to the introduction to commands and responses in Section 4 or 5 for further information.

Table F-4
COORDINATE CONVERSION CHART

Low Order X										Low Order Y	
ASCII	DEC.	X or Y Coordinate								DEC.	ASCII
@	64	0	128	256	384	512	640	768	896	96	`
A	65	4	132	260	388	516	644	772	900	97	a
B	66	8	136	264	392	520	648	776	904	98	b
C	67	12	140	268	396	524	652	780	908	99	c
D	68	16	144	272	400	528	656	784	912	100	d
E	69	20	148	276	404	532	660	788	916	101	e
F	70	24	152	280	408	536	664	792	920	102	f
G	71	28	156	284	412	540	668	796	924	103	g
H	72	32	160	288	416	544	672	800	928	104	h
I	73	36	164	292	420	548	676	804	932	105	i
J	74	40	168	296	424	552	680	808	936	106	j
K	75	44	172	300	428	556	684	812	940	107	k
L	76	48	176	304	432	560	672	816	944	108	l
M	77	52	180	308	436	564	676	820	948	109	m
N	78	56	184	312	440	568	680	824	952	110	n
O	79	60	188	316	444	572	684	828	956	111	o
P	80	64	192	320	448	576	688	832	960	112	p
Q	81	68	196	324	452	580	672	836	964	113	q
R	82	72	200	328	456	584	676	840	968	114	r
S	83	76	204	332	460	588	680	844	972	115	s
T	84	80	208	336	464	592	784	848	976	116	t
U	85	84	212	340	468	596	788	852	980	117	u
V	86	88	216	344	472	600	792	856	984	118	v
W	87	92	220	348	476	604	796	860	988	119	w
X	88	96	224	352	480	608	800	864	992	120	x
Y	89	100	228	356	484	612	804	868	996	121	y
Z	90	104	232	360	488	616	808	872	1002	122	z
[	91	108	236	364	492	620	812	876	1006	123	{
\	92	112	240	368	496	624	816	880	1010	124	'
]	93	116	244	372	500	628	820	884	1012	125	}
∧	94	120	248	376	504	632	824	888	1016	126	~
_	95	124	252	380	508	636	828	892	1020	127	rubout DT
DEC. ------>		32	33	34	35	36	37	38	39		
ASCII------>		SP	!	“	#	$	%	&	`		
		High Order X & Y									

Table F-4 (cont)
COORDINATE CONVERSION CHART

Low Order X		X or Y Coordinate								Low Order Y	
ASCII	DEC.									DEC.	ASCII
@	64	1024	1152	1280	1408	1536	1664	1792	1920	96	`
A	65	1028	1156	1284	1412	1540	1668	1796	1924	97	a
B	66	1032	1160	1288	1416	1544	1672	1800	1928	98	b
C	67	1036	1164	1292	1420	1548	1676	1804	1932	99	c
D	68	1040	1168	1296	1424	1552	1680	1808	1936	100	d
E	69	1044	1172	1300	1428	1556	1684	1812	1940	101	e
F	70	1048	1176	1304	1432	1560	1688	1816	1944	102	f
G	71	1052	1180	1308	1436	1564	1692	1820	1948	103	g
H	72	1056	1184	1312	1440	1568	1696	1824	1952	104	h
I	73	1060	1188	1316	1444	1572	1700	1828	1956	105	i
J	74	1064	1192	1320	1448	1576	1704	1832	1960	106	j
K	75	1068	1196	1324	1452	1580	1708	1836	1964	107	k
L	76	1072	1200	1328	1456	1584	1712	1840	1968	108	l
M	77	1076	1204	1332	1460	1588	1716	1844	1972	109	m
N	78	1080	1208	1336	1464	1592	1720	1848	1976	110	n
O	79	1084	1212	1340	1468	1596	1724	1852	1980	111	o
P	80	1088	1216	1344	1472	1600	1728	1856	1984	112	p
Q	81	1092	1220	1348	1476	1604	1732	1860	1988	113	q
R	82	1096	1224	1352	1480	1608	1736	1864	1992	114	r
S	83	1100	1228	1356	1484	1612	1740	1868	1996	115	s
T	84	1104	1232	1360	1488	1616	1744	1872	1200	116	t
U	85	1108	1236	1364	1492	1620	1748	1876	2004	117	u
V	86	1112	1240	1368	1496	1624	1752	1880	2008	118	v
W	87	1116	1244	1372	1500	1628	1756	1884	2012	119	w
X	88	1120	1248	1376	1504	1632	1760	1888	2016	120	x
Y	89	1124	1252	1380	1508	1636	1764	1892	2020	121	y
Z	90	1128	1256	1384	1512	1640	1768	1896	2024	122	z
[	91	1132	1260	1388	1516	1644	1772	1900	2028	123	{
\	92	1136	1264	1392	1520	1648	1776	1904	2032	124	¦
]	93	1140	1268	1396	1524	1652	1780	1908	2036	125	}
∧	94	1144	1272	1400	1528	1656	1784	1912	2040	126	~
_	95	1148	1276	1404	1532	1660	1788	1916	2044	127	rubout D_T
DEC. ------>		40	41	42	43	44	45	46	47		
ASCII------>		(	)	*	+	`	-	.	/		
		High Order X & Y									

Table F-4 (cont)

COORDINATE CONVERSION CHART

Low Order X										Low Order Y	
ASCII	DEC.				X or Y Coordinate					DEC.	ASCII
@	64	2048	2176	2304	2432	2560	2688	2816	2944	96	`
A	65	2052	2180	2308	2436	2564	2692	2820	2948	97	a
B	66	2056	2184	2312	2440	2568	2696	2824	2952	98	b
C	67	2060	2188	2316	2444	2572	2700	2828	2956	99	c
D	68	2064	2192	2320	2448	2576	2704	2832	2960	100	d
E	69	2068	2196	2324	2452	2580	2708	2836	2964	101	e
F	70	2072	2200	2328	2456	2584	2712	2840	2968	102	f
G	71	2076	2204	2332	2460	2588	2716	2844	2972	103	g
H	72	2080	2208	2336	2464	2592	2720	2848	2976	104	h
I	73	2084	2212	2340	2468	2596	2724	2852	2980	105	i
J	74	2088	2216	2344	2472	2600	2728	2856	2984	106	j
K	75	2092	2220	2348	2476	2604	2732	2860	2988	107	k
L	76	2096	2224	2352	2480	2608	2736	2864	2992	108	l
M	77	2100	2228	2356	2484	2612	2740	2868	2996	109	m
N	78	2104	2232	2360	2488	2616	2744	2872	3000	110	n
O	79	2108	2236	2364	2492	2620	2748	2876	3004	111	o
P	80	2112	2240	2368	2496	2624	2752	2880	3008	112	p
Q	81	2116	2244	2372	2500	2628	2756	2884	3012	113	q
R	82	2120	2248	2376	2504	2632	2760	2888	3016	114	r
S	83	2124	2252	2380	2508	2636	2764	2892	3020	115	s
T	84	2128	2256	2384	2512	2640	2768	2896	3024	116	t
U	85	2132	2260	2388	2516	2644	2772	2900	3028	117	u
V	86	2136	2264	2392	2520	2648	2776	2904	3032	118	v
W	87	2140	2268	2396	2524	2652	2780	2908	3036	119	w
X	88	2144	2272	2400	2528	2656	2784	2912	3040	120	x
Y	89	2148	2276	2404	2532	2660	2788	2916	3044	121	y
Z	90	2152	2280	2408	2536	2664	2792	2920	3048	122	z
[	91	2156	2284	2412	2540	2668	2796	2924	3052	123	{
\	92	2160	2288	2416	2544	2672	2800	2928	3056	124	¦
]	93	2164	2292	2420	2548	2676	2804	2932	3060	125	}
∧	94	2168	2296	2424	2552	2680	2812	2936	3064	126	~
_	95	2172	2300	2428	2556	2684	2816	2940	3068	127	rubout D_T
DEC. ------>		48	49	50	51	52	53	54	55		
ASCII------>		0	1	2	3	4	5	6	7		
					High Order X & Y						

Table F-4 (cont)
COORDINATE CONVERSION CHART

Low Order X										Low Order Y	
ASCII	DEC.	X or Y Coordinate								DEC.	ASCII
@	64	3072	3200	3328	3456	3584	3712	3840	3968	96	`
A	65	3076	3204	3332	3460	3588	3716	3844	3972	97	a
B	66	3080	3208	3336	3464	3592	3720	3848	3976	98	b
C	67	3084	3212	3340	3468	3596	3724	3852	3980	99	c
D	68	3088	3216	3344	3472	3600	3728	3856	3984	100	d
E	69	3092	3220	3348	3476	3604	3732	3860	3988	101	e
F	70	3096	3224	3352	3480	3608	3736	3864	3992	102	f
G	71	3100	3228	3356	3484	3612	3740	3868	3996	103	g
H	72	3104	3232	3360	3488	3616	3744	3872	4000	104	h
I	73	3108	3236	3364	3492	3620	3748	3876	4004	105	i
J	74	3112	3240	3368	3496	3624	3752	3880	4008	106	j
K	75	3116	3244	3372	3500	3628	3756	3884	4012	107	k
L	76	3120	3248	3376	3504	3632	3760	3888	4016	108	l
M	77	3124	3252	3380	3508	3636	3764	3892	4020	109	m
N	78	3128	3256	3384	3512	3640	3768	3896	4024	110	n
O	79	3132	3260	3388	3516	3644	3772	3900	4028	111	o
P	80	3136	3264	3392	3520	3648	3776	3904	4032	112	p
Q	81	3140	3268	3396	3524	3652	3780	3908	4036	113	q
R	82	3144	3272	3400	3528	3656	3784	3912	4040	114	r
S	83	3148	3276	3404	3532	3660	3788	3916	4044	115	s
T	84	3152	3280	3408	3536	3664	3792	3920	4052	116	t
U	85	3156	3284	3412	3540	3668	3796	3924	4056	117	u
V	86	3160	3288	3416	3544	3672	3800	3928	4060	118	v
W	87	3164	3292	3420	3548	3676	3804	3932	4064	119	w
X	88	3168	3296	3424	3552	3680	3808	3936	4068	120	x
Y	89	3172	3300	3428	3556	3684	3812	3940	4072	121	y
Z	90	3176	3304	3432	3560	3688	3816	3944	4076	122	z
[	91	3180	3308	3436	3564	3692	3820	3948	4080	123	{
\	92	3184	3312	3440	3568	3696	3824	3952	4084	124	¦
]	93	3188	3316	3444	3572	3700	3828	3956	4088	125	}
∧	94	3192	3320	3448	3576	3704	3832	3960	4092	126	~
_	95	3196	3324	3452	3580	3708	3836	3964	4096	127	rubout $^{D}{}_{T}$
DEC. ------>		56	57	58	59	60	61	62	63		
ASCII------>		8	9	:	;	<	=	>	?		
		High Order X & Y									

Appendix G

DEFAULT PAGE, VIEWPORT, AND WINDOW COORDINATES

MAXIMUM X AND Y COORDINATES FOR GIVEN ASPECT RATIOS (MINIMUM = Ø,Ø)

INITIAL PAGE SIZE AND PAGE FORMAT			INITIAL ASPECT RATIO											
			FULL PAGE	3X:2Y		4X:3Y (4096/3124)		1X:1Y		3X:4Y (3124/4096)		2X:3Y		
A-DRAFTING 10.5" x 7.74"			Page Viewport Window	Viewport	Window	Viewport	Window	Viewport	Window	Viewport	Window	Viewport	Window	
ADU		X	4096	4096	4096	3958.77	4096	3019.34	4096	2302.83	3124	2012.89	2730.67	X
		Y	3019.23	2730.67	2730.67	3019.34	3124	3019.34	4096	3019.34	4096	3019.34	4096	Y
GDU		X	135.66	135.66	150	131.11	131.11	100	100	76.27	100	66.67	100	X
		Y	100	90.44	100	100	100	100	100	100	131.11	100	150	Y
MM		X	266.7	266.7	266.7	257.8	257.8	196.6	196.6	149.9	149.9	131.1	131.1	X
		Y	196.6	177.8	177.8	196.6	196.6	196.6	196.6	196.6	196.6	196.6	196.6	Y
A-GRAPHING 10" x 7"														
ADU		X	4096	4096	4096	3759.30	4096	2867.20	4096	2186.80	3124	1911.47	2730.67	X
		Y	2867.20	2730.67	2730.67	2867.20	3124	2867.20	4096	2867.20	4096	2867.20	4096	Y
GDU		X	142.86	142.86	150	131.11	131.11	100	100	76.27	100	66.67	100	X
		Y	100	95.24	100	100	100	100	100	100	131.11	100	150	Y
MM		X	254.0	254.0	254.0	233.1	233.1	177.8	177.8	135.6	135.6	118.5	118.5	X
		Y	177.8	169.3	169.3	177.8	177.8	177.8	177.8	177.8	177.8	177.8	177.8	Y
B-DRAFTING 15.76" x 10.24"														
ADU		X	4096	3992.04	4096	3489.41	4096	2661.36	4096	2029.81	3124	1774.24	2730.67	X
		Y	2661.36	2661.36	2730.67	2661.36	3124	2661.36	4096	2661.36	4096	2661.36	4096	Y
GDU		X	153.91	150	150	131.11	131.11	100	100	76.27	100	66.67	100	X
		Y	100	100	100	100	100	100	100	100	131.11	100	150	Y
MM		X	400.3	390.1	390.1	341.0	341.0	260.1	260.1	198.4	198.4	173.4	173.4	X
		Y	260.1	260.1	260.1	260.1	260.1	260.1	260.1	260.1	260.1	260.1	260.1	Y
B-GRAPHING 15" x 10"														
ADU		X	4096	4096	4096	3580.29	4096	2730.67	4096	2082.67	3124	1820.44	2730.67	X
		Y	2730.67	2730.67	2730.67	2730.67	3124	2730.67	4096	2730.67	4096	2730.67	4096	Y
GDU		X	150	150	150	131.11	131.11	100	100	76.27	100	66.67	100	X
		Y	100	100	100	100	100	100	100	100	131.11	100	150	Y
MM		X	381.0	381.0	381.0	333.0	333.0	254.0	254.0	193.7	193.7	169.3	169.3	X
		Y	254.0	254.0	254.0	254.0	254.0	254.0	254.0	254.0	254.0	254.0	254.0	Y

MAXIMUM X AND Y COORDINATES FOR GIVEN ASPECT RATIOS (MINIMUM = Ø,Ø)

INITIAL PAGE SIZE AND PAGE FORMAT		INITIAL ASPECT RATIO											
		FULL PAGE	3X:2Y		4X:3Y (4096/3124)		1X:1Y		3X:4Y (3124/4096)		2X:3Y		
C-DRAFTING 21" x 15.5"		Page Viewport Window	Viewport	Window	Viewport	Window	Viewport	Window	Viewport	Window	Viewport	Window	
ADU	X Y	4096 3023.24	4096 2730 67	4096 2730.67	3963.89 3023.24	4096 3124	3023.24 3023.24	4096 4096	2305.81 3023.24	3124 4096	2015.49 3023.24	2730.67 4096	X Y
GDU	X Y	135.48 100	135.48 90.32	150 100	131.11 100	131.11 100	100 100	100 100	76.27 100	100 131.11	66.67 100	100 150	X Y
MM	X Y	533.4 393.7	533.4 355.6	533.4 355.6	516.2 393.7	516.2 393.7	393.7 393.7	393.7 393.7	300.3 393.7	300.3 393.7	262.5 393.7	262.5 393.7	X Y
C-GRAPHING 20.5" x 16"													
ADU	X Y	4096 3196.88	4096 2730.67	4096 2730.67	4096 3124	4096 3124	3196.88 3196.88	4096 4096	2438.24 3196.88	3124 4096	2131.25 3196.88	2730.67 4096	X Y
GDU	X Y	128.3 100	128.3 85.42	150 100	128.13 97.72	131.11 100	100 100	100 100	76.27 100	100 131.11	66.67 100	100 150	X Y
MM	X Y	520.7 406.4	520.7 347.1	520.7 347.1	520.7 397.14	520.7 397.14	406.4 406.4	406.4 406.4	310.0 406.4	310.0 406.4	270.9 406.4	270.9 406.4	X Y
A2-DRAFTING 22.60" x 15.75"													
ADU	X Y	4096 2854.36	4096 2730.67	4096 2730.67	3742.45 2854.35	4096 3124	2854.35 2854.35	4096 4096	2177.00 2854.35	3124 4096	1902.90 2854.35	2370.67 4096	X Y
GDU	X Y	143.50 100	143.50 95.67	150 100	131.11 100	131.11 100	100 100	100 100	76.27 100	100 131.11	66.67 100	100 150	X Y
MM	X Y	574.0 400.0	574.0 382.67	574.0 382.67	524.5 400.0	524.5 400.0	400.0 400.0	400.0 400.0	305.1 400.0	305.1 400.0	266.7 400.0	266.7 400.0	X Y
A2-GRAPHING 22.20" x 15.75"													
ADU	X Y	4096 2904.96	4096 2730.67	4096 2730.67	3808.81 2904.96	4096 3124	2904.96 2904.96	4096 4096	2215.60 2904.96	3124 4096	1936.64 2904.96	2730.67 4096	X Y
GDU	X Y	141.00 100	141.00 94.00	150 100	131.11 100	131.11 100	100 100	100 100	76.27 100	100 131.11	66.67 100	100 150	X Y
MM	X Y	564.0 400.0	564.0 376.0	564.0 376.0	524.5 400.0	524.5 400.0	400.0 400.0	400.0 400.0	305.1 400.0	305.1 400.0	266.7 400.0	266.7 400.0	X Y

MAXIMUM X AND Y COORDINATES FOR GIVEN ASPECT RATIOS (MINIMUM = Ø,Ø)

INITIAL PAGE SIZE AND PAGE FORMAT			INITIAL ASPECT RATIO											
			FULL PAGE	3X:2Y		4X:3Y (4096/3124)		1X:1Y		3X:4Y (3124/4096)		2X:3Y		
A3-DRAFTING 15.75" x 10.91"			Page Viewport Window	Viewport	Window	Viewport	Window	Viewport	Window	Viewport	Window	Viewport	Window	
	ADU	X Y	4096 2836.48	4096 2730.67	4096 2730.67	3719.03 2836.48	4096 3124	2836.48 2836.48	4096 4096	2163.37 2836.48	3124 4096	1890.99 2836.48	2730.67 4096	X Y
	GDU	X Y	144.40 100	144.40 96.27	150 100	131.11 100	131.11 100	100 100	100 100	76.27 100	100 131.11	66.67 100	100 150	X Y
	MM	X Y	400.0 277.0	400.0 266.67	400.0 266.67	363.2 277.0	363.2 277.0	277.0 277.0	277.0 277.0	211.3 277.0	211.3 277.0	184.7 277.0	184.7 277.0	X Y
A3-GRAPHING 15.35" x 10.91														
	ADU	X Y	4096 2909.21	4096 2730.67	4096 2730.67	3814.38 2909.21	4096 3124	2909.21 2909.21	4096 4096	2218.84 2909.21	3124 4096	1939.48 2909.21	2730.67 4906	X Y
	GDU	X Y	140.79 100	140.79 93.86	150 100	131.11 100	131.11 100	100 100	100 100	76.27 100	100 131.11	66.67 100	100 150	X Y
	MM	X Y	390.0 277.0	390.0 260.0	390.0 260.0	363.2 277.0	363.2 277.0	277.0 277.0	277.0 277.0	211.3 277.0	211.3 277.0	184.7 277.0	184.7 277.0	X Y
A4-DRAFTING 10.91" x 7.48"														
	ADU	X Y	4096 2809.53	4096 2730.67	4096 2730.67	3683.69 2809.53	4096 3124	2809.53 2809.53	4096 4096	2142.82 2809.53	3124 4096	1873.02 2809.53	2730.67 4096	X Y
	GDU	X Y	145.79 100	145.79 97.19	150 100	131.11 100	131.11 100	100 100	100 100	76.27 100	100 131.11	66.67 100	100 150	X Y
	MM	X Y	277.0 190.0	277.0 184.7	277.0 184.7	249.1 190.0	249.1 190.0	190.0 190.0	190.0 190.0	144.9 190.0	144.9 190.0	126.7 190.0	126.7 190.0	X Y
A4-GRAPHING														
	ADU	X Y	4096 2735.61	4096 2730.67	4096 2730.67	3586.77 2735.61	4096 3124	2735.60 2735.60	4096 4096	2086.44 2735.61	3124 4096	1823.74 2735.61	2730.67 4096	X Y
	GDU	X Y	149.73 100	149.73 99.82	150 100	131.11 100	131.11 100	100 100	100 100	76.27 100	100 131.11	66.67 100	100 150	X Y
	MM	X Y	277.0 185.0	277.0 184.7	277.0 184.7	242.6 185.0	242.6 185.0	185.0 185.0	185.0 185.0	141.1 185.0	141.1 185.0	123.3 185.0	123.3 185.0	X Y

Appendix H

USING A 4663 IN A 4662 SYSTEM SETUP

INTRODUCTION

This appendix describes limitations of the TEKTRONIX 4663 Plotter when substituted into a system that is programmed for a TEKTRONIX 4662 Plotter. These limitations may require small program adjustments to permit the 4663 Plotter to correctly interpret commands intended for a 4662 Plotter.

SERIAL INTERFACE OPERATION

The following information pertains only to Serial interface use.

CHARACTER ARGUMENTS

A 4662 character argument corresponds to a 4663 undelimited string argument that is one character long. However, 4663 undelimited string arguments cannot contain the following characters:

${}^{S}{}_{P}$, Comma, Slash, Quote, Apostrophe, or ${}^{A}{}_{T}$

These characters must be expressed within a delimited string (versus an undelimited string) to be used in 4663 string arguments.

NUMERIC ARGUMENTS

The 4662 numeric arguments for the following commands must be integer values only:

- BLOCK END
- SET TURNAROUND DELAY
- SET BLOCKSIZE

To be compatible with the 4663, the arguments for these commands must not be terminated by any of the following characters:

. (period)
E
e

These characters would be interpreted as part of the more general numeric argument forms allowed for all 4663 numeric arguments.

MAXIMUM BLOCK MODE BLOCKSIZE SPECIFICATION

In the standard 4662 Plotter, the maximum Block mode blocksize is about 1500 characters (or about 7500 characters with an additional input buffer memory installed – Option 30). For 4663 operating baud rates of 2400 or less, the maximum blocksize varies from 340 to 1020 characters, depending upon the internal expansion factor. If the baud rate is greater than 2400, additional blocksize restrictions may apply. Refer to *Block Mode Communication* in Section 4 for a discussion of the maximum blocksize specification. In most applications, a typical blocksize for both Plotters is approximately 150 characters.

DEVICE RESET COMMAND

A 4663 Plotter DEVICE RESET command does not reset interface communications parameters (this is done by an INTERFACE PARAMETER RESET command); neither command clears the 4663 output buffer. The 4663 output buffer can be cleared with a front panel Initialize function.

OPERATOR DIGITIZE OPERATION

When digitizing with the 4662, the PEN and CALL buttons specify Move or Draw points. With the 4663, the front panel Move Point or Draw Point functions specify the points where the digitizing is done with the crosshair; the pen is physically always up. In addition, the Last Point function is used on the 4663 to terminate a sequence of points. To terminate a sequence of points on the 4662 Plotter, hold the CALL button down until you hear a beep.

READ STATUS COMMAND

The information reported in Bits 10 through 15 of Status Word 0 and Bits 11 through 15 of Status Word 1 are different. Refer to the respective READ STATUS command descriptions of each plotter for details. In addition, the 4663 has defined responses for Status Words 2 through 7 while the 4662 returns zeros.

SET STATUS COMMAND

The 4663 Plotter does not support the SET STATUS command. The "P" command code is used for a PRINT command instead.

GPIB INTERFACE OPERATION

The following information pertains only to GPIB interface use.

SRQ FUNCTION

Programs using the CALL DIGITIZE function (which reads a single point at each occurrence of the SRQ bus message) will fail to read all the coordinates when used with the 4663 Plotter if coordinates are generated faster than they are read. Avoid this situation by adding a Serial Poll operation so that after the SRQ message occurs, coordinates are read until Bit 7 (in the Serial Poll Status byte) reads 0.

CALL DIGITIZE FUNCTION

The 4663 Plotter can only transmit output responses in the order requested. Thus, other output cannot be read from the 4663 when there is a CALL DIGITIZE response pending (as can be done with the 4662 Plotter).

OPERATOR DIGITIZE OPERATION

When digitizing with the 4662, the PEN and CALL buttons specify Move or Draw points. With the 4663, the front panel Move Point or Draw Point functions specify the points where the digitizing is done with the crosshair; the pen is physically always up. In addition, the Last Point function is used on the 4663 to terminate a sequence of points. To terminate a sequence of points on the 4662 Plotter, hold the CALL button down until you hear a beep. The 4663 termination action causes a three-value response to be transmitted (Status Word 0, Status Word 1, and TAG), while the 4662 termination action causes a single value status response to be transmitted (Status Word 0).

READ STATUS COMMAND

The information reported in Bits 10 through 15 of Status Word 0 and Bits 11 through 15 of Status Word 1 are different. Refer to the respective READ STATUS command descriptions of each plotter for details. For example, the 4663 has defined responses for Status Words 2 through 7 while the 4662 returns zeros.

SET STATUS COMMAND

The 4663 Plotter does not support the SET STATUS command. The "W" command code is not used.

DEVICE RESET COMMAND

A 4663 DEVICE RESET command does not clear the output buffer. The output buffer can be cleared with a front panel Initialize function.

OUTPUT NUMERIC RANGE

Noncoordinate numeric output in ASCII decimal form (such as status responses) from the 4662 has a range of −32767 to + 32767 while the output range from the 4663 is 0 to 65535.

DEVICE ADDRESSES

The 4662 device addresses can be set from 0 to 31. The 4663 device addresses can be set from 1 to 7.

MSA COMMAND DISABLE

The MSA command form cannot be disabled on the 4663 Plotter.

IFC (INTERFACE CLEAR FUNCTION)

The 4662 performs a Device Reset function (in addition to the Interface Clear function) when an IFC bus message is received. The 4663 Plotter performs only the Interface Clear function. You must use the DEVICE RESET command to perform a Device Reset function in the 4663.

Appendix I

PLOTTER/4014 COMMAND COMPATIBILITY

The following tables indicate the Plotter action each ASCII character causes with respect to the current command decoding mode. These actions are as close as possible to the action the character would initiate if it were received by a 4014 terminal equipped with the Extended Graphics Module (EGM) Option that was in the same mode.

The response of most 4010 Series terminals is similar to the response of a 4014 (except for EGM-indicated functions). If the action described in the tables does not duplicate the terminal action, it is because:

- The command is not within the capabilities of the Plotter.
- There is a corresponding command for the Plotter (implemented uniquely, either because it is desired to perform the function independently of the terminal, or because there is more data required to perform the function for the Plotter, requiring an expanded calling sequence).
- The Plotter is a loop-thru device intended for use in conjunction with a terminal. This dictates that some functions be slightly different to complement the capabilities of the terminal.

In summary, all major EGM functions for 4014's and 4016's are supported with the exception of:

● Margin 1	(Not supported)
● Alpha Wraparound	(Not supported)
● Special Point Plot	(Not supported)
● Incremental Plot	(Not supported)
● Write-Thru Vectors	(Drawn as normal vectors)
● Defocused Vectors	(Drawn as normal vectors)

The Style II command characters and Plotter responses are summarized in Table I-1. The action shown in this table assumes that the alpha tab separator parameter is 0 and that the current attention character is $^{\mathrm{E}}\mathrm{C}$. The characters shown in the table begin with $^{\mathrm{N}}\mathrm{U}$ (ADE 1) and increase up to $^{\mathrm{D}}\mathrm{T}$ (ADE 127), as listed in Appendix B.

NOTE

Attention Action mode is referred to in the 4014 documentation as LCE (Last Character Escape) mode.

Table I-1

4663 STYLE II COMMAND CHARACTERS

Character	Character Effect		
	Alpha Mode	Graph/Point Plot Mode	Attention Action Mode
$^{N}{}_{U}$			Remain in $^{A}{}_{T}$ mode
$^{S}{}_{H}$ through $^{A}{}_{K}$			Return from $^{A}{}_{T}$ mode with no action
$^{B}{}_{L}$	Ring bell	Ring bell, set next graphic coordinate interpretation to DRAW	Ring bell, does not set next graphic coordinate interpretation to DRAW
$^{B}{}_{S}$	Move left 1 space, no wraparound at left margin		Return from $^{A}{}_{T}$ mode, does not move left one space
$^{H}{}_{T}$	Move right 1 space, wraparound at right margin with auto $^{C}{}_{R}$/$^{L}{}_{F}$ if margin is active		Return from $^{A}{}_{T}$ mode, does not move right one space
$^{L}{}_{F}$	Move down 1 line, no wraparound or move to Margin 1		Remain in $^{A}{}_{T}$ mode, does not do $^{L}{}_{F}$
$^{V}{}_{T}$	Move up 1 line, no wraparound at top of page		Return from $^{A}{}_{T}$ mode, does not move one line up
$^{F}{}_{F}$			Move to home position and set Alpha mode, set solid line type, does not clear extra graphic bytes
$^{C}{}_{R}$	Move to left margin, does not clear extra graphic bytes, no $^{C}{}_{R}$/$^{L}{}_{F}$ option (2741 Interface)	Does not set Alpha mode, does not do $^{C}{}_{R}$	Remain in $^{A}{}_{T}$ mode, does not do $^{C}{}_{R}$
$^{S}{}_{O}$		Begin circle spec, the X part of the coordinate pair is interpreted as radius (Option 31 only)	Activates alternate alpha font
$^{S}{}_{I}$		Begin arc spec, next two coordinate pairs are interpreted as the intermediate and end points of the arc (Option 31 only)	Activates standard alpha font
$^{D}{}_{L}$ through $^{S}{}_{B}$			Return from $^{A}{}_{T}$ mode with no action
$^{E}{}_{C}$	Set $^{A}{}_{T}$ mode[a]	Set $^{A}{}_{T}$ mode[a]	Remain in $^{A}{}_{T}$ mode[a]
$^{F}{}_{S}$	Set Point mode, save Line type	Set Point mode, save Line type	Return from $^{A}{}_{T}$ mode, does not do special point plot
$^{G}{}_{S}$	Set Graph mode, set next graphic coordinate interpretation to MOVE	Set next graphic coordinate interpretation to MOVE	Set Graph mode if now in Alpha mode, set next graphic coordinate interpretation to MOVE
$^{R}{}_{S}$	Does not do incremental plot	Does not do incremental plot	Return from $^{A}{}_{T}$ mode, does not do incremental plot
$^{U}{}_{S}$		Set Alpha mode, restore line type	Set Alpha mode, restore line type

Table I-1 (cont.)

4663 STYLE II COMMAND CHARACTERS

Character	Character Effect		
	Alpha Mode	Graph/Point Plot Mode	Attention Action Mode
$^{S}{}_{P}$ and 1 thru 7	DRAW character	HIY or HIX. If last graphic byte received was not LOY or if a $^{G}{}_{S}$ was received after the last LOY, then interpret as a HIY graphic byte. If last graphic byte was LOY, then set HIX	Return from $^{A}{}_{T}$ mode with no action
8			Set character dimension to 4014 large character size
9			Set character dimension to 4014 #2 character size
:			Set character dimension to 4014 #3 character size
;			Set character dimension to 4014 small character size
< = >			Return from $^{A}{}_{T}$ mode with no action
?			The ? character is converted to a $^{D}{}_{L}$ and interpreted as a LOY graphic byte
@ thru –		Interpret as LOX graphic byte, execute MOVE, DRAW, ARC, or CIRCLE. Set next graphic coordinate interpretation to DRAW	Return from $^{A}{}_{T}$ mode with no action
' and e thru h m thru p u thru w	DRAW character	Interpret as LOY graphic byte. If last byte was LOY, set previous LOY as extra graphic byte (EB), previous EB as EEB, and previous EEB as EEEB	Set SOLID line type, does not do "defocus" or "write-through"
a i q			Set 4014 DOTTED line type, does not do "defocus" or "write-through"
b j r			Set 4014 DOT-DASH line type, does not do "defocus" or "write-through"
c k s			Set 4014 SHORT DASH line type, does not do "defocus" or "write-through"
d l t			Set 4014 LONG DASH line type, does not do "defocus" or "write-through"
$^{D}{}_{T}$	Draw $^{D}{}_{T}$ unless DEL IGNORE is specified	Interpret as LOY graphic byte unless DEL IGNORE specified	Remain in $^{A}{}_{T}$ mode

[a] These actions are the effects of whatever character has been selected as the current attention character.

Appendix J

ERROR TYPES

The following tables show the Parameter Entry switch display for each error type used in the Plotter. Table J-1 lists nonfatal error types while Table J-2 lists fatal error types. The Parameter Entry display shows the states of the lighted switches on the Parameter Entry Module if (1) the Parameter Entry Card was pushed all the way in when the error occurred, or (2) the Error Data line on the Parameter Entry Card has been selected and the ERROR CODE SWITCH is pressed. All references to errors in this manual identify the errors by the HEX equivalent value of the light displays. Refer to the *Error Data Line* description in the *4663 Operator's Manual* to access the error parameters using the Parameter Entry Module.

NONFATAL ERRORS

Nonfatal errors shown in Table J-1 indicate when the Plotter is unable to recognize or perform a command. The usual response is that the command is ignored or some assumptions are made and processing continues. The first group of errors (01 through 0C) indicate a problem in communicating with the Plotter. The second group of errors (40 through 51) indicate a problem with recognition and/or specification of a command. The third group (80 through 8D) indicate problems with command processing (which may be caused by the specification and/or use of a command).

Table J-1
NONFATAL ERROR TYPES

* = Light is ON

Parameter Entry Display	Hex	Decimal	Cause of Error
OOOOOOO*	01	1	Serial—data overrun
OOOOOO*O	02	2	Serial—framing error
OOOOOO**	03	3	Serial—parity error
OOOOO*OO	04	4	Serial—input buffer full error
OOOOO*O*	05	5	Output attempt when message is in progress
OOOOO**O	06	6	Unused
OOOOO***	07	7	Output Buffer full
OOOO*OOO	08	8	Serial—block size exceeds input buffer size
OOOO*OO*	09	9	Serial—block checksum error
OOOO*O*O	OA	10	GPIB—illegal secondary address
OOOO*O**	OB	11	Unused
OOOO**OO	OC	12	Unused
O*OOOOOO	40	64	Integer argument exceeds 16-bit value
O*OOOOO*	41	65	Illegal paper advance command
O*OOOO*O	42	66	Illegal macro command usage
O*OOOO**	43	67	Macro called itself
O*OOO*OO	44	68	Integer outside legal range
O*OOO*O*	45	69	Toc many entries or string too long (dash pattern)
O*OOO**O	46	70	Unused
O*OOO***	47	71	Unidentified interface command error
O*OO*OOO	48	72	Invalid command for selected output format
O*OO*OO*	49	73	Scale = 0
O*OO*O*O	4A	74	Viewport of 0 length or outside page boundary
O*OO*O**	4B	75	Window less than or equal to 0
O*OO**OO	4C	76	Page upper-right on top of lower-left position
O*OO**O*	4D	77	Argument type error (interface)
O*OO***O	4E	78	Unused
O*OO****	4F	79	Output value range error
O*O*OOOO	50	80	Illegal downloadable character command
O*O*OOO*	51	81	Illegal downloadable character command usage
*OOOOOOO	80	128	Invalid switch entry (Parameter Entry)
OOOOOO	81	129	Battery backup RAM checksum error
*OOOOO*O	82	130	Transform stack underflow or overflow
*OOOOO**	83	131	Transform cannot produce virtual digitizing data because of transform values
*OOOO*OO	84	132	RAM verify error—block not used
*OOOO*O*	85	133	Insufficient memory for buffer allocation
*OOOO**O	86	134	Cannot find specified macro
*OOOO***	87	135	Illegal alpha table command type
*OOO*OOO	88	136	Unused
*OOO*OO*	89	137	Illegal monitor usage
*OOO*O*O	8A	138	Output processing routine missing
*OOO*O**	8B	139	Number greater than 99999.999 in fixed ASCII conversion
*OOO**OO	8C	140	Too many PROM alpha tables in system
*OOO**O*	8D	141	Selected interface hardware or ROM not present

FATAL ERRORS

Fatal errors normally indicate either a Plotter failure or incorrect installation of Plotter hardware. Fatal errors 0A and 11 indicate unexpected software processing conditions that can occur from improper use of Plotter commands. The remainder of the fatal errors usually indicate a hardware failure and require diagnosis by a qualified service technician.

NOTE

Under some conditions, a fatal error will lock up the Plotter's Parameter Entry Module and prevent access of any error parameters.

Table J-2

FATAL ERROR TYPES

* = Light is ON

Parameter Entry Display	**Hex**	**Decimal**	**Cause of Error**
OOOOOOO*	01	1	Insufficient RAM
OOOOOO*O	02	2	RAM check error
OOOOOO**	03	3	ROM check error
OOOOO*OO	04	4	Undefined common subroutine
OOOOO*O*	05	5	Command dispatch error
OOOOO**O	06	6	Software interrupt
OOOOO***	07	7	Nonmaskable interrupt
OOOO*OOO	08	8	Nonexistent memory reference
OOOO*OO*	09	9	Unable to create a buffer (insufficient RAM)
OOOO*O*O	OA	10	Unexpected system error
OOOO*O**	OB	11	No service routine for level
OOOO**OO	OC	12	No self interrupt routine for level
OOOO**O*	OD	13	Motion Synchronization address not in first 256 bytes of ROM
OOOO***O	OE	14	No routine for specified system command
OOOO****	OF	15	ROM in wrong socket
OOO*OOOO	10	16	Motion buffer overrun
OOO*OOO*	11	17	Unexpected memory overflow

NONFATAL ERROR CAUSES AND RESPONSES

The following listing describes possible causes for each nonfatal error detected by the Plotter. The Plotter's response to the error condition is also indicated (when appropriate). You can use either the hexidecimal or decimal forms of the error number to locate the error description. Decimal values can be used if the error parameter was obtained by a READ ERROR command response. Hexidecimal values are decoded from the lights on the Parameter Entry Module.

Some errors listed can be generated by more than one cause. For most errors, refering to the indicated error parameters (EP1, EP2, . . .) may identify the cause of the error. Both the hex and decimal equivalents of the error parameter entries are listed. For the few errors where error parameter information is insufficient to identify the error cause, the Plotter functions performed at the time of error reporting may help identify the error cause. Refer to the *4663 Interactive Digital Plotter Operator's Manual* for more information on how to read the error parameters.

The description of most errors refers to related Plotter commands. Go to the appropriate command description for more information about legal and illegal usage of the command.

When the contents of an error parameter is the ASCII Decimal Equivalent value of the character, refer to Appendix B to determine the character.

HEX	DEC	Description (Nonfatal Errors)
01	1.	**Serial — Data Overrun**
		Plotter processing does not allow characters to be read as fast as they are received and characters have been lost. Switch to a lower baud rate.
02	2.	**Serial — Framing Error**
		A garbled character was received and characters have been lost. Usually due to a noisy communication environment or wrong baud rate.
03	3.	**Serial — Parity Error**
		Occurs when character is garbled or when receive parity selection does not agree with the host. Character is used as received.
04	4.	**Serial — Input Buffer Full Error**
		Ability of the Plotter to temporarily store commands has been exceeded and some commands are lost. Input rate control method is inappropriate (see Serial Interface section).
05	5.	**Output Attempt When Message Is In Progress**
		Another type of output was requested while operator digitizing was enabled (non-permanently). Operator digitizing is disabled and the requested response is transmitted, (See OPERATOR DIGITIZE ENABLE command).
06	6.	**Unused**
07	7.	**Output Buffer Full**
		More than 127 output requests have been received and no responses have been transmitted. Command is ignored.
08	8.	**Serial — Block Size Exceeds Input Buffer Size**
		SET BLOCK SIZE command argument larger than bytes available for input buffer use. Block size is set to 0. (See Serial Interface section).

HEX	DEC	DESCRIPTION (Nonfatal Errors)
09	9.	**Serial — Block Checksum Error**

		A garbled blockmode block has been received. Block is discarded and a negative acknowledge is generated.
		EP2 = Received checksum (high byte)
		EP3 = Received checksum (low byte)
		EP4 = Computed checksum (high byte)
		EP5 = Computed checksum (low byte)

0A	10.	**GPIB — Illegal Secondary Address**
		While the Plotter has been addressed as a listener or is in Listen Only mode, a secondary address is received that does not have a TEKTRONIX 4050 Series secondary address command definition. The secondary address is ignored.
0B	11.	**Unused**
0C	12.	**Unused**
40	64.	**Integer Argument Exceeds 16-Bit Value**
		The argument of an OPERATOR DIGITIZE ENABLE command is less than 0 or greater than 65535. The command is ignored.
41	65.	**Illegal Paper Advance Command**

		An illegal media change command/function has been received or activated via the front panel while the media mode is sheet. Command/function is ignored.
		EP1 = 64/100. ADVANCE MEDIA command.
		EP1 = 0A/10. Manual Media function or SET FORM LENGTH command.

42	66.	**Illegal Macro Command Usage**

		A macro definition contains an illegal command as shown below. Command is ignored and macro definition continues.
		EP1 = 7B/123. READ MACRO STATUS command.
		EP1 = 7C/124. SET AUTO MACRO command.
		EP1 = 70/125. DELETE MACRO command.
		or
		An END MACRO DEFINITION command is received without a preceeding BEGIN MACRO DEFINITION command. Command is ignored.
		EP1 = 79/121. END MACRO DEFINITION command.

43	67.	**Macro Calls Itself**
		While expanding a macro, an EXPAND MACRO command is encountered which references a macro already partly expanded. The command is ignored and the macro expansion continues. EP1 = The number of the partially expanded macro.
44	68.	**Integer Outside Legal Range**
		One of the following groups of commands contains a numeric argument which is outside the defined range for the command. The number may be expressed in any numeric form. The legal argument ranges, the error parameters saved, and the resultant command action are different for each command and are summarized in Table J-3. Some error reports are not unique and must be identified by considering the commands currently in use.

HEX	DEC	DESCRIPTION (Nonfatal Errors)
45	69.	**Too Many Entries or String Too Long**
		One of the following commands which cause either several numeric values or string characters to be stored has overflowed the allocated storage space. ● SET SIGNATURE CHARACTER command. Storage size = 1. For strings larger than one character, the first character is used. One error reported on last character. EP1 = 9B/155. ● SET BYPASS CANCEL CHARACTER command. Storage size = 1. For strings larger than one character, the first character is used. One error reported on last character. EP1 = AC/172. ● SET PROMPT STRING command. Storage size = 4. For strings larger than four characters, the first four characters are used. One error reported on last character. EP1 = FD/253. ● BEGIN CHARACTER DEFINITION command. Storage size = 1. Definition continues for first string character. One error reported on first character. EP1 = 56/86. EP4 = Current font number. EP5 = First string character (ADE Value). ● SET DASH PATTERN command. Storage size = 20. More than 20 dash pattern elements have been received. First 20 patterns are saved and one error reported for each additional pattern element. EP1 = Current dash pattern element value (0-255).
46	70.	**Unused**
47	71.	**Unidentified Interface Command Error**
		The current command code is not recognized as a Plotter command. The command and any following arguments are discarded. EP4 = First unrecognized command code character (ADE value). EP5 = Second unrecognized command code character (ADE value — if the first command code character is A, B, C, or D), otherwise EP5 is undefined.
48	72.	**Invalid Command for Selected Output Format**

		The current command generates output which cannot be encoded for transmission using the current Command/Response Format. The command is discarded.
		EP1 = 3F/63. READ VIEWPORT command. (2 errors indicated)
		EP1 = 62/98. READ MACRO STATUS command. (1 error/macro indicated)
		EP1 = 62/98. READ FORM LENGTH command.
		If a READ ERROR command response cannot be encoded, no error is reported and no output is generated.

HEX	DEC	DESCRIPTION (Nonfatal Errors)
49	73.	**Scale = 0**
		If an argument of a SET ALPHA SCALE command is 0, the corresponding scale value is set to a small positive value and execution continues. Also, if an argument of a previous SET ALPHA SCALE command was 0, each time an attempt is made to print an alpha character, this error is reported, the corresponding scale value is set to a small positive value, and the character is printed. In both cases, one error is indicated for each 0 scale argument.
4A	74.	**Viewpoint of 0 Length or Outside Page Boundary**
		If a Viewport specification set either by the SET VIEWPORT command or by the front panel Set Viewport function results in a Viewport axis of 0 length, the axis is set to a small positive length and execution continues. If any specified Viewport axis end point is outside the Page boundary, the point is collapsed to the Page boundary, the affected axes are set to a small positive length (if the collapsed axis length is negative or 0), and execution continues. One error is indicated for each affected axis.
4B	75.	**Window Less Than or Equal to 0**
		If any Window axis length specified by a SET WINDOW command is less than or equal to 0, the axis is set to a small positive value and execution continues. One error is indicated for each affected axis.
4C	76.	**Page Upper-Right on Top of Lower-Left Position**
		If the length of any Page axis specified by a front panel Set Page function is 0, the axis is set to a small positive value and execution continues. One error is indicated for each affected axis.
4D	77.	**Argument Type Error**
		An argument for the current command cannot be recognized as the type of argument defined for the command. The command and arguments are discarded. EP1 = Command Code (see below) EP4 = Received argument type 01/1. String Argument 02/2. Numeric Argument EP5 = Expected argument type 01/1. String Argument 02/2. Numeric Argument The command code (EP1) could be one of the following: 06/6. DRAW (GPIB) command. 18/24. JOYSTICK AXIS DISABLE command. 1F/31. MOVE (GPIB) command. 21/33. SELECT PEN command. 22/35. BLOCK END (Serial) command. 2E/46. SET VIEWPORT command. 2F/47. SET WINDOW command. 30/48. SET TRANSLATION command. 31/49. SET SKEW command. 32/50. SET ALPHA ROTATION command. 33/51. SET ALPHA SCALE command. 37/55. SELECT COORDINATE TYPE (GPIB) command. 3E/62. SELECT CLIPPING CONTROL command.

HEX	DEC	DESCRIPTION (Nonfatal Errors)
		43/67. SET CHARACTER X-EXTENT command.
		47/71. OPERATOR DIGITIZE ENABLE (GPIB) command.
		49/73. CHARACTER MOVE command.
		4A/74. SELECT GRAPHIC UNITS (GPIB) command.
		4A/74. SET ALPHA DIMENSION command.
		4B/75. SET ALPHA SIZE command.
		4C/76. SET ALPHA RATIO command.
		4D/77. SET TAB SEPARATION command.
		4E/78. SET ALPHA MARGIN SEPARATION command.
		4F/79. SET ALPHA SCALE command.
		50/80. READ STATUS (GPIB) command.
		50/80. SET ALPHA ROTATION command.
		51/81. SET ALPHA SLANT command.
		52/82. SET ALPHA STANDARD FONT command.
		53/83. SELECT COMMAND/RESPONSE FORMAT (GPIB) command.
		53/83. SELECT ALTERNATE ALPHA FONT command.
		54/84. SELECT ALPHA SPACING CONTROL command.
		56/86. BEGIN CHARACTER DEFINITION command.
		57/87. DELETE FONT DEFINITION command.
		58/88. DELETE CHARACTER DEFINITION command.
		59/89. SET DOWNLOADED CHARACTER SIZE command.
		5C/92. DEVICE RESET (GPIB) command.
		63/99. SET FORM LENGTH command.
		64/100. ADVANCE MEDIA command.
		65/101. SELECT DEVICE UNITS (GPIB) command.
		67/103. SET DASH PATTERN command.
		68/104. SET DASH PATTERN LENGTH command.
		69/105. SELECT LINE TYPE command.
		6E/110. AXIS command.
		72/114. SET ARC SMOOTHNESS command.
		78/120. BEGIN MACRO DEFINITION command.
		7A/122. EXPAND MACRO command.
		7C/124. SET AUTO MACRO command.
		7D/125. DELETE MACRO command.
		80/128. BLOCK END (GPIB Format 7) command.
		80/128. READ STATUS (GPIB Format 7) command.
		80/128. SET BLOCKSIZE (GPIB Format 7) command.
		80/128. SET BYPASS CANCEL CHARACTER (GPIB Format 7) command.
		80/128. SET PROMPT STRINT (GPIB Format 7) command.
		80/128. SET SIGNATURE CHARACTER (GPIB Format 7) command.
		80/128. SET TURNAROUND DELAY (GPIB Format 7) command.
		BC/188. SELECT COMMAND/RESPONSE FORMAT (Serial) command.
		DA/218. OPERATOR DIGITIZE ENABLE (Serial) command.
		DD/221. DRAW (Serial) command.
		E0/224. MOVE (Serial) command.
		E3/227. SELECT COORDINATE TYPE (Serial) command.
		E6/230. SELECT DEVICE UNITS (Serial) command.
		E9/233. SELECT GRAPHIC UNITS (Serial) command.
		EC/236. DEVICE RESET (Serial) command.
		F2/242. READ STATUS (Serial) command.
		F5/245. SET TURNAROUND DELAY (Serial) command.
		FB/251. SET BLOCK SIZE (Serial) command.

HEX	DEC	DESCRIPTION (Nonfatal Errors)
4E	78.	**Unused**
4F	79.	**Output Value Range Error**

		A response value generated by a DIGITIZE command or an Operator Digitize Point function cannot be encoded in the packed binary output format because of its value (value limits for this format are 0 to 4096). The value is clipped to the nearest limit.
		EP1 = 5A/90. DIGITIZE command.
		EP1 = 5B/91. Operator Digitize Point function.

50	80.	**Illegal Downloadable Character Command**
		A command other than SET CHARACTER X-EXTENT, MOVE, DRAW, DRAW CIRCLE, DRAW ARC, or SELECT NONADVANCING CHARACTER was received within a character definition. The command is discarded and the partial character definition is deleted.
51	81.	**Illegal Downloadable Character Command Usage**

		One of the following commands was received when a downloadable character definition was not in process. The command is ignored.
		EP1 = 43/67. SET CHARACTER X-EXTENT command.
		EP1 = 44/68. SELECT NONADVANCING CHARACTER command.
		EP1 = 45/69. END CHARACTER DEFINITION command

80	128.	**Invalid Switch Entry**
		A Parameter Entry Module switch was pushed while the card was fully inserted.
81	129.	**Battery Backup RAM Checksum Error**
		A powerup checksum check of the contents of the memory in the Parameter Entry has failed. Possible low battery or other hardware problem. All Parameter Entry Card selections should be verified.
82	130.	**Transform Stack Overflow or Underflow**
		A RESTORE PREVIOUS TRANSFORM command was executed without a corresponding preceeding SAVE CURRENT TRANSFORM command. Processing continues utilizing the current transform values. or A SAVE CURRENT TRANSFORM command was executed with insufficient memory available to store the transform values. The store is not done and processing continues.
83	131.	**Transforming Cannot Produce Virtual Digitizing Data Due to Transform Values**
		If the sum of the two axis skew values specified in a SET SKEW command is 90 or 270 degrees, subsequent Digitize or front panel Point functions cause this error to be generated. Output values are generated but they are only partially processed and are invalid.
84	132.	**RAM Verify Error**
		One or more blocks of 64 bytes failed the powerup RAM test and was masked as unavailable for users. This error indicates a hardware problem. EP2 and EP3 contain the beginning address (high byte and low byte respectively) of the last bad block.

HEX	DEC	DESCRIPTION (Nonfatal Errors)
85	133.	**Insufficient Memory for Buffer Allocation**

		Insufficient memory exists to complete the storage of a programmable macro or a downloadable character. If a programmable macro definition cannot be completed, then the partial definition is deleted. The definition of a downloadable character is a two stage process: First, the character is defined, and secondly, the font containing the updated character is recorded in the font directory. If memory is exhausted during the character definition, the partial character is deleted. If insufficient memory is available to record the updated font (in the font directory) the entire font is deleted.
		If the error parameters contain one of the following values, the overflow occurred during a downloadable character definition.
		EP1 = 56/86. BEGIN CHARACTER DEFINITION command.
		EP1 = 59/89. SET DOWNLOADED CHARACTER SIZE command.
		EP1 = 45/69. END CHARACTER DEFINITION command.
		EP1 = 43/67. SET CHARACTER X-EXTENT command.
		EP1 = 44/68. SELECT NONADVANCING CHARACTER command.
		EP1 = 34/52. MOVE command.
		EP1 = 35/53. DRAW command.
		EP1 = 74/116. DRAW ARC command.
		EP1 = 75/117. DRAW CIRCLE command.
		EP4 = Current font number.
		EP5 = Current character being defined (ADE value).
		If the error parameters contain the following values, then the overflow occurred during a programmable macro definition.
		EP1 = Current macro number
		EP2 = 58/88. and EP3 = 55/85. (Macro too large)
		or
		EP2 = 5C/92. and EP3 = B7/183. (Insufficient storage to begin definition)
		or
		EP2 = 5D/93. and EP3 = 5B/91. (Macro too large)

86	134.	**Cannot Find Specified Macro**
		A macro specified by either the EXPAND MACRO command or the SET AUTO MACRO command is not currently defined when an attempt was made to expand the macro. The command is ignored. EP1 = Non-existant macro number.
87	135.	**Illegal Alpha Table Command Type**
		An unrecognizable encoded command byte was read from a stored alpha character definition. Drawing of the character is aborted.
88	136.	**Unused**
89	137.	**Illegal Monitor Usage**
		Current Plotter internal processing has produced unexpected status. The current processing is aborted and command execution continues.
8A	138.	**Output Processing Routine Missing**
		The output encoding routine referred by an output generating command cannot be located. The command is aborted.

HEX	DEC	DESCRIPTION (Nonfatal Errors)
8B	139.	**Number Greater Than 99999.996 in Fixed ASCII Conversion**
		One or more of the coordinates generated by a DIGITIZE command or an Operator Digitize Point function has a value greater than 99999.996 and the output format is fixed point ASCII decimal. The value is clipped to 99999.996 and execution continues.
8C	140.	**Too Many PROM Alpha Tables in System**
		At powerup more than two optional alpha tables were encountered. Additional tables are ignored.
8D	141.	**Selected Interface Hardware or ROM Not Present**
		The hardware and/or firmware to be activitated by a Parameter Entry Card selection made on the Interface Select line is not present.

Table J-3
INTEGER OUTSIDE LEGAL RANGE

Command	Legal Range	Error Parameters			Comments
		1	2	3	
AXIS	> 0	6E/110.	none	none	Error if < 0. Command ignored.
BEGIN CHARACTER DEFINITION	0-15	56/86.	55/85.	8A/138.	Command and following character definition is ignored.[a]
BEGIN MACRO DEFINITION	0-255	78/120.	5C/92.	94/148.	Command and following macro definition is ignored.
BLOCK END	0-4095	23/35.	none	none	Clips checksum to nearest limit. Causes checksum error and continues.
DELETE CHARACTER DEFINITION	0-15	58/88.	57/87.	3C/60.	Command ignored.[a]
DELETE FONT DEFINITION	0-15	57/87.	57/87.	7C/124.	Command ignored
DELETE MACRO	0-255	7D/125.	5F/95.	BE/190.	Command ignored.
DEVICE RESET	0-1	01/1.	C6/198.	AA/170.	Command ignored.
DRAW					See footnote b.
DRAW ARC					See footnote b.
DRAW CIRCLE					See footnote b.
EXPAND MACRO	0-255	7A/122.	5E/94.	7F/127.	Command ignored.
JOYSTICK AXIS DISABLE	0-3	none	A7/167.	D3/211.	Command ignored.
MOVE					See footnote b.
PRINT		47/71.			See footnote b.
PRINT CENTERED		48/72.			See footnote b.
READ STATUS	0-7	03/3.	C7/199.	7B/171.	Serial Formats 1-2 and GPIB Formats 1-4. Sets value to 0 and continues.
	0-15	03/3.	C7/199.	7B/171.	Serial Formats 3-4 and GPIB Formats 5-6. Sets value to 0 and continues.
SELECT ALPHA SPACING CONTROL	0-1	54/84.	94/148.	85/133.	Command ignored.
SELECT ALTERNATE ALPHA FONT	0-15	53/83.	94/148.	5F/95.	Command ignored.

Table J-3 (cont)

INTEGER OUTSIDE LEGAL RANGE

Command	Legal Range	Error Parameters 1	2	3	Comments
SELECT CLIPPING CONTROL	0-1	3E/62.	BA/186.	09/9.	Command ignored.
SELECT COMMAND/ RESPONSE FORMAT	1-4 Serial	none	E9/233.	26/38.	For arguments < 0 and > 4. Command ignored.
	1-4 Serial	BC/188.	none	none	For arguments = 0. Command ignored.
	1-7 GPIB	none	EA/234.	4A/74.	For arguments < 0 and > 7. Command ignored.
	1-7 GPIB	53/83.	none	none	For arguments = 0. Command ignored.
SELECT COORDINATE TYPE	0-1	none	EA/234.	78/120.	Serial, Command Ignored.
	0-1	none	E9/233.	94/148.	GPIB, Command Ignored.
SELECT DEVICE UNITS	0-2	none	EA/234.	8B/139.	Serial, Command Ignored.
	0-2	none	E9/233.	C5/197.	GPIB, Command Ignored.
SELECT GRAPHIC UNITS	0-1	none	EA/234.	B7/183.	Serial, first argument command ignored.
	0-1	none	E9/233.	FD/253.	GPIB, first argument, command ignored.
	0-1	40/64.	BD/141.	3E/62.	Second arguments: Input graphic units changed — output graphic units not changed.
SELECT LINE TYPE	0-3	69/105.	80/128.	ED/237.	Command Ignored.
SELECT PEN	0-255	21/33.	A7/167.	98/152.	Command Ignored.
SELECT STANDARD ALPHA FONT	0-15	52/82.	94/148.	5F/95.	Command ignored.
SET AUTO MACRO	0-255	7C/124.	58/88.	D0/208.	Command ignored.
SET BLOCK SIZE	0-65535	FB/251.	none	none	Clips blocksize to nearest limit and continues.
SET BYPASS CANCEL CHARACTER		AC/172.			See footnote a.
SET DASH PATTERN	0-255	67/103.	81/129.	1A/26.	Pattern element is set to 255 (MAX dash) and continues.
SET DOWNLOADED CHARACTER SIZE	0-15	59/89.	54/84.	75/117.	Command ignored.
SET FORM LENGTH	0-65535	63/99.	8F/143.	4B/75.	Command ignored.
SET PROMPT STRING		FD/253.			See footnote a.
SET SIGNATURE CHARACTER		9B/155.			See footnote a.
SET TURNAROUND DELAY	0-65535	F5/245.	none	none	Clips delay to nearest limit and continues.

[a] If the "/number" form of string arguments is used, and if the numeric argument range is less than 0 or greater than 255, then a value of 0 is assumed and execution continues.

[b] If a numeric argument for these commands (when used in a downloadable character definition) has a value outside the following ranges, then an undefined error is generated, the command is discarded, and definition of the downloadable character continues:

- xmin-64 TO xmin+175 (for x arguments or circle radius arguments)
- ymin-64 TO ymin+175 (for y arguments)

FATAL ERROR CAUSES/RESPONSES

The following listing describes error parameters for each fatal error detected by the Plotter.

HEX	DEC	DESCRIPTION (Fatal Errors)
01	1.	**Insufficient RAM**
		At power-up, specific RAM locations (the first location in each 1k byte block of memory, starting at 0) are scanned for read/write function to determine available memory. This error is issued if less than 3K bytes of RAM exist (which is the minimum amount required for useful operation). Error parameters are: EP2 = Address high byte of the beginning of the last 1K byte block of functional RAM. EP3 = Address low byte of the beginning of the last 1K byte block of functional RAM.
02	2.	**RAM Check Error**
		This error indicates a failure of a RAM test performed at powerup on the minimum amount of RAM required for useful operation (3K bytes starting at 0). Error parameters are: EP1 = The bit pattern which failed to store. EP2 = Address high byte of the bad location. EP3 = Address low byte of the bad location.
03	3.	**ROM Check Error**
		This error indicates that a powerup checksum test has failed on one of the system ROMs. Error parameters are: EP1 = Starting address high byte of the ROM being tested. EP2 = Checksum high byte of the ROM being tested. EP3 = Checksum low byte of the ROM being tested.
04	4.	**Undefined Common Subroutine**
		During operation, a firmware subroutine was referenced that had not been defined at power-up. This error indicates one of the following: ● Modified firmware has referenced a nonexistant subroutine, ● A system ROM was missing or not accessible during powerup, or ● A RAM failure occurred which caused the address of a common subroutine defined at powerup to be lost. Error parameters are: EP2 = Address high byte of the ROM where subroutine was called from. EP3 = Address low byte of the ROM where subroutine was called from. EP4 = Address high byte of the RAM where the subroutine starting address was expected. EP5 = Address low byte of the RAM where the subroutine starting address was expected.

HEX	DEC	DESCRIPTION (Fatal Errors)
05	5.	**Command Dispatch Error**
		An invalid service routine start address, interrupt level identifier, Parameter Entry Card line identifier, or command code was encountered at powerup. This error indicates one of the following: ● Modified firmware which references a service routine incorrectly. ● A ROM access failure which causes a service routine to be incorrectly defined. Error parameters are: EP1 = The current interrupt level identifier, the current Parameter Entry Card line identifier, or the current command code. EP2 = Address high byte of the ROM where the service routine was incorrectly defined. EP3 = Address low byte of the ROM where the service routine was incorrectly defined. or EP2 = High byte of bad service routine vector.
06	6.	**Software Interrupt**
		An unexpected software interrupt has occurred. Software interrupts are not used for normal system operation.
07	7.	**Nonmaskable Interrupt**
		An unexpected non-maskable interrupt has occurred. Non-maskable interrupts are not used for normal system operation.
08	8.	**Nonexistant Memory Reference**
		This error indicates that a system processor tried to read or write to a hardware address which did not respond causing the system bus to time out. Either the hardware at that address failed, or a a system failure caused an undefined address to be generated. Error paraemters are: EP2 = Address high byte of the location following the instruction that the processor was executing when the bus timed out. EP3 = Address low byte of the location following the instruction the processor was executing when the bus timed out.
09	9.	**Unable to Create a Buffer (Insufficient RAM)**
		Insufficient RAM was found during power-up to allow the creation of a system buffer.

HEX	DEC	DESCRIPTION (Fatal Errors)
0A	10.	**Unexpected System Error**
		This error indicates a condition that could be detected by the firmware but was not expected to occur during normal operation. Because the error was not expected, appropriate assumptions could not be made and processing was halted. The following group of errors lists the activities when an Unexpected System Error could be detected. EP2 = 57, EP3 = 1E A currently defined Option 32 font buffer was being searched for a specified character, and a Font 0 marker was followed by a character count of 0. EP2 = 50, EP3 = 50 An Option 32 character stroke load/unload pointer is outside the stroke buffer area. EP2 = AA, EP3 = 27 Undefined internal motion command restart status found. EP2 = AA, EP3 = 57 Undefined internal motion task code read from motion task buffer. EP1 contains the task code. EP2 = AC, EP3 = BD Unexpected internal motion command buffer full when entering internal motion command. — OR — Unexpected internal motion task buffer empty when reading an internal motion task. EP2 = B9, EP3 = E4 Unexpected stack overflow while pushing transform bytes into the transform stack. EP2 = CA, EP3 = EA Unexpected buffer overflow while entering command restart request. EP2 = E4, EP3 = 98 Unexpected output data buffer empty while transmitting an output message. EP2 = E9, EP3 = 2F Unexpected buffer overflow while entering a command into the input command buffer. EP2 = F5, EP3 = 09 Unexpected underflow while reading an address from the command restart buffer.

HEX	DEC	DESCRIPTION (Fatal Errors)
0B	11.	**No Service Routine for Level**
		A hardware interrupt has occurred which is assigned to a particular firmware interrupt priority level where no service routine is currently defined. This error is also generated if the DATA CARRIER DETECT line (of the RS-232-C interface) is "glitched" while the Plotter is transmitting.
0C	12.	**No Self-Interrupt Routine for Level**
		The firmware has requested an internal self interrupt for which no service routine is defined.
0D	13.	**MSync Address Not in First 256 Bytes of ROM**
		An illegal motion synchronization address was encountered.
0E	14.	**No Routine for Specified System Command**
		An internal Plotter command has been encountered for which a processing routine is not currently defined.
0F	15.	**ROM in Wrong Socket**
		A non-relocatable ROM was found (during powerup) at an address which does not agree with the location-dependant firmware contained within the ROM. Error Parameters are: EP1 = Address high byte where the ROM firmware should begin. EP2 = Address high byte of a location less than 256 bytes from where the ROM firmware actually begins. EP3 = Address low byte of a location less than 256 bytes from where the ROM firmware actually begins.
10	16.	**Motion Buffer Overrun**
		Unexpected internal motion task buffer full when entering an internal motion task.

HEX	DEC	DESCRIPTION (Fatal Errors)
11	17.	**Unexpected Buffer Overflow/Underflow**
		This error indicates that a buffer overflow/underflow condition has been detected. This fatal error occurs if firmware determines that space exists, but later an overflow occurs. The error also occurs when firmware determines that data exists, but later an underflow occurs. The following groups of errors list activities when an Unexpected Buffer Overlow/Underflow error could be detected: ● ERRORS RELATED TO OPTION 31 OPERATION EP1 = Command code of Option 31 related command being processed. EP2 = 58, EP3 = 74 Memory overflow while trying to enter a command byte into the current macro buffer. EP2 = 5D, EP3 = 72 Memory overflow while trying to: — Create a one block macro buffer for a new macro. — Enter the macro number into a new macro buffer. — Clear the header in a new macro buffer. — Enter a new macro list termination byte. EP2 = 5E, EP3 = 70 Memory overflow while trying to record the macro in the macro list. EP2 = 5F, EP3 = E9 Memory overflow while trying to enter a macro list termination byte after a macro delete. EP4 or EP5 The number of the current macro being processed.
		● OUTPUT RELATED ERRORS EP2 = 7E, EP3 = 48 The output data buffer overflowed during loading of output block. EP1 contains the number of bytes previously written.
		● ERRORS RELATED TO OPTION 32 OPERATION EP1 = Command Code of current Option 32 command being processed. EP2 = 53, EP3 = 8A Memory overflow while trying to record an Option 32 font in the font list. EP2 = 52, EP3 = 95 Memory overflow while trying to enter a new Option 32 font list termination byte. EP2 = 53, EP3 = BC Memory overflow while trying to enter an Option 32 character definition byte into the current font buffer. EP2 = 55, EP3 = 2E Memory overflow while trying to enter a new Option 32 font list termination byte. EP2 = 55, EP3 = 4D Memory overflow while trying to create a one block Option 32 font buffer for a new font. EP2 = 56, EP3 = 09 Memory overflow while initializing the header for a new Option 32 font buffer. EP2 = 57, EP3 = B4 Memory overflow while trying to enter an Option 32 font list termination byte after a character delete. EP3 or EP5 = Current Option 32 font number. EP4 = Current Option 32 character.
		● TRANSFORM RELATED ERRORS EP2 = B9, EP3 = E4 Unexpected memory underflow while checking for space to push transform bytes on the transform stack.

Appendix K

GLOSSARY

address character
(Serial) The second character of most commands. Can be set to ASCII characters A through H.

ADUs
Addressable Device Units. One of three choices of Device Units.

argument
A parameter associated with a command (e.g., for the angle of rotation or for the X-coordinate of a Move.

argument separators
Special character sequences that define the end of one argument and the beginning of the next. A space or comma character is most often used.

ASCII character
One of the 128 characters defined by the ASCII (American National Standard Code for Information Interchange) standard.

ASCII Decimal Equivalent (ADE)
The decimal value associated with ASCII characters (e.g., 32 for SP, 65 for A).

attention character
(Serial) The first character of most commands. Can be set to ESC, !, ∧ , or SYN using the Parameter Entry Card.

Attention Action commands
(Serial) A command consisting of a series of two characters beginning with the attention character (AT).

auto move-to-home
The process of automatically performing a move to the home position whenever a PRINT command follows power-up, media change, or a front panel Locate function without an intermediate Move or Draw.

baud rate
(Serial) See *receive baud rate* or *transmit baud rate*.

bit
A binary digit having two possible values: 0 or 1.
1 = true or set while 0 = false or cleared.

block
(Serial) Formatted group of commands or responses.

Block mode
(Serial) A procedure for block Serial communications.

block size
(Serial) The size (in characters) of a Block mode block.

byte
A group of eight bits.

character argument
A command argument consisting of a single ASCII character.

character x-extent
The left and right coordinates that specify the left and right edges of a printing character.

checksum
(Serial) Part of a Block mode block. Determined by summing the contents of the block used to detect transmission errors.

clipping
The process by which the portion of a plot that falls outside of a specified boundary is not plotted.

command code
One or two ASCII characters that uniquely identify a command.

command syntax
The conventions for a particular application (such as Serial, GPIB, or 4050 Series BASIC) that defines how to form a command.

command terminator
A character that marks the end of the command. See *discarded terminator* and *reprocessed terminator*.

Command/Response Format
The selection made on the Initial Command/Response Format line of the Parameter Entry Card that configures either the Serial or GPIB interface to accept commands and send responses in a specified format.

coordinate
The X-axis and Y-axis values used to describe a point.

crosshair cursor
The plastic sight located on the pen carriage.

DAB command
(GPIB) A command consisting of data bytes from a controller that contains the command code and arguments.

delimited string
A sequence of ASCII characters beginning and ending with a closing single quotation mark ('), an opening double quotation mark ("), or a slant character followed by a character ADE value. Longer strings can be formed from strings of this form.

device coordinate system
A Cartesian coordinate system for the Page. The origin is the lower-left Page corner. (See *Device Units* for additional details.)

device status
See *status information.*

Device Units
One of three sets of numerical values assigned to the Initial Page for the Device Coordinate System. The default selection will be ADUs for a Serial interface and GDUs for a GPIB interface. The range of values will be 4096 along the long axes for ADUs, 100 along the short axis for GDUs, or directly in millimeters for mms. Both axes will have physically identical unit size. See Section 1 for more information.

digitizing
The process of locating and transmitting the coordinate information to the host.

discarded terminator
A terminator that explicitly delimits the command. Once recognized, these terminators are discarded without further interpretation.

downloaded character
(Requires Option 32) A definition of how a specific alphanumeric character or symbol is to be printed.

Draw
The Plotter draws a line to a specified position.

EOI (End Or Identify)
(GPIB) This bus message is sent with the last data byte of a message, indicating that the message is complete and terminating the last command in the message.

error codes
Occurrence of errors is reported by a flashing or steady Reset light and by sounding the bell. Lights on the Parameter Entry Module (with the Error Data line selected) can be interpreted as an error code to determine the source of the error. Refer to Appendix J for more information.

escape sequence
(Serial) Same as an Attention Action command with the specific attention character of ^{E}c.

fatal error
A detected error serious enough that all plotting is halted. The Reset light will be on steady. (See *nonfatal error.*)

font
A set of one or more character definitions for print actions.

form length
(Requires Option 36) The length of media advanced during a Media Change function.

full duplex
Allows communication in two directions at the same time.

GDUs
Graphic Device Units. One of three choices of Device Units.

GPIB
Abbreviation for General Purpose Interface Bus. Used in reference to interface or interface-dependent characteristics.

home position
A point located one character height below the upper-left corner of the Viewport along a line perpendicular to the current printing position.

Initial Page
One of several predefined Pages (including the user-defined selection) that is established by Parameter Entry Card selections. See Section 1 for additional information.

Initial Viewport
The Viewport that corresponds to the Initial Aspect Ratio line selection on the Parameter Entry Card. See Section 1 for more information.

Initial Window
A Window that corresponds to the Initial Viewport. An Initial Window exists for each of the three Device Units selections of ADUs, GDUs, and mm's. See Section 1 for more information.

Initialize function
This front panel function resets the active interface after communications with the host or controller have been lost. However, it does not modify any of the current programmable interface parameters (such as prompt character, signature character, turnaround-delay time, etc.).

input buffer
Refers to the memory of the Plotter that permits incoming commands to be temporarily stored until the Plotter can generate the corresponding Plot actions.

line type
Refers to either solid, fixed dash-pattern-length, variable dash-pattern-length, or point (the plotting of endpoints for each line segment) methods of drawing lines with the Plotter.

listener
(GPIB) A device that receives commands being transmitted over the GPIB interface.

loop-thru
(Serial) A scheme where the Plotter is inserted into the normal communication path between two pieces of equipment, such as a computer or modem and a computer terminal.

macro
(Requires Option 31) A group of Plotter commands that can be stored in the Plotter and later executed by a single command.

media
The paper, mylar, etc. upon which a plot is drawn.

MLA (My Listen Address)
(GPIB) The device address that enables the Plotter to be a listener.

Move
The pen carriage is moved to a specified position (without the pen contacting the media).

MSA (My Secondary Address)
(4050 Series BASIC) A method of specifying command codes with a nonstandard use of the GPIB MSA message unique to the 4050 Series Graphic System.

MTA (My Talk Address)
(GPIB) The device address that enables the Plotter to be a talker.

nonfatal error
A detected error that is reported (Reset light flashes), but does not halt plotting operation.

NULL arguments
An argument either of zero value (for NULL numeric arguments) or consisting of the single ASCII character NU (for a NULL string argument). NULL argument values are established by simply not specifying the argument and including only the argument separator (or terminator).

numeric argument
A numeric argument expressed in one of three standard forms: integer, floating point, or scientific notation (E format).

output block
A portion of an output response containing one or three values.

output message
A transmission from the Plotter consisting of one or more output blocks in response to a specific host command.

output responses
Information sent by the Plotter in response to commands.

output terminator
(Serial) A character sent following an output message transmitted by the Plotter.

padding
(Serial) The process of intentionally limiting the average rate of information transfer by inserting extra characters that will be ignored.

Parameter Entry Card
A directory (part of the Parameter Entry Module) that assists the user in selecting the Plotter's operating parameters.

Parameter Entry Module
That portion of the Plotter that includes the Parameter Entry Card, the INIT light, and the lighted switches.

Page
A rectangular area on the platen that can be used for plotting.

pen carriage
The part of the Plotter that includes the crosshair cursor, Pen Holder 1, and Pen Holder 2.

platen
The flat surface upon which the media is placed.

platen clipping boundary
The border outlining the maximum area on the platen that is available for plotting.

plot
Refers to the composite line and character information drawn on the media.

plot time
See *accumulated plot time*.

Point Plot mode
This mode saves the current line type and then establishes *point* as the current line type (causing a dot to be placed at each specified plot coordinate). The next time Graph mode is entered, the previously saved line type is restored.

primary address
(GPIB) A number assigned to each device on the GPIB bus. The Plotter's primary address is selected by the GPIB Device Address line on the Parameter Entry Card.

prompt string
(Serial) A sequence of ASCII characters that must be received by the Plotter before any output is transmitted to the host.

receive baud rate
(Serial) The rate at which information can be received by the Plotter.

reprocessed terminator
These terminators end a previous command, but are also used as part of the next command (such as when the $^{A}{}_{T}$ character for a command terminates the previous command).

response block
A grouping of information, ranging from 7 to 33 characters in length, that consists of two data values and a TAG value.

RS-232-C
(Serial) An industry standard for the interface between data terminal equipment and data communication equipment employing serial binary data interchange.

secondary address
(GPIB) See *MSA*.

Serial interface
Refers to the RS-232-C interface or characteristics associated with that interface. The interface transmits and receives ASCII characters composed of individual bits transmitted one bit at a time (i.e., serially).

signature character
(Serial) A unique character optionally sent as the first character of each Plotter output block. This character can be used to identify a specific Plotter when more than one Plotter is connected to the host.

status information
All or part of the internal Plotter information (such as out-of-media or active pen selection) that the Plotter can send on request to describe its current state.

string arguments
A sequence of one or more ASCII characters. See *delimited string* and *undelimited string*.

Style I commands
Unique Plotter commands that use ASCII decimal coding.

Style II commands
Plotter commands that use binary data coding. Used with TEKTRONIX 4010 Series terminals.

syntax
See *command syntax*.

TAG value
A third numeric value sent with response values in most output blocks.

talker
(GPIB) A device that transmits information on the GPIB bus to other specified devices.

terminator
See *command terminator*.

transform commands
Commands that modify incoming data to the Plotter.

transmit baud rate
(Serial) The rate at which information is transmitted by the Plotter.

turnaround delay
A function that delays Plotter transmissions for a set time after host/controller information has been received by the Plotter.

undelimited string
A sequence of ASCII characters where the first character cannot be a slant, apostrophe, or quote mark; the string cannot begin with or include a comma, $^{S}{}_{P}$, $^{D}{}_{1}$, $^{D}{}_{3}$, $^{N}{}_{U}$, $^{S}{}_{Y}$, $^{D}{}_{L}$ (if DEL IGNORE is selected), $^{C}{}_{R}$, $^{A}{}_{T}$, or semicolon (if Command/Response Formats 1 and 2). Command/Response Formats 3 and 4 limit the string to a single character.

Viewport
The rectangular portion of the Page onto which the plot is drawn.

Window
The rectangular portion of the World Space that is projected onto the Viewport.

World Coordinate System
A two-dimensional Cartesian coordinate system for World Space.

World Space
The space associated with the object or picture. World Space can include graphs, maps, mechanical drawings, or other pictures.

World Units
Units for the World Coordinate system.

Appendix L

COMMAND INDEX

INDEX