TEK PROGRAMMER'S REFERENCE MANUAL

Part No. 070-4164-00
Product Group 12

4662

INTERACTIVE DIGITAL PLOTTER

Please Check for
CHANGE INFORMATION
at the Rear of this Manual

First Printing APR 1982

Tektronix®
COMMITTED TO EXCELLENCE

MANUAL REVISION STATUS

PRODUCT: 4662 Interactive Digital Plotter

This manual supports the following versions of this product: Serial Numbers B010100 and up.

REV DATE	DESCRIPTION
APR 1982	Original Issue

CONTENTS

Section 4 (cont) **Page**

Section 5 **GPIB INTERFACE CONCEPTS**

Section 6 **INTRODUCTION TO GPIB COMMANDS AND RESPONSES**

Section 7 **GPIB COMMAND DESCRIPTIONS** **Page**

ILLUSTRATIONS

TABLES

PREFACE

ABOUT THIS MANUAL

This manual is for the programmer who is writing a package of computer subroutines that communicate with the TEKTRONIX 4662 Interactive Digital Plotter (Figure 1-1). This manual describes the plotter commands and tells how to issue them by entering sequences of ASCII characters from 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 Serial (RS-232-C) communications, General Purpose Interface Bus (GPIB) communications, or TEKTRONIX 4050 Series BASIC commands. This manual is written so that individuals with limited computer experience can learn to communicate with the plotter.

This manual is organized in relation to the two interfaces present in the plotter. These interfaces are the Serial (RS-232-C) and the GPIB Interfaces. For each interface, this manual has three sections: a section describing interface concepts (this section includes important programming information, which you use to send commands to the plotter), a section introducing the commands used by that interface, and a section describing all the interface commands. This manual contains the following sections:

- Section 1 (Introduction) describes the basic plotter graphing system.
- Section 2 (Serial Interface Concepts) describes programming information when you are using the Serial Interface.
- Section 3 (Introduction to Serial Commands and Responses) describes the types of Serial commands used and the notation used in the command descriptions in Section 4.
- Section 4 (Serial Command Descriptions) describes each Serial command used by the plotter.
- Section 5 (GPIB Interface Concepts) describes programming information when using the GPIB Interface.
- Section 6 (Introduction to GPIB Commands and Responses) describes the types of GPIB commands used and the notation used in the command descriptions in Section 7.
- Section 7 (GPIB Command Descriptions) describes each GPIB and 4050 Series Graphic System command used by the plotter.

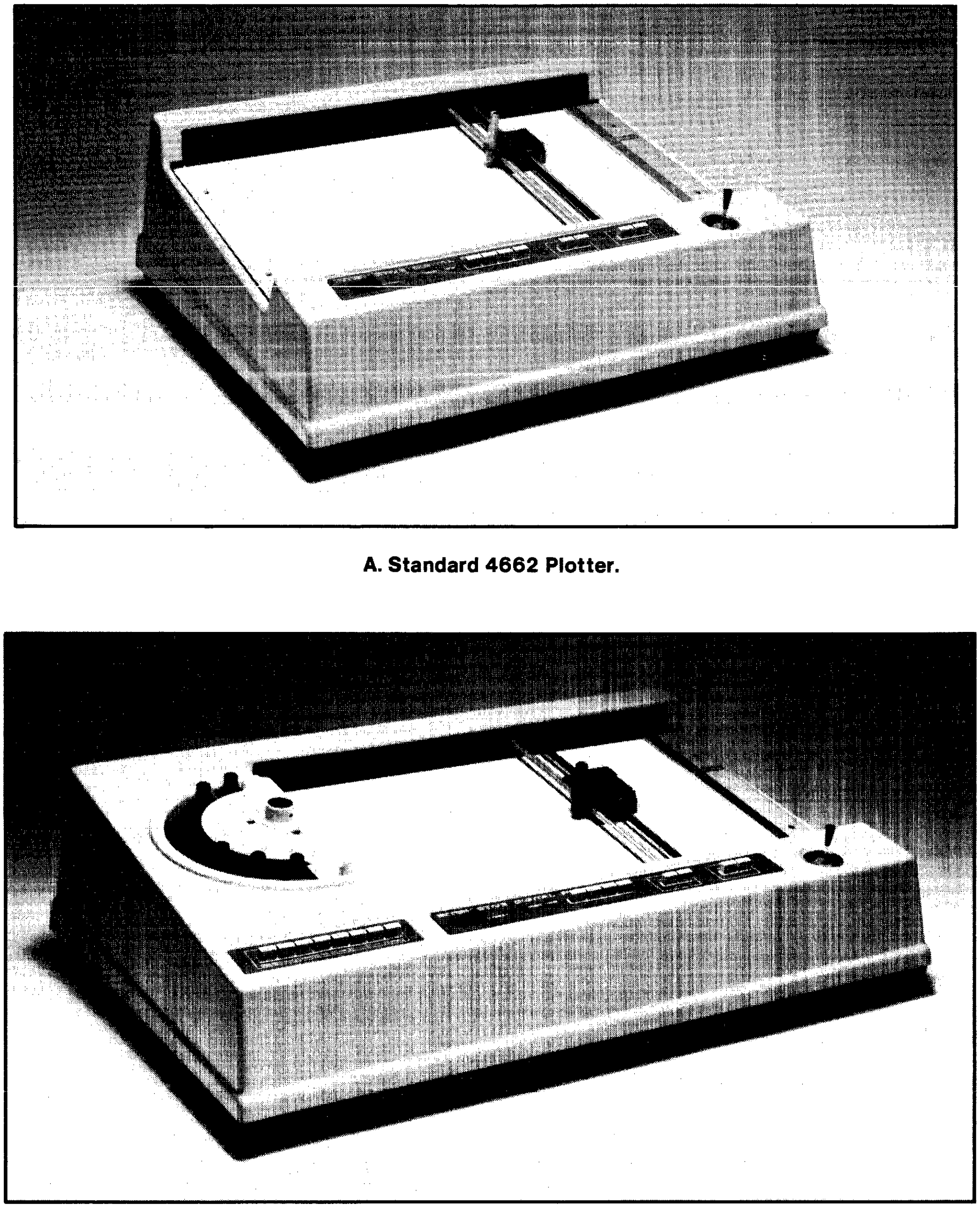
While reading this manual, the user may find it helpful to refer to these appendices and the index, which is located after the appendices:

- Serial Command Summary — Appendix A
- GPIB Command Summary — Appendix B
- ASCII Code Chart — Appendix C
- Serial Interface Connectors — Appendix D
- Area Fill Patterns — Appendix E
- Coordinate Conversion Chart — Appendix F
- Basic Program for Coordinate Conversion — Appendix G
- Example Plots — Appendix H
- Glossary — Appendix I

OTHER 4662 MANUALS

The following related manuals are available:

- *4662 Interactive Digital Plotter Operator's Manual.*
- *4662 Interactive Digital Plotter Reference Guide.*
- *4662 Interactive Digital Plotter Service Manual.*



A. Standard 4662 Plotter.

B. 4662 Plotter Equipped with Option 31.

4165-1

Figure 1-1. 4662 Interactive Digital Plotter.

Section 1

INTRODUCTION

This section describes some of the basic concepts and terminology fundamental to 4662 plotter graphics. These terms and concepts are used throughout this manual and should be understood before proceeding.

This section includes descriptions of:

- The Plotter
- Pen
- Platen
- Page
- Plotter Units (and Coordinate System)
- Data Resolution
- Error Reporting
- Page Scaling

Concepts for specific Serial or GPIB commands are included in Sections 2 and 5, *Serial Interface Concepts* and *GPIB Interface Concepts.*

THE PLOTTER

The primary function of the plotter is to transfer information to media such as paper or polyester film. This transfer is accomplished by commanding the pen on the plotter to draw lines and print alphanumeric characters.

The commands associated with drawing lines are called *graphics commands*. These include MOVE and DRAW:

- MOVE causes the pen to move to a specified position without drawing a line (with the pen up).
- DRAW causes the pen to draw a line to a specified position (like a Move but with the pen down).

Commands to print alphanumeric characters are called *alpha commands* and are discussed in Sections 4 and 7.

The plotter performs an additional function called *digitizing*. This function transfers pen coordinates back to a host computer or terminal.

PEN

Since the pen may be physically positioned over the media at any point specified by plotting coordinates, this manual uses the term *pen position* to mean the location of the writing tip of the pen in the pen carriage.

PLATEN

The *platen* is the large flat surface of the 4662 plotter on which the media (paper, film, etc.) is placed.

PAGE

The *page* is a rectangular area on the platen which is used for plotting. The page provides a plotting area (or viewing area) similar to the full screen on a terminal. The default page is 15x10 inches (381x254 mm). Figure 1-2 shows the relationship of the media, platen, and page.

The operator may modify the size and location of the page using front-panel SET LOWER LEFT and SET UPPER RIGHT switches (see the *4662 Interactive Digital Plotter Operator's Manual*). Pages can not be set by the host.

Refer to the *4662 Interactive Digital Plotter Operator's Manual* for the ranges of page sizes. The plotter will not draw any portion of a vector which has an end-point outside the page boundary.

The *4662 Interactive Digital Plotter Operator's Manual* describes special effects associated with setting pages from the front panel switches. These effects include upside-down and mirror images, which are simply page reversals. (The SET LOWER LEFT and SET 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 *4662 Interactive Digital Plotter Operator's Manual.)*

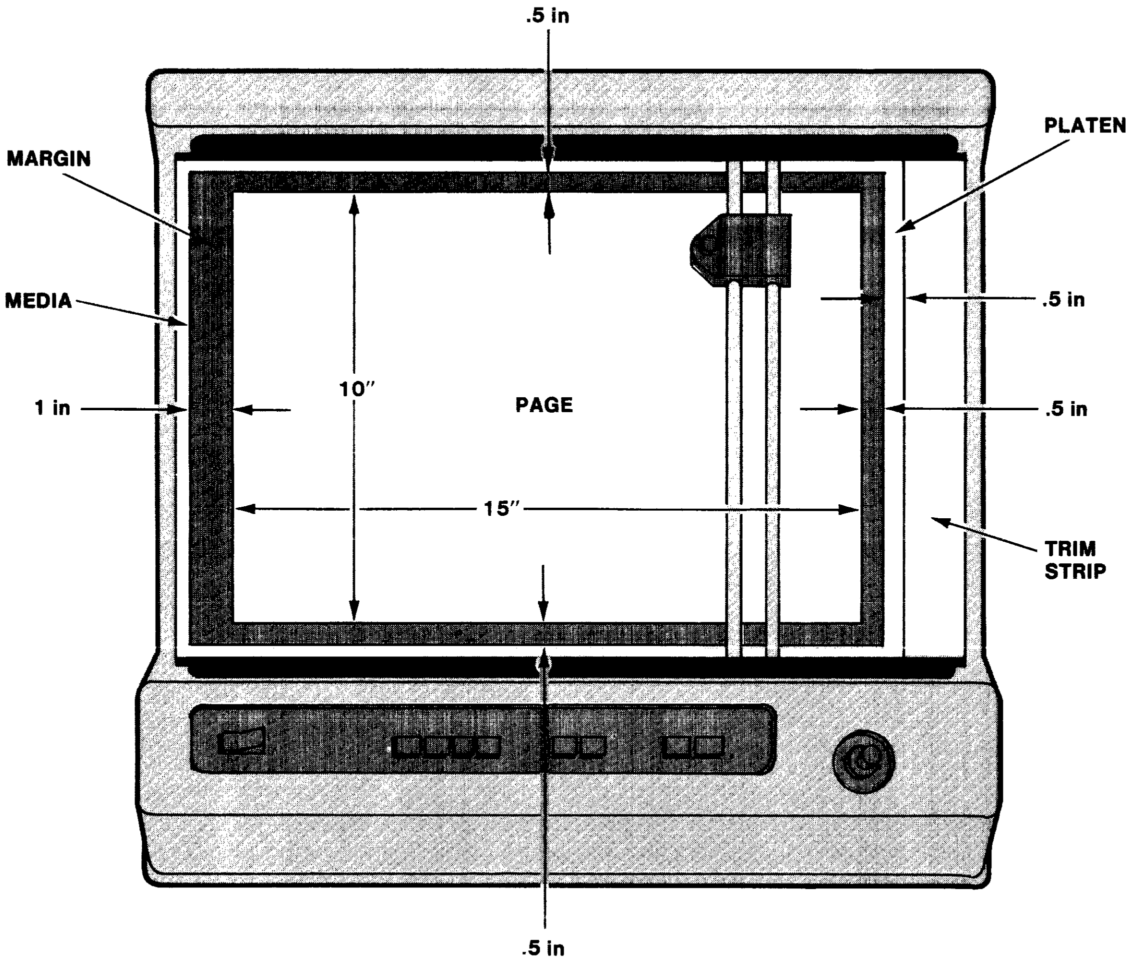


Figure 1-2. Media, Platen, and Page.

PLOTTER UNITS

The plotter selects one of two plotter units at power-up:

- Addressable Device Units (ADUs) when operating with the Serial Interface.
- Graphics Device Units (GDUs) when operating with the GPIB interface.

ADUs and GDUs correspond to units associated with screens of the TEKTRONIX 4010 Series Computer Display Terminals and the TEKTRONIX 4050 Series Graphics Systems, respectively.

Plotter Coordinate System and Plotter Units

The plotter coordinate system is a Cartesian coordinate system for the page. The 0,0 location or origin of the coordinate system is the page's lower-left corner. The physical location of the lower-left corner may be anywhere as described earlier (i.e., page reversal or mirror-image situations).

The page ranges when operating with the Serial Interface are as follows:

- Standard (non-Copy) mode:
  0 to 4096 ADUs horizontally and 0 to 2731 ADUs vertically.
- Copy mode:
  0 to 4096 ADUs horizontally and 0 to 3124 ADUs vertically. The right-most 13% of the page is not addressable. (See the *4662 Interactive Digital Plotter Operator's Manual* for a description of Copy mode.)
- Standard (non-Copy) mode with Modified ADUs:
  0 to 3000 ADUs horizontally and 0 to 2000 ADUs vertically. (See Multiple Pen, Option 31, commands for more information on Modified ADUs.)
- Copy mode with Modified ADUs:
  0 to 3000 ADUs horizontally and 0 to 2288 ADUs vertically. (See *4662 Interactive Digital Plotter Operator's Manual* for description of Copy mode and Multiple Pen, Option 31, commands in this manual for more information on Modified ADUs.)

The page range when operating with the GPIB Interface is 0 to 150 GDUs horizontally and 0 to 100 GDUs vertically.

These ranges are not changed if the front-panel Set Page operations change the page size or position. However, the physical length of each plotter unit is changed and/or made unequal for each axis if unequal scaling is introduced when you modify page sizes.

NOTE

The SELECT PLOTTER UNITS command (available in SERIAL ONLY and units equipped with Firmware Version 5 or higher) allows a "modified" ADU in which the plotter's coordinate system ranges from 0 to 3000 ADUs along the X-axis and from 0 to 2000 ADUs along the Y-axis. See Serial Multiple Pen (Option 31) commands.

DATA RESOLUTION

The maximum resolution between points on the plotter is .005 inches (0.127 mm). However, the plotter accepts arguments with greater resolution, but the positioning of the pen and the reporting of coordinate positions are limited to the plotter's maximum resolution.

ERROR REPORTING

To report an error, the 4662 Plotter displays an ERROR light and sounds the bell.

PAGE SCALING

For some applications the physical length of the ADU or GDU must be known. This can be found by measuring the length of each axis and dividing that length by the number of plotter units. An example is provided here for determining page scaling.

SERIAL INTERFACE

The default non-Copy mode page has 4096 ADUs along its X-axis of 15 inches (381 mm). Therefore the page scaling is:

$$\frac{15 \text{ inches}}{4096 \text{ ADUs}} = 0.0036621 \text{ inches/ADU or } 273.07 \text{ ADUs/inch}$$

or

$$\frac{381 \text{ mm}}{4096 \text{ ADUs}} = 0.0930 \text{ mm/ADU or } 10.751 \text{ ADUs/mm}$$

GPIB INTERFACE

The default page has 150 GDUs along its X-axis of 15 inches (381 mm). Therefore, the page scaling is:

$$\frac{15 \text{ inches}}{150 \text{ GDUs}} = 0.10 \text{ inches/GDU or } 10 \text{ GDUs/inch}$$

or

$$\frac{381 \text{ mm}}{150 \text{ GDUs}} = 2.54 \text{ mm/GDU or } 0.3937 \text{ GDUs/mm}$$

Scaling for page sizes set by the front-panel switches can be calculated in a similar manner.

Section 2

SERIAL INTERFACE CONCEPTS

INTRODUCTION

This section describes how to install the plotter into a serial communications system and the programming considerations (type of serial communications, methods, buffering, etc.) of host-plotter serial communications. This information is especially helpful when designing a host and serial communication system to direct the plotter. This section does not cover the actual plotter commands or how they are represented in this manual. That information is found in Sections 3 and 4. If a GPIB system is used, refer to Section 5 for similar GPIB programming considerations.

INSTALLING THE PLOTTER INTO A SYSTEM

CONNECTING THE PLOTTER TO OTHER DEVICES

The flexibility provided by the front-panel LOCAL switch allows the plotter to be connected with other devices in a variety of ways. The following summarizes four ways to interconnect the plotter with a host computer, a terminal, or a storage device:

- The plotter can be connected as a peripheral to the host.
- The plotter can be connected in a *loop-through* configuration to a terminal (where the host talks through the plotter to the terminal).
- The plotter can be connected as a peripheral to a terminal.
- The plotter can operate in conjunction with an "off-line" storage device (for example, where a host does not exist in the system).

Details on these four configurations follow.

PLOTTER CONNECTED AS A HOST PERIPHERAL

In this configuration, the host views the plotter as an independent peripheral device. A host peripheral configuration is advantageous when the plotter is a resource available to users of the host-system. The plotter may be either located near the host or located at a remote location (possibly with a terminal).

Figure 2-1A illustrates the simplest version of this configuration, where the plotter is connected to the host through a dedicated input/output (I/O) port. (Using a dedicated I/O port is the simplest configuration for the host software to support.) When the plotter is located near the host, modems are not required.

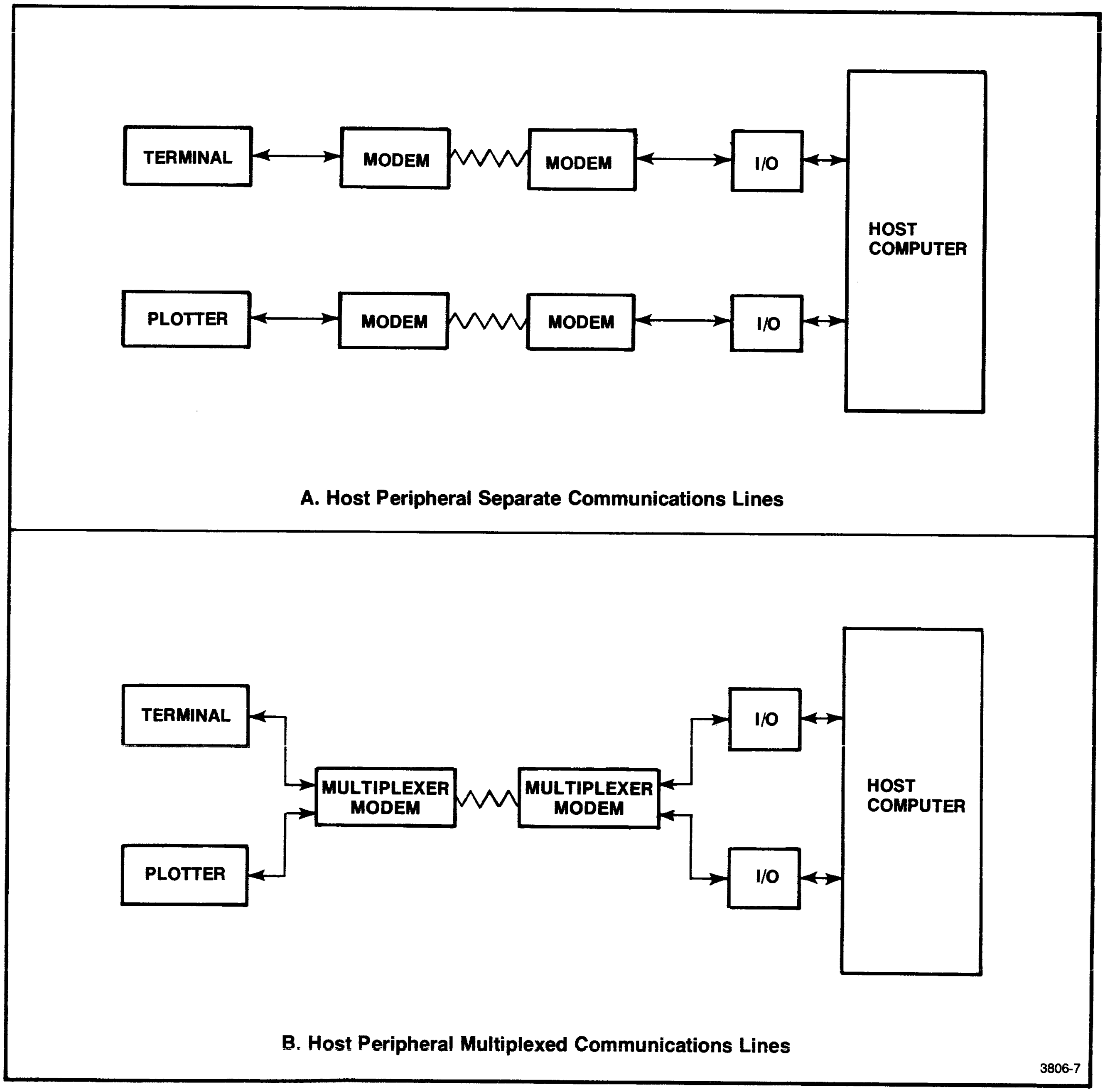


Figure 2-1. Host-Peripheral Configurations.

Figure 2-1B 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 2-1A.

PLOTTER CONNECTED IN A LOOP-THROUGH CONFIGURATION

In Figure 2-2, the plotter and the terminal are connected in a "loop-through" configuration and they share a common communications line to the host. This configuration is appropriate when the host, while under interactive control of the terminal, generates graphics to be drawn on the plotter. Software support for this configuration is slightly more complex, because 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 PLOTTER ON and PLOTTER OFF commands. When the host sends a PLOTTER ON command to the plotter (enabling the plotter to recognize subsequent commands), the plotter interrupts the command flow to the terminal (when TERMINAL MUTE ENABLE is selected on the four rear-panel switches).

A subsequent PLOTTER OFF command disables the plotter and allows the host to access the terminal through the plotter again. Also, in this configuration, the plotter may be used as a terminal peripheral if the 4662 front-panel LOCAL switch is pressed down and locked. The loop-through configuration allows operation of the plotter with virtually any type of terminal having an RS-232-C-compatible Serial Interface.

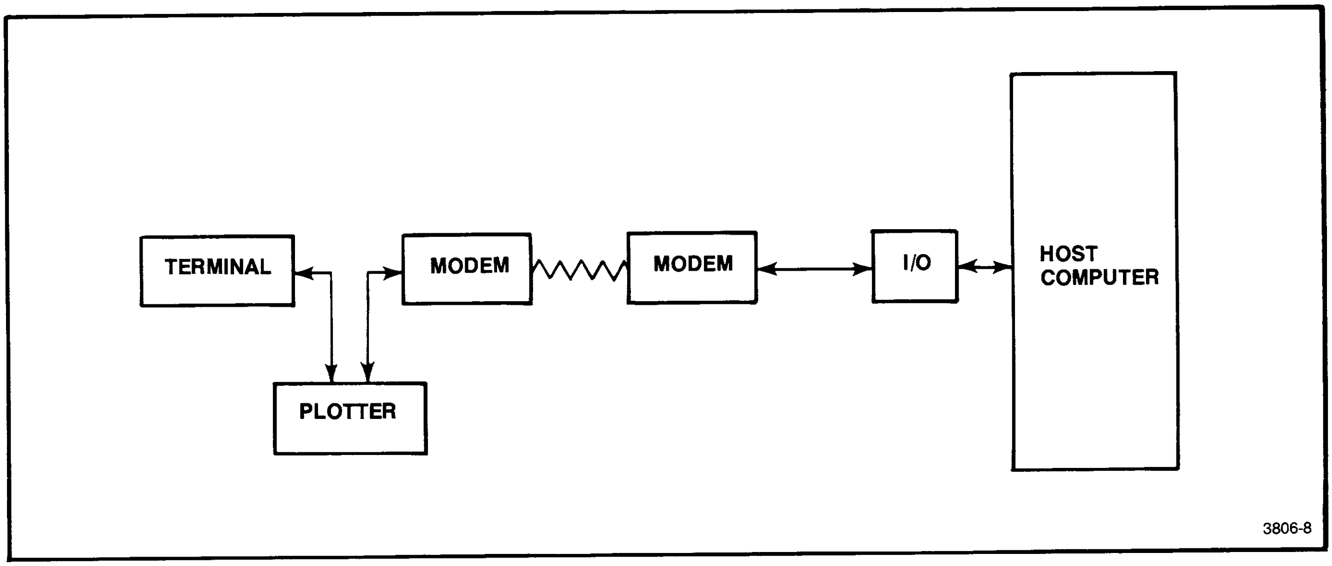


Figure 2-2. Plotter Connected in a Loop-Through Configuration.

PLOTTER CONNECTED AS A TERMINAL PERIPHERAL

The configuration shown in Figure 2-3 gives the terminal priority over the host computer in using the plotter as a peripheral. The terminal peripheral configuration is most appropriate when either the host or the terminal are generating graphics to be plotted. The terminal communicates with the plotter directly, while the host must pass data through the terminal to reach the plotter. Limiting host communication to only the terminal may simplify host support programs which deal with the terminal through the I/O port. Advantages of this configuration depend upon the terminal's capabilities. Refer to the terminal operator's manual. For example, some terminals can do background plotting from local data storage while continuously communicating with the host. Some possible configurations with Tektronix terminals are listed in Table 2-1, which shows which terminal interface options are required.

Table 2-1

OPTIONS REQUIRED WITH PLOTTER AS A TERMINAL PERIPHERAL

Tektronix Terminals	Interface Description	Terminal Option
4010, 4012 4014, 4015	With Dual Interface Capability	Option 36
4016	With Dual Interface Capability	Option 35
4110 Series	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, *GPIB Interface Concepts*, for GPIB information.

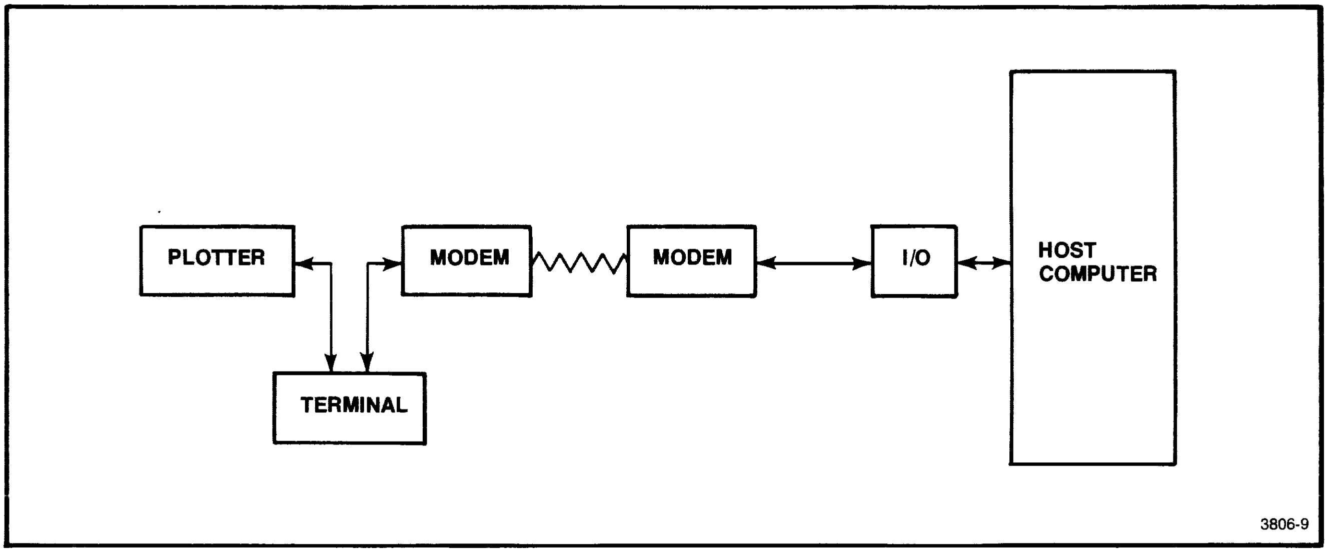


Figure 2-3. Plotter as a Terminal Peripheral.

OFF-LINE 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 2-4. This configuration permits data to be plotted without using a terminal or a host computer as a controller. The TEKTRONIX 4923 Digital Tape Drive Unit with Option 01 is an example of a data storage device with a Serial Interface.

Two systems are shown in Figure 2-5. These systems use a TEKTRONIX 4010 Series terminal with either a standard TEKTRONIX 4923 Digital Tape Drive Unit or a TEKTRONIX 4923 Digital Tape Drive Unit with Option 01. These configurations allow plot data to be recorded while "on-line" with the host computer and then later to be plotted during "off-line" time. The REMOTE and LOCAL switching functions on both the tape unit and the plotter direct the routing of the plot data.

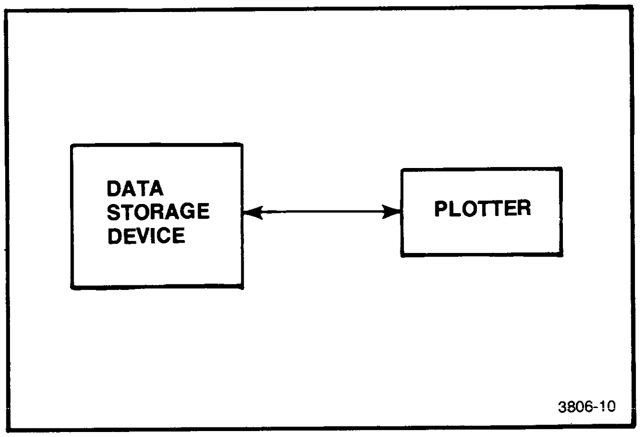


Figure 2-4. OFF-LINE Plotting Configuration.

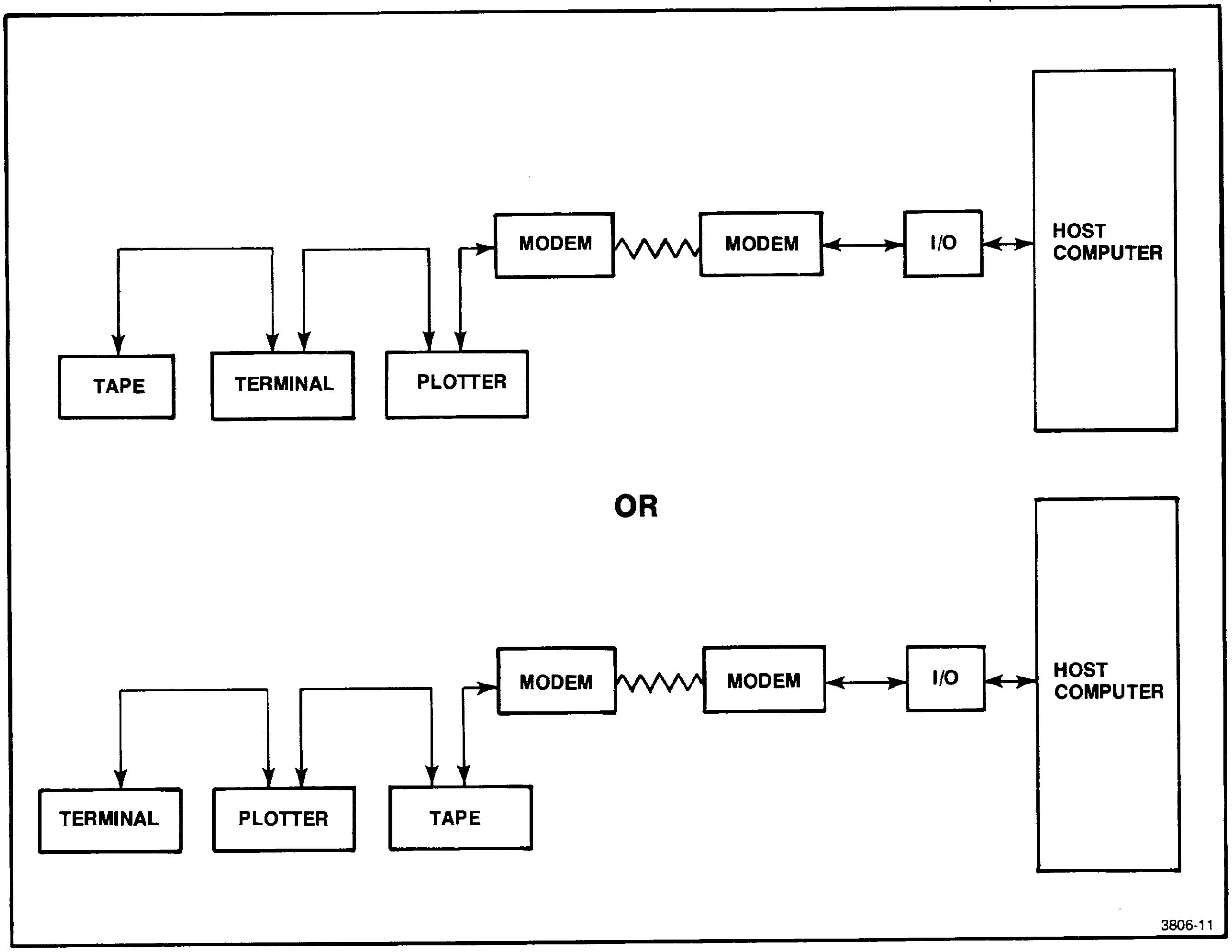


Figure 2-5. ON-LINE Recording and OFF-LINE Plotting.

MULTIPLE PLOTTERS

Several plotters may be installed in the same loop-through configuration in place of the single plotter previously described. All of the plotters can be set to the same device address and receive identical data in parallel or each one can use a different device address and receive data independently. Refer to *Communicating With Other Devices* later in this section for details about multiple plotter configurations.

COMMUNICATING WITH OTHER DEVICES

The following pages describe how command and response transmissions are sent between the plotter and other connected devices. This information is described under the following topics:

- Selecting a Communications Control Mode.
- Interface Switching Function.
- RS-232-C Lines for the Loop-Through Configuration.
- Device-to-Plotter Transmission.
- Plotter-to-Device Transmission.

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. Since the plotter is an electromechanical 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 called *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 discusses how to select the optimum method to manage communication between the Plotter and connected devices.

SELECTING A COMMUNICATIONS CONTROL MODE

These paragraphs summarize the advantages and limitations of each of the communication control modes which provide input rate control. This summary should help select the communications control mode most appropriate for a given application or communication environment.

There are three individual modes: Continuous mode, DC1/DC3 Flagging mode (Option 31 only), and Block mode. Information for each of these modes is presented under *Device-to-Plotter Transmission*, later in this section.

Information on selecting control methods for plotter transmission to connected devices is described under *Plotter-to-Device Transmission*, later in this section.

Continuous Mode

Continuous mode is the plotter's default mode on power-up if DC1/DC3 Flagging has not been selected by the four rear-panel switches. Input rate control is provided by selection of an appropriate receive baud rate (giving a simple low-performance control mode). Advantages and limitations are:

Advantages

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

Limitations

Input baud rate must satisfy limitations. For small plots (less than about 250 graphic commands using the standard plotter, or 1400, if Option 20 is installed), any baud rate may be used. For larger plots, the baud rate must be reduced (often to less than 300 baud). For some very large plots, it may be impossible to pick a baud rate that will allow the plot to be drawn without data loss.

Only parity error and data overflow detection and no correction of transmission errors is provided.

NOTE

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

DC1/DC3 Flagging Mode (Option 31 Equipped Plotters Only)

DC1/DC3 mode permits communication at high baud rates, but requires DC1/DC3 flagging support in the transmitting device's interface. This mode in the plotter is enabled by selecting DC1/DC3 Flagging on the four rear-panel switches.

Advantages or limitations are as follows:

Advantages

Any baud rate may be used for small or large plots.

No transmitting device support unique to the plotter is required.

Flagging may be done through a modem connection to the transmitting device.

Limitations

The transmitting device must be able to suspend transmission within 133 characters without loss of data upon receipt of an ASCII $^{D}{}_{3}$ (stop) character and the device must be able to restart transmission upon receipt of an ASCII $^{D}{}_{1}$ (start) character.

Plotter only detects parity errors, but can not correct transmission errors.

Block Mode

Block mode is a powerful but more complex control mode that provides the best performance in most 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 and limitations of Block mode are:

Advantages

Any baud rate may be used for small or large plots.

Parity and checksum transmission error detection is provided. Limited error correction is also provided by retransmitting portions of the data.

Limitations

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.

Off-line storage and later plotting are difficult because it requires intelligence in the storage device to recognize and later transmit using the plotter's Block mode protocol.

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, using the front-panel LOCAL switch, you can modify the connections between devices without having to physically relocate cables. In addition to the description of the LOCAL switch following, refer to *RS-232-C Lines in Loop-Through 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-through configuration and is described in respect to this configuration. The plotter's front-panel LOCAL switch provides two conditions for the interface switching function, as shown in Figures 2-6A and 2-6B. These two conditions are described next.

LOCAL Switch is Released (On-Line Remote)

With the LOCAL switch in the released or up position, the plotter functions like a host peripheral because the terminal and plotter communicate with the host (Figure 2-6A). The plotter ignores host commands to the terminal when the plotter has not been turned on with a PLOTTER 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 PLOTTER ON command and if TERMINAL MUTE ENABLE has been selected by the four rear-panel switches. 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 because the transmitted characters could become garbled.

LOCAL Switch is Locked Down (On-Line Local)

The plotter and terminal communicate while the host control is isolated (Figure 2-6B). This configuration switches the plotter from a host peripheral to a terminal peripheral.

NOTE

When the plotter's POWER switch is in the OFF position, or if the plotter's line cord is disconnected from the power source, the interface switching function goes into an off-line remote state, which allows the terminal and host to freely communicate (see Figure 2-6C).

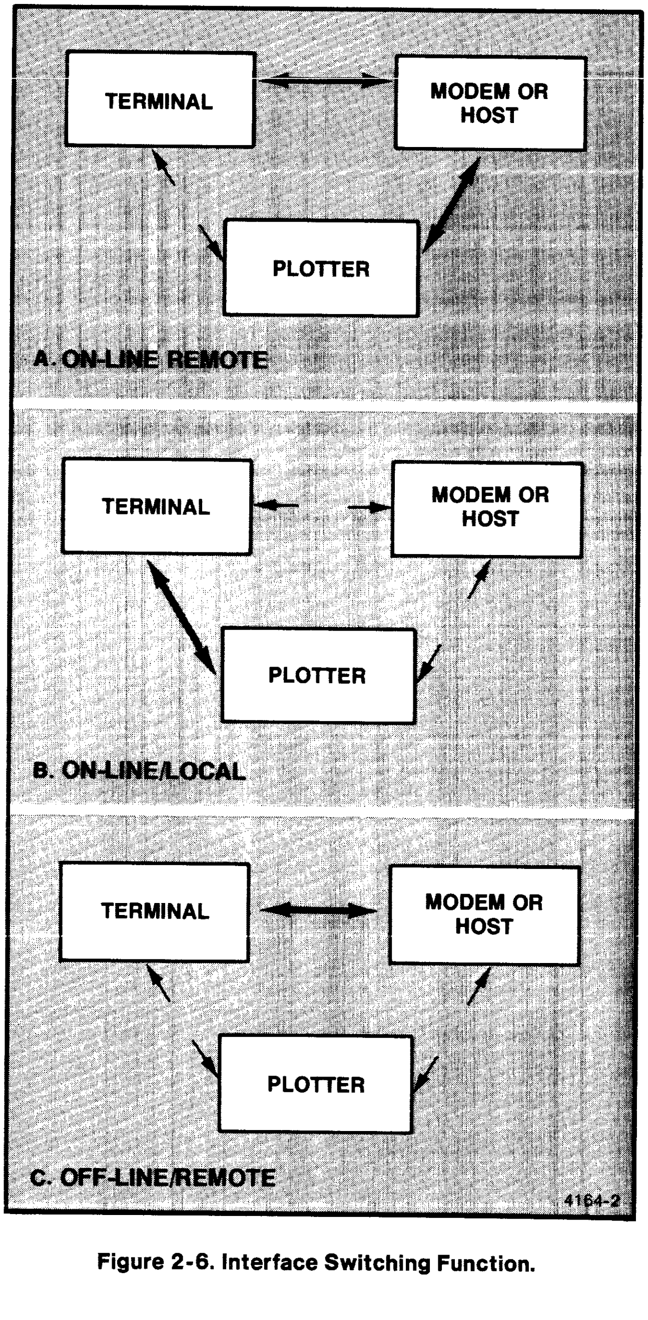


Figure 2-6. Interface Switching Function.

RS-232-C LINES IN LOOP-THROUGH CONFIGURATION

In the loop-through configuration, the interface switching function actually switches two RS-232-C signal lines: RECEIVE DATA (BB) and TRANSMIT DATA (BA). 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 or a terminal connected to the plotter.

Refer to Appendix D, *Serial Interface Connectors*, for definitions of all RS-232-C lines.

DEVICE-TO-PLOTTER TRANSMISSION

The following topics, *Input Buffering, Input Rate Controls*, and *Block Mode Communication*, describe the transmission of commands from the host computer (or data storage device) to the plotter. *Input Buffering* describes the amount of data (in the form of commands) that can be continuously transmitted to the plotter (at a given baud rate). *Input Rate Controls* describes the various transmission rate control modes used with the Input Buffering function to keep commands from being lost. The information under *Block Mode Communication* describes the various parameters and host computer functions involved in block data transmission. This information can help you choose 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 pages 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. Refer to Figure 2-7 during the following discussion. This figure illustrates a simplified model of plotter input buffering. All commands are received as a series of ASCII characters and stored in the Input Command Buffer. When the plotter is ready for another command, these stored commands are retrieved from the buffer and executed. Serial Interface commands (see Section 4 for a listing of these commands) are also decoded as they are placed in the buffer and all or some of their actions may take place immediately upon receipt.

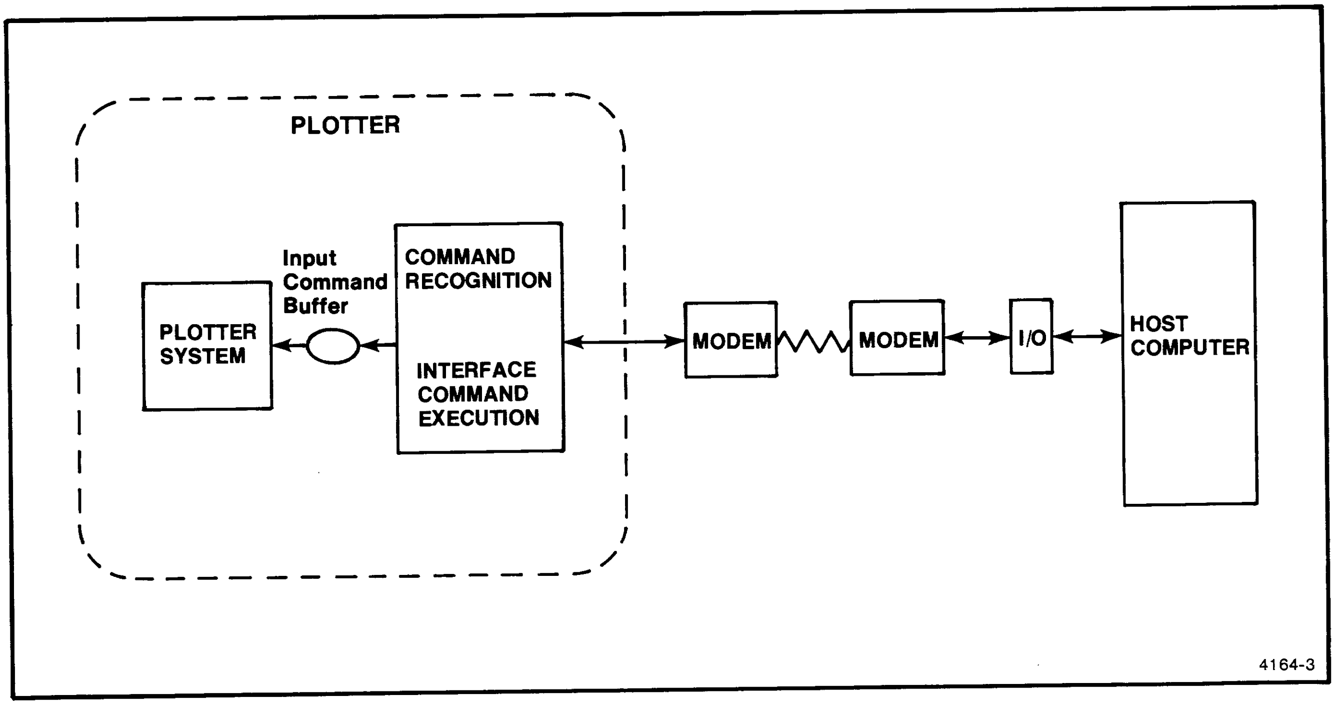


Figure 2-7. Input Buffering Mode.

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* (Section 3); refer to that description if you are not familiar with Style I and/or Style II formats.

Commands are stored in the command buffer in the same form that they are received, using one byte of buffer storage for each received character. This is even true for interface commands, which are still stored in the input buffer even though they have already been executed by the interface command processor. Any padding and terminator characters are also stored. The storage of these extra characters must be considered when using systems that automatically generate padding characters. Some hosts send padding characters after certain characters, such as CR, to allow the extra time necessary to execute these commands on teletype-like terminals. The only padding characters which are not stored are NU, SY, and DT. NU and SY are always discarded immediately upon receipt by the plotter and DT is discarded immediately if the rear-panel switches are set to the DELETE IGNORE position.

Memory Limits for the Input Command Buffer. Using the current rate control method, the plotter attempts to temporarily inhibit the transmitting device from sending more commands if memory space is not available. If further commands are received before memory is available, the commands are lost and an error is reported on the plotter. When stored data is subsequently processed, the allocated memory becomes available again for storing more commands.

NOTE

When a buffer overflow error occurs, several commands are usually lost.

The input command buffer memory of 2K bytes can store about 1400 characters (Option 20 expands this memory to 8K bytes, which can store about 7400 characters).

NOTE

The plotter's input buffer is cleared if the front-panel LOCAL or LOAD switch is pressed (see the 4662 Interactive Digital Plotter Operator's Manual).

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. Three input rate control modes may be used with the plotter: Continuous mode, DC1/DC3 Flagging mode (in Option 31 equipped plotters), 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 in *Selecting A Communications Control Mode*, located earlier in this section.

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

- After plotter power-up if DC1/DC3 Flagging mode has not been selected by the four rear-panel switches.
- After a PLOTTER ON command.

Rate control is accomplished by selecting a host-to-plotter input baud rate that transfers commands to the plotter at a rate that is equal to or less than the rate at which the commands may be processed or stored. For short plots (less than about 250 graphic commands) baud rates up to 1200 may be used. For large plots, baud rates of 300 or less are normally required. For very large plots, it may be impossible to select a baud rate low enough to prevent the plotter from losing data. 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. Therefore, Continuous mode may be used with data storage devices not having active control capabilities. Continuous mode is useful for plotting commands (such as 4010 Series Style II commands) that were generated for a terminal and contain no Block mode rate control commands for the plotter.

Parity Error detection and loss of data monitoring are the only transmission error functions available, and with Continuous 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 data, the reception of the Read Status back from the plotter informs the transmitting device that the input buffer is empty and another block of data may be sent. This allows operation at higher baud rates and plots of arbitrary length to be sent. It is up to the transmitting device to make sure that individual transmissions do not overflow the plotter's input buffer.

DC1/DC3 Flagging Mode (Option 31 Equipped Plotters). With DC1/DC3 Flagging mode, plotter sends an ASCII ␓ control character to the host to stop data transmission when the buffer becomes nearly full (there is space for 133 characters left in the buffer when the ␓ character is sent, allowing the host to "coast" to a stop). Later when the processor has processed stored commands so that space is available for 195 characters, the plotter sends an ASCII ␑ control character to the host to indicate it should again begin transmitting data until the plotter receives another ␓ character. In DC1/DC3 Flagging mode, the host input/output port must be able to stop data transmission without data loss when it receives ␓, and to restart when the ␑ character is received.

Because DC1/DC3 Flagging is done with ASCII characters rather than with control lines, DC1/DC3 Flagging can control devices attached to either the terminal or a modem connector. Some data storage peripherals can operate with DC1/DC3 Flagging. DC1/DC3 Flagging is useful for plotter commands (such as 4010 Series Style II commands) that were generated for a terminal and contain no Block mode rate control commands for the plotter.

DC1/DC3 Flagging mode is enabled by setting four rear-panel switches on plotters equipped with Option 31 (multiple pens).

NOTE

Error detection is the same as in Continuous mode.

NOTE

If a communications error causes a DC1/DC3 character to be garbled in transmission, the handshake involved with host computer causes computer communications to stop before data is lost. If this occurs, (1) transmission of commands from the the host computer must be manually interrupted, (2) the plotter's power must be recycled to reinitialize the plotter, and (3) the transmission of commands must be restarted.

Block Mode. Block mode may be used to provide both rate control and/or error detection and correction. Block mode can be used in any communications environment because no special hardware or communications support is required. However, there are several host computer functions required that need some software support. These functions and the methods for selecting values for their related parameters are discussed in the following topic, *Block Mode Communications.* The following provides a brief overview of Block mode operation.

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, surrounded by Block Start and Block End commands. After each block is transmitted, the host pauses for a response from the plotter. When the plotter receives a block, it checks the block's checksum to determine if the data was received correctly. If the received checksum value and the calculated value differ, a negative acknowledge character (an ASCII **I**) is sent to the host to cause the host to retransmit the entire block. If the received checksum and the calculated checksum agree, the input buffer is examined for space to store another block. If there is sufficient memory, then 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 there is insufficient space in the input buffer, the transmission of the acknowledge character is deferred until sufficient plotter commands are processed to make the required space available. This action matches the effective input baud rate to the rate at which plotter processing takes place, regardless of the selected input baud rate.

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 host computer causes computer communications to stop. If this occurs, (1) the command transmission from the host must be manually interrupted, (2) the plotter's power must be recycled to reinitialize the plotter, and (3) the transmission of commands must be restarted.

Block Mode Communication

Block mode communications requires the host computer to perform four related functions that include:

1. grouping the plotter commands into blocks,
2. computing a checksum for the block,
3. transmitting the block, and
4. processing the resultant block acknowledge character.

The following pages describe in detail how to determine what blocksize to use under various conditions. Also described are procedures for assembling blocks, computing the block checksum, and then using plotter commands to control Block mode communications. These procedures must be implemented in a user-written support package containing Block mode host software if Tektronix PLOT 10 host software support is not used. The available Tektronix host support that is used with the plotter is described briefly.

Maximum Blocksize. These paragraphs describe 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 which can be used if optimum values are not desired.

The maximum blocksize is specified to the plotter by a SET BLOCK SIZE command before Block mode is initiated. 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 ascertain that another block of this size can be stored in the plotter's input buffer.

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 characters contained in a block between the BLOCK START and BLOCK END commands should not be greater than this specified value minus the number of characters in the BLOCK START command and the BLOCK END command, the checksum (10) and the block end command terminator (normally 2). The ASCII characters — $^{N}{}_{L}$, $^{S}{}_{N}$, and $^{D}{}_{T}$ (DEL if the rear panel switches' DEL IGNORE selection is active) do not need to be included in the count.

The block size should be small enough to allow room for at least two blocks in the plotter's buffer. This provides a continuous flow of commands to the plotter.

A maximum blocksize of 125 characters works well when communicating with a standard plotter (with 2K bytes of memory) under most circumstances and allows up to 113 data characters to be sent in each block from most hosts.

NOTE

Using very large blocksizes can prevent any communications overlap with plotter command execution. 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 and reduce the possibility of an error in any one block.

CAUTION

Use of block sizes larger than 125 characters may cause incompatibility with plotters with a firmware level less than four.

The requirements for maximum blocksize can be summarized as follows (for more restrictive operation, use a smaller blocksize):

- Standard maximum blocksize is 125 characters or less.
- The number of data characters in a block is less than or equal to the blocksize — 12 in most systems.

Block Contents. Blocks may include any plotter command except Serial Interface commands (see Section 4) and the DEVICE RESET command. The Serial Interface commands, used to establish communication parameter values, should be transmitted in Continuous mode before Block mode is initiated.

Only complete Style I commands and Attention Action sequences must be included in a block (they must never be split between two blocks). (For example, BLOCK START or BLOCK END commands cannot appear within other Style I commands). Style II commands may include Style I commands and can be split between two blocks (although to ensure future product compatibility, this practice is not recommended). Refer to *Introduction to Serial Commands and Responses* (Section 3) 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 the ASCII decimal equivalent (ADE) value of each 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 the **(** in the BLOCK START command to the **)** in the BLOCK END command, as shown in Figure 2-8. To arrive at the checksum shown in Figure 2-8, add the ADEs of characters **(**, **A**, **B**, **C**, ^{E}C, **A**, and **)**:

```
   40
   65
   66
   67
   27
   65
 + 41
 = 371
```

See Appendix C for ASCII chart showing decimal values.

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

- The ^{N}L character.
- The ^{S}N character.
- The ^{D}T character if DEL IGNORE has been selected by the four rear-panel switches. This allows systems which use ^{D}T characters for padding to be accommodated. An ^{E}C **?** character sequence should be substituted for the ^{D}T character in Style II commands if this selection is made. In this case, both the ^{E}C and **?** (not the replaced ^{D}T character) must be included in the checksum and may not be split between blocks.

The 12-bit accumulator is initially set to "0". If the addition of the ADE value of each 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 integer string and transmitted as a BLOCK END command argument followed by a command terminator. When the block is received by the plotter, the checksum is recomputed using the same procedure to check for errors in the block's data.

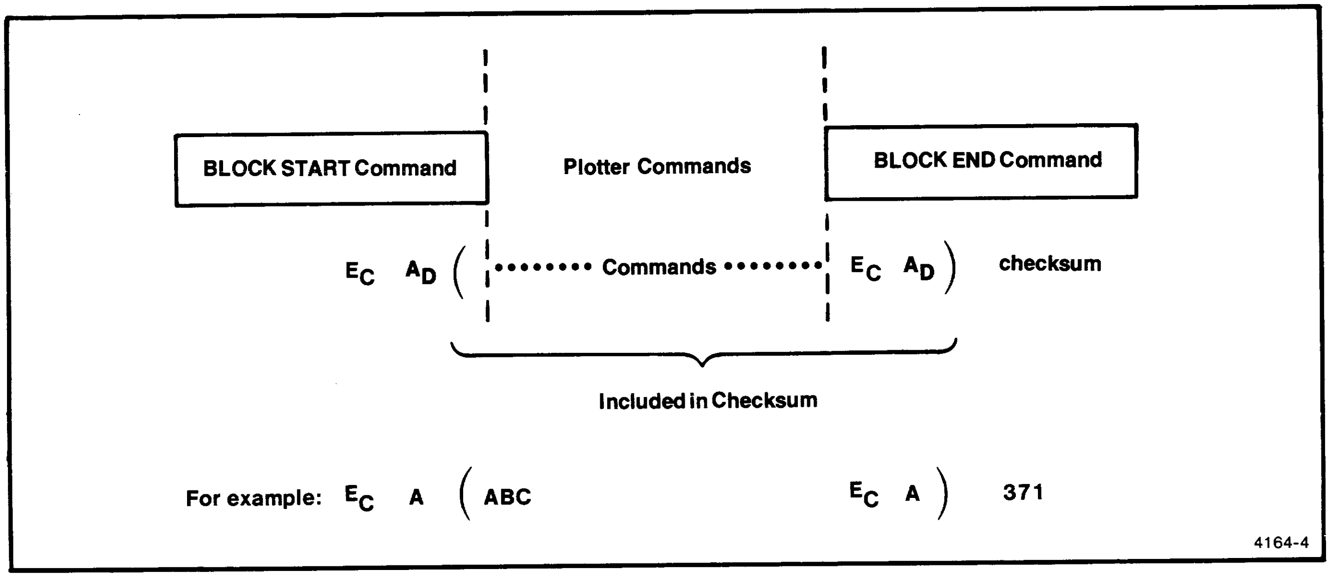


Figure 2-8. Block Mode Checksum.

Turning Block Mode ON and OFF. Described here are procedures for enabling and disabling Block mode communications. Figure 2-9 is a state transition diagram which illustrates the relationship between Block mode and Continuous mode. Each circle in the figure 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.

NOTE

DC1/DC3 Flagging mode is not included in this diagram. This is because DC1/DC3 Flagging mode operates independently of, and is compatible with, either Block or Continuous mode.

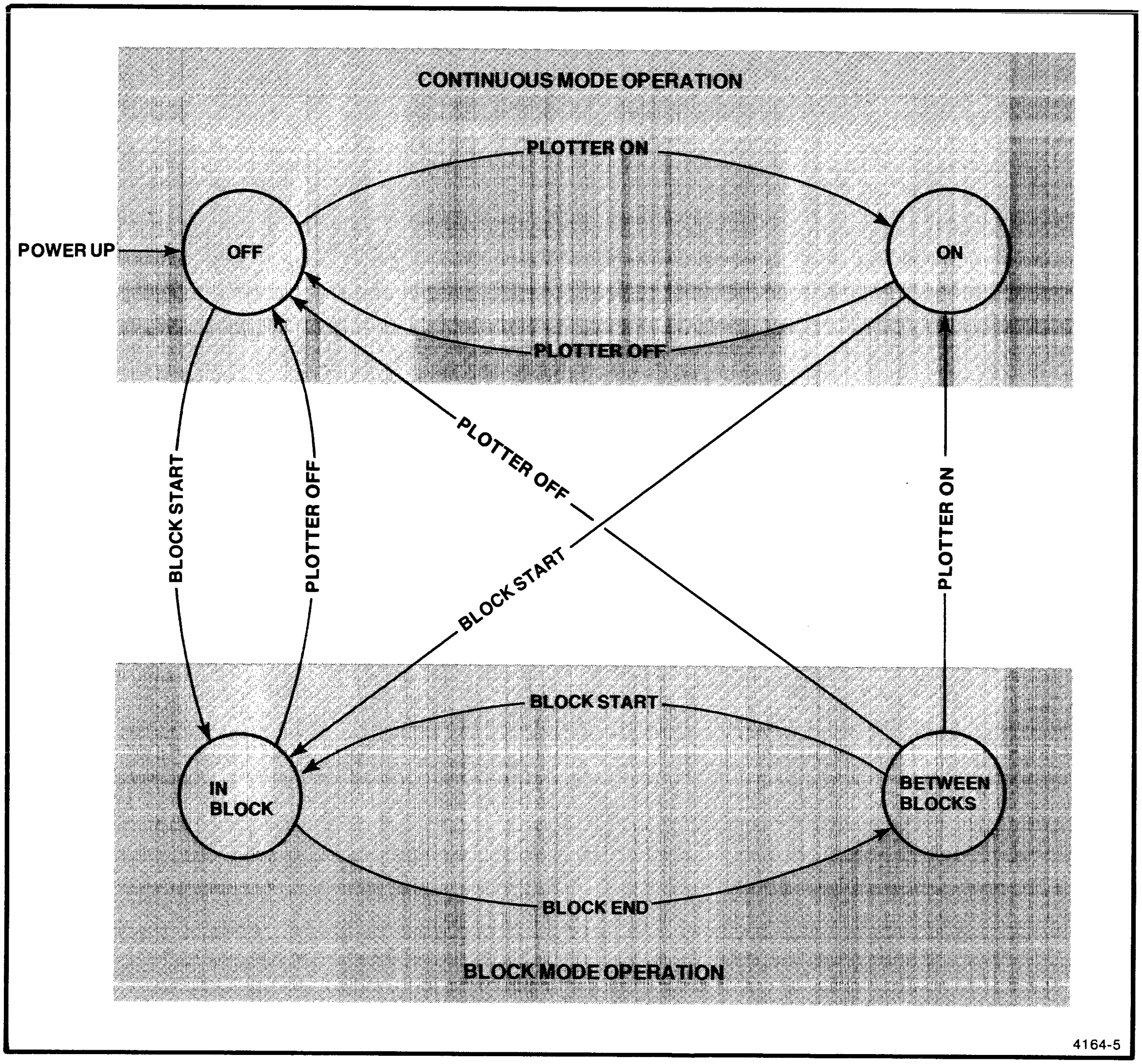


Figure 2-9. Communications Mode State Diagram.

Continuous mode automatically occurs at plotter power-up. Continuous mode is represented by the PLOTTER ON and PLOTTER OFF states. In these states, commands are recognized and executed as described under *Continuous Mode*, earlier in this section.

If a BLOCK START command is received, Block mode communication is initiated (the IN BLOCK state). Communication continues in Block mode until either a PLOTTER ON or PLOTTER OFF command is received. 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 and the checksum is verified, the plotter enters the BETWEEN BLOCKS state 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/commands except PLOTTER ON, PLOTTER OFF, or BLOCK START commands are ignored. This state allows the interface to ignore any unexpected characters transmitted by the host between blocks.

Existing Block Mode Support Software. Several host support software packages are available from Tektronix and may 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 software. These packages provide capabilities from simple graphing to state-of-the-art graphics for the plotter. For further information about these software products, contact your local Tektronix sales office.

PLOTTER-TO-DEVICE TRANSMISSION

The plotter transmits several types of information to the host computer in response to certain plotter commands. The following paragraphs describe the transmission of output information:

- 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.

Communications Control. One type of communications control information is the ${}^{D}{}_{1}$ and ${}^{D}{}_{3}$ characters used for DC1/DC3 flagging rate control (in Option 31 equipped plotters). 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, preceded by a signature character (if one is programmed) and followed by a selected output terminator.

Command Responses. The other class of plotter output is the output which is generated in response to receiving certain plotter commands or to an operator using the front-panel CALL switch to digitize the current pen coordinates. The response to each command is transmitted as one output block. Each output block contains two numeric response values and a value that identifies the type of output.

The size of the output block can be either one or seven characters, plus a possible signature character if one is programmed. A zero- to two-character terminator (determined by the setting of the four rear-panel switches) is also appended to the end of each output message, which may, under certain circumstances, contain more than one data block.

Output Block Identification

When the host receives an output response block, the response must be identified and associated with the output-generating command which 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 you identify the output-generating commands:

- 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, and call digitizing responses, which are generated at the time the operator presses the CALL switch. (Block acknowledge responses have priority over these digitizing responses.)
- The last data character in the response block is encoded to identify the type of response.
- 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

The following describes how the plotter's output blocks are stored and ordered for transmission. As the plotter's outputs are generated, each is stored until all output transmission constraints are satisfied before being transmitted to the host. These contraints are discussed in more detail later in this section. Only one flagging control character or one Block mode control block can be ready for transmission at any one time. Two or more command response blocks may be ready for transmission at once, provided the number of characters in **ALL** pending outputs does not exceed the plotter's output buffer capacity (72 characters).

NOTE

In general, the host should wait for the response of one output-generating command to be returned from the plotter before sending another output-generating command; otherwise, output from subsequent commands may exceed the buffer capacity and be lost.

The output buffer is cleared whenever the front-panel LOCAL switch is pressed or by a DEVICE RESET command.

Formation of Output Messages

Two criteria determine the way in which types of buffered output are returned to the host: the priority of the output, and any possible limitations of the host computer in receiving the response.

Buffered output may consist of $^D_1$ and $^D_3$ control characters, Block mode control blocks, or response blocks (including operator digitizing blocks).

$^D_1$ and $^D_3$ flagging control characters have the highest priority for transmission from the plotter. Because the host computer continually scans for the $^D_1$ and $^D_3$ characters, the plotter can transmit them at any time regardless of any current output activity. When transmission is not in process, the flagging character is transmitted immediately, and the plotter ignores any output initiation constraints. When transmission is currently in process, the flagging character is the next character transmitted, even if this places the character within an output block. Output termination characters are not added to the flagging characters when they are transmitted, but are transmitted as single characters.

The next priority is any pending Block mode control block (Block acknowledge characters). A pending control block is always contained in the next output message which is initiated 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 only store incoming data and inhibit processing until an output terminator (End of Line) is received. Since the block containing the Block mode acknowledge character is a control block, the information needs to be processed quickly and is always immediately followed by the currently defined output terminator characters selected by the four rear-panel switches. 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 transmitted when the output initiation constraints have been satisfied.

Many host computers also have limitations on the length of the line received. Since the selected output termination characters are appended to each output message and a message is at most 72 characters, line length is normally not a problem. However, the maximum size of a response message is limited to one response block (seven characters plus a signature character, if set), followed by the selected terminator, if the host limits the number of response-generating commands in a data block to one.

Output Initiation

In many cases, due to hardware or software limitations, the host computer may only be able to receive responses from the plotter when it is "ready for input." This 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 can be lost. Two output initiation mechanisms are available in the plotter to control response transmissions to ensure that the host is ready to receive them. These mechanisms 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 have been satisfied, transmission starts and continues until a complete output block (ending with the selected output termination characters) has been transmitted.

Turnaround Delay Function. This function accommodates host computers that are not able to read for a short time after each transmission. 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 is enabled to initiate transmission. Each time a character (including a padding character) is 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 also are not able to read for a short period after they receive an output terminator (End of Line). The Turnaround Delay function accommodates this restriction by inhibiting the start of a second transmission message (for the Turnaround Delay period) after the first output message has been transmitted, even if all other transmission requirements have been satisfied.

Prompt Function. The second mechanism controlling output initiation is the Prompt function. Here, prompts are useful for host computers that exhibit a varying time period between when host transmission ends and when the host is ready to receive information. These hosts transmit a predetermined prompt character to indicate to the plotter that they are ready to accept a message. This function is enabled when the SET PROMPT CHARACTER command changes the prompt character to a non-null value.

The prompt characters may be any ASCII character except $^{N}{}_{L}$, $^{S}{}_{Y}$, and $^{D}{}_{T}$ (if DEL IGNORE is active).

In Continuous mode, the plotter is enabled to scan for the prompt character after receipt of any output-generating command (DIGITIZE, SIZE, or STATUS), if the SET PROMPT CHARACTER command has set the prompt character to some non-null value. Prompt character scanning stops when the block is transmitted in response to the command. A PROMPT LIGHT ON command also enables scanning for a prompt character. The scanning process is not disabled until a PROMPT LIGHT OFF, PLOTTER ON, PLOTTER OFF, or DEVICE RESET command is received, assuming that the operator transmits a number of blocks to the host using the front-panel CALL switch while the PROMPT light is on.

In Block mode, the plotter is enabled to scan for the prompt character only between blocks.

While prompt character scanning is enabled, only PLOTTER ON, PLOTTER OFF, BLOCK START, and DEVICE RESET commands (and the PROMPT LIGHT OFF command in Continuous mode) are recognized, all of which terminate the prompt character scanning.

The prompt character is not recognized within a block during Block mode. The plotter recognizes a prompt character only when prompt scanning is enabled.

Prompt charcters can not be queued up. If the host wants the plotter to transmit two stored outputs, it is necessary to send one prompt character prior to each transmission. One or two prompts will not cause the plotter to transmit more than one stored output message at a time. A new prompt character must be sent after each response block is received if more than one response block has been generated.

Remember that BLOCK END commands (which cause a block acknowledge to be generated) and call digitizing (which allows the operator to generate digitizing response blocks) are included in the output-generating commands category. However, the Call Digitize function is permanently enabled. This permits messages to be generated without a corresponding output-generating command to enable the prompt scan. For these formats, the PROMPT LIGHT ON command must be used to enable the prompt scan if operator-digitized output is expected but not yet generated. If the prompt scan is enabled by a PROMPT LIGHT ON command, it will be disabled only by a PROMPT LIGHT OFF or a DEVICE RESET command. Other plotter functions may turn off the Prompt light, but they will not disable the prompt scan.

Combined Output Initiation. A combination of the Turnaround Delay and the Prompt mechanisms satisfy the constraints of most host computers. If a Turnaround Delay or Prompt character 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 character 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. An example is when the host transmits the prompt string and then stops until the turnaround delay period has expired.

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 first processed and its checksum verified.
2. A positive Block mode acknowledge control block is generated.
3. When the Prompt character has been received from the host computer and the Turnaround Delay has expired, the control block is transmitted as an output message preceded by any defined signature character and 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 character is received from the host and the Turnaround Delay has again expired, the two response blocks are transmitted as an output message.

   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 next prompt then causes the two response blocks to be transmitted as one message, each preceded with any defined signature character, but with only one terminator sequence.

NOTE

In general, each block mode data block transmitted by the host should contain only one output-generating command. If more than one response is required, a separate data block should be sent with another output-generating command after the response to the first command has been received. However, if a host data block contains more than one response-generating command, the host must continue to send prompts and accept all of the responses generated before the next data block is 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 normally 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 set a bypass cancel character. The bypass cancel character may be set to any ASCII character except NL, SY, or DT (if DEL IGNORE is active).

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 echoes CR and LF characters for each CR received, the bypass character should be 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.

Section 3

INTRODUCTION TO SERIAL COMMANDS AND RESPONSES

INTRODUCTION

This section describes how to form commands that control the plotter, and how to interpret responses sent by the plotter. A host, a terminal, or a data storage device issues commands to the plotter to initiate actions. An action, for example, may be drawing a line or a response (to be sent back to the host computer).

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

Style I commands are plotter commands which use ASCII decimal data coding.

Style II commands use binary data coding for efficient transmission. Style II commands are the type used with the TEKTRONIX 4010 Series Computer Display Terminals.

The following pages describe Style I and Style II commands separately and then show how both command styles are combined to control the plotter.

NOTE

> *As characters received by the Serial Interface are processed into commands, any* $^{\mathrm{N}}\mathrm{U}$, $^{\mathrm{S}}\mathrm{Y}$, $^{\mathrm{D}}\mathrm{T}$ *characters (*$^{\mathrm{D}}\mathrm{T}$ *if DEL IGNORE is selected by the four rear-panel switches), are discarded immediately on receipt and may appear anywhere within the data transmission.*

Appendix H shows an example of a simple plot using both Style I and Style II commands.

STYLE I COMMANDS (PLOTTER COMMANDS)

The general form of Style I commands is:

<Attention Character> <Address Character> <Command Code> <Arguments>

Figure 3-1 shows the command form for Style I commands. Each part of the Style I command, along with any separators and terminators required to correctly punctuate the command, is discussed in the following paragraphs. Also presented is the plotter response to incomplete or unrecognized commands.

ATTENTION CHARACTER

Each Style I command begins with an attention character $^{A}_{T}$. The attention character identifies the beginning of a command. The attention character is $^{E}_{C}$ and should not be used for other purposes, except as part of a Style II attention action sequence command.

ADDRESS CHARACTER

The address character is a single uppercase ASCII character which must agree with the address character chosen by the four rear-panel switches for the command to be accepted by the plotter. This allows commands to be directed to a single plotter when several 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.

NOTE

If several plotters, set for different addresses, are connected to the same communication line, only one plotter should be logically on at a time (only one plotter should have been sent a PLOTTER ON command at a time). While the other plotters will ignore commands not addressed to them, they will attempt to interpret any command arguments as text and print the command arguments.

NOTE

If all plotters have the same address, then no command (or block mode) which generates output can be used because all of the plotters would attempt to respond simultaneously.

The address choices are **A, B, C,** or **D**.

COMMAND CODE

The command code consists of one or two uppercase ASCII characters which identify the command. The command code for each command is defined in Section 4.

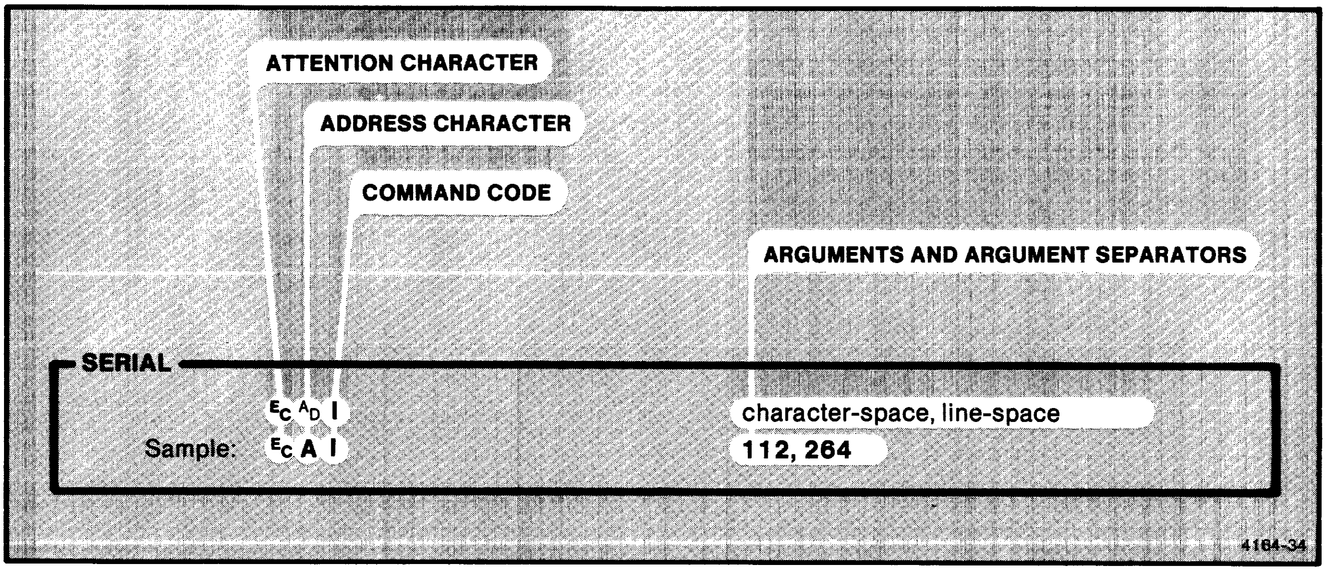


Figure 3-1. Typical Command Form for Serial Style I Commands.

ARGUMENTS

Any arguments defined for the particular command follow the command code. Specific arguments for each command are described in Section 4. Command arguments may be either numeric, integer, or character. Individual numeric arguments are separated 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*, described later in this section.

Numeric Arguments

Numeric arguments may be expressed in three forms: integer, floating point, or scientific notation. Here are examples of each form:

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

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 may be expressed in either integer or in floating point form.
- The E should be uppercase (lowercase is accepted but not recommended) and the exponent must be expressed in 1 or 2 digits.
- A "+" sign is not required if the exponent is positive.
- A space may 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 which 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 adhere to ANSI Standard X3.42. The units, resolution, and allowable range of numeric arguments are unique to each command and are noted in the command descriptions.

Integer Arguments

Integer arguments consist of only the digits 0 through 9 and they must immediately follow the command code with no spaces. Integer arguments are terminated by the first character not in the range 0 through 9. Leading zeroes may be included and are ignored. The allowable range of the arguments are specified for each command.

Character Arguments

A character argument is a single ASCII character which may be any character except NL, SY, and DT (if DELETE IGNORE is selected by the four rear-panel switches). The character argument must be the very next character after the command character.

ARGUMENT SEPARATORS

Argument separators are special sequences of characters which 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 may also be expressed in one of two more general forms:

- A sequence of one or more spaces, or
- A comma, followed by a sequence of zero or more spaces.

NOTE

The argument separator implies that another argument follows. Therefore, do not follow the last argument of a command with a space or comma character (which would be interpreted as an argument separator).

COMMAND TERMINATORS

A command terminator is a character which 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.

NOTE

Commands that have no arguments, such as PLOTTER ON or PLOTTER OFF, or commands that have a character argument, such as SET PROMPT CHARACTER, do not require a terminator.

The terminator used in the plotter is called a *reprocessed terminator.* Reprocessed terminators are characters which terminate a command but are also used as a part of the next command. An example is when the attention character for one command terminates that command and also starts the next command. The last argument of a command must be properly terminated. If it is not known what character will follow a Style I command, the Style I command should be terminated by an explicit command terminator.

NOTE

It is recommended that all Style I commands be explicitly terminated by a terminator that will be ignored as a reprocessed command terminator. SB *is an excellent choice for this function.*

NOTE

The characters NU, SY, DT *(if DEL IGNORE is selected by the four rear-panel switches), the prompt character, or the bypass cancel character (when bypass is active) will not act as command terminators.*

Both the semicolon character and the CR character are treated as reprocessed command terminators and as part of a Style II command.

COMMAND INTERPRETATION

Errors in forming or communicating commands may cause the plotter to receive commands which cannot be recognized as a valid Style I command. The following outlines the plotter's responses to different situations:

- 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 **D**), but is not the current plotter address character, the command is ignored, but the command code and arguments are treated as Style II commands.
- If the address character is the current plotter address, but the command code is not recognized, the command is ignored but the arguments are treated as Style II commands.
- If a command is entered in error, a new command can be started anywhere if it is 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.

Careful termination of commands assists the recovery from command interpretation errors.

STYLE II COMMANDS (4010 SERIES TERMINAL COMMANDS)

The Style II commands recognized by the plotter are very similar to the host commands that control TEKTRONIX 4010 Series Computer Display Terminals. For most standard functions, the plotter responds to host commands in the same way as a TEKTRONIX 4014 Computer Display Terminal. This allows plots developed for a 4010 Series terminal to be sent to the plotter for a high quality hard copy. The coding of Style II commands results in high efficiency communication. These commands are normally generated by host computer support software (such as Tektronix PLOT 10 software).

The basic Style II commands are shown in Figure 3-2. Each command controls some basic plotter action, such as printing characters, moving the pen, and drawing lines with the pen. The Attention Action commands control special functions. Each of these commands (with their functions and parameters) are discussed later in this section.

Style II commands are expressed to the plotter by first sending a mode control character (such as, US, GS, or EC) to place the plotter into the proper mode. The relationship between these modes is defined by the mode transition diagram shown in Figure 3-3. The mode control characters are similar to command codes. The US (alpha) and GS (graphic) modes "remember" that the corresponding commands are being processed. Therefore, the mode control characters do not need to be repeated for each successive command unless the mode is changed. While the plotter interface is in either Graph or Alpha mode, successive ASCII characters received are interpreted as arguments for the corresponding commands. The required coding for these arguments is described for each of the modes.

ALPHA MODE

In order to print characters, a U_S character is sent to the plotter to force the current mode to Alpha. The plotter automatically resets to Alpha mode at power-up or after a DEVICE RESET command. In Alpha mode, subsequent characters sent to the plotter are normally printed beginning at the current pen position. Refer to *Alpha Commands* in Section 4 for a detailed description of the print actions for each ASCII character.

MODE	COMMANDS	GENERAL FORM
Alpha	PRINT	U_S ...char... ...char... – – – ...char...
Graph	MOVE/ DRAW•••	G_S ...xy coordinate... ...xy coordinate... – – – ...xy coordinate...
Attention Action	ATTENTION	E_C ...char...

4164-7

Figure 3-2. Basic Style II Commands.

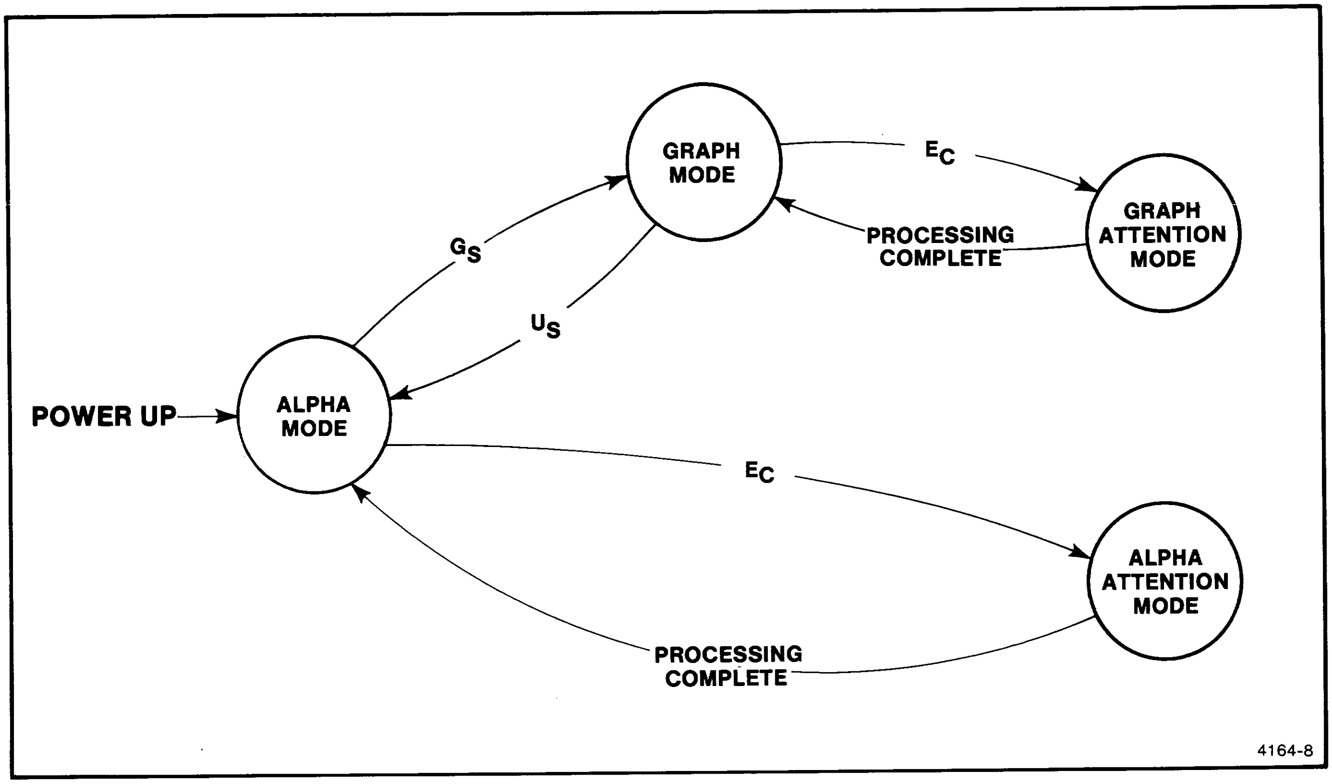


Figure 3-3. Style II Command Modes.

GRAPH MODE

To draw lines utilizing Style II commands, a $^{G}{}_{S}$ character is sent to the plotter to place it into Graph mode. In Graph mode, the plotter can perform moves or draw lines. To perform move and draw actions, you specify the command arguments (as encoded pairs of X and Y coordinates).

The first XY 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 XY coordinate pairs are interpreted as DRAW command coordinates, which cause a straight line to be drawn from the current position to the specified positions.

A "move" may follow "draws" by preceding the MOVE command's coordinate with another $^{G}{}_{S}$ character. The "move" following the $^{G}{}_{S}$ may be suppressed either by (1) following the $^{G}{}_{S}$ with the coordinates of the current pen position (which generates a move to the current position), or (2) following the $^{G}{}_{S}$ character with a $^{B}{}_{L}$ (bell) character (which causes the first coordinate pair to be interpreted as a "draw" and rings the bell).

Coordinate Encoding

The XY coordinate pairs sent to the plotter as MOVE or DRAW command coordinates consist of two 12-bit binary coordinates in Figure 3-4. This gives a 0 to 4095 coordinate range where the units are in ADUs.

NOTE

The XY coordinates specified in Style II commands are always interpreted as ADUs.

NOTE

The 4662 plotter accepts 16-bit graphics suitable to the 4663 plotter (i.e., graphics containing EEB and EEEB bytes). However, these extra bytes do not increase the 4662 plotter's resolution.

The following paragraphs describe how coordinates are coded and sent to the plotter. Figure 3-5 defines five groups of binary bits within the 12-bit binary description. The bit patterns which may appear in these groups are mapped into portions of the ASCII character set as shown in Figure 3-6. 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 five coding groups. The appropriate ASCII characters are then used to express the encoded coordinates (see Table 3-1). The HIX and HIY groups are assigned to the same portion of the ASCII character set and are identified by the order in which the plotter receives them. The same holds true for the extra byte (EB) which is assigned to the same portion as 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, a following sequence must begin with another $^{G}{}_{S}$ character to mark the beginning of a sequence.

NOTE

This coding scheme requires that the host computer is able to transmit all of the characters in the last six columns of Figure 3-6, except $^{D}{}_{T}$ plus the $^{E}{}_{C}$, $^{G}{}_{S}$, and $^{U}{}_{S}$ characters.

Some host computers use the $^{D}{}_{T}$ character for system functions. In this case, an $^{E}{}_{C}$? attention action command may be substituted for the $^{D}{}_{T}$ character during encoding of the XY coordinate pair, and the plotter's rear-panel switches may be set to select DELETE IGNORE. This causes the plotter to discard the $^{D}{}_{T}$ characters. Refer to *Attention Mode*, later in this section, for more information.

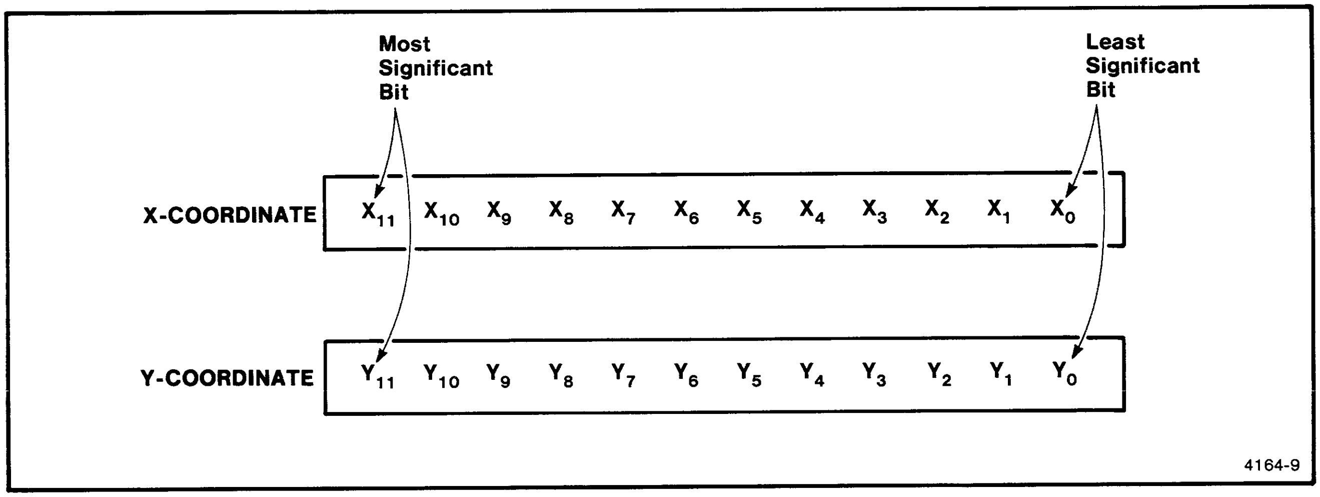

Most Significant Bit
Least Significant Bit
X-COORDINATE
X_{11} X_{10} X_9 X_8 X_7 X_6 X_5 X_4 X_3 X_2 X_1 X_0
Y-COORDINATE
Y_{11} Y_{10} Y_9 Y_8 Y_7 Y_6 Y_5 Y_4 Y_3 Y_2 Y_1 Y_0
4164-9

Figure 3-4. 12-Bit Binary Coordinates.

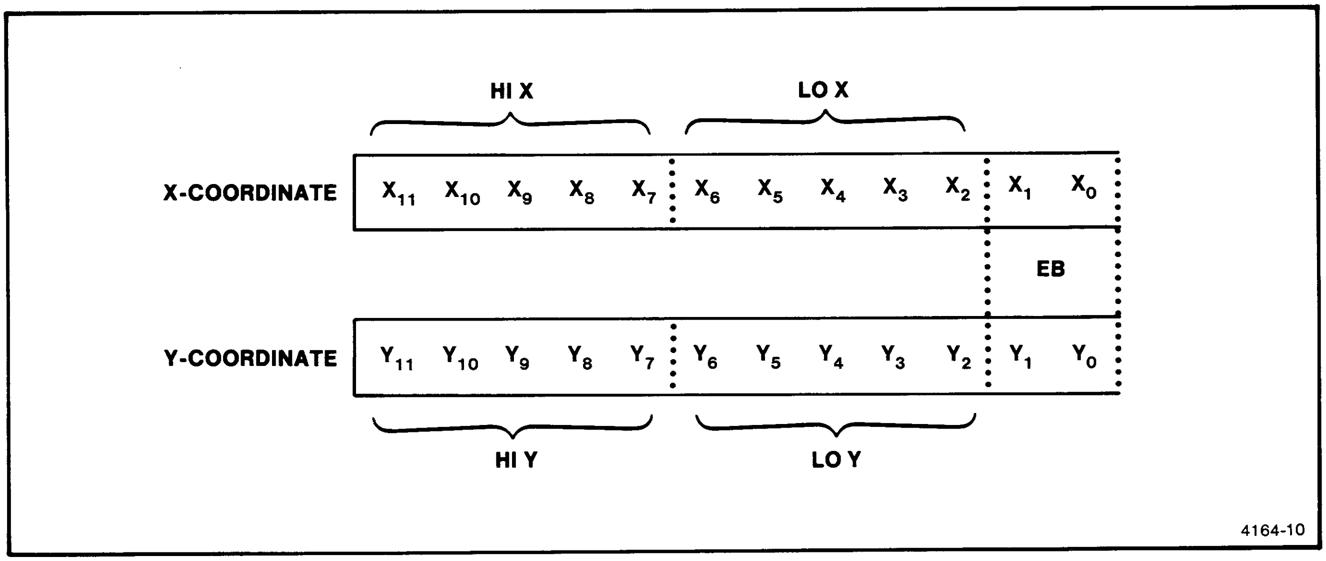

HI X
LO X
X-COORDINATE
X_{11} X_{10} X_9 X_8 X_7 X_6 X_5 X_4 X_3 X_2 X_1 X_0
EB
Y-COORDINATE
Y_{11} Y_{10} Y_9 Y_8 Y_7 Y_6 Y_5 Y_4 Y_3 Y_2 Y_1 Y_0
HI Y
LO Y
4164-10

Figure 3-5. Definition of Graphic Coding Groups.

BITS 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
	CONTROL		HI X	HI Y	LO X		LO Y EB	
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 GTL SH 1 SOH 1	21 LLO 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 SDC ET 4 EOT 4	24 DCL 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 PPC EQ 5 ENQ 5	25 PPU 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 GET BS 8 8	30 SPE 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 TCT HT 9 9	31 SPD 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	mnemonic decimal

4164-11

Figure 3-6. ASCII Code Chart Showing Assignment of Graphic Coding Groups.

Table 3-1 details the bytes required to encode the XY coordinate pair, including the order in which they are transmitted to the plotter. Byte 1 is transmitted first.

Table 3-1

PACKED BINARY COORDINATE CODING DEFINITION

<table>
<tr><th rowspan="3">Byte No.</th><th rowspan="3">Byte Name</th><th colspan="7">7-Bit ASCII Character</th></tr>
<tr><th colspan="2">Fixed Bits</th><th colspan="5">Data Bits</th></tr>
<tr><th>B7</th><th>B6</th><th>B5</th><th>B4</th><th>B3</th><th>B2</th><th>B1</th></tr>
<tr><td rowspan="2">1</td><td rowspan="2">High Order Y (HIY)</td><td rowspan="2">0</td><td rowspan="2">1</td><td>y_{11}</td><td>y_{10}</td><td>y_9</td><td>y_8</td><td>y_7</td></tr>
<tr><td colspan="5">5 most significant bits of Y</td></tr>
<tr><td rowspan="2">2</td><td rowspan="2">Extra Byte (EB)</td><td rowspan="2">1</td><td rowspan="2">1</td><td>0</td><td>y_1</td><td>y_0</td><td>x_1</td><td>x_0</td></tr>
<tr><td></td><td colspan="2">2 low bits of Y</td><td colspan="2">2 low bits of X</td></tr>
<tr><td rowspan="2">3</td><td rowspan="2">Low Order Y (LOY)</td><td rowspan="2">1</td><td rowspan="2">1</td><td>y_6</td><td>y_5</td><td>y_4</td><td>y_3</td><td>y_2</td></tr>
<tr><td colspan="5">5 intermediate bits of Y</td></tr>
<tr><td rowspan="2">4</td><td rowspan="2">High Order X (HIX)</td><td rowspan="2">0</td><td rowspan="2">1</td><td>x_{11}</td><td>x_{10}</td><td>x_9</td><td>x_8</td><td>x_7</td></tr>
<tr><td colspan="5">5 most significant bits of X</td></tr>
<tr><td rowspan="2">5</td><td rowspan="2">Low Order X (LOX)</td><td rowspan="2">1</td><td rowspan="2">0</td><td>x_6</td><td>x_5</td><td>x_4</td><td>x_3</td><td>x_2</td></tr>
<tr><td colspan="5">5 intermediate bits of X</td></tr>
</table>

This coding results in the following graphic character sequence, which specifies a 12-bit XY coordinate pair:

HIY EB LOY HIX LOX

These characters must appear in the defined order to be correctly interpreted. Appendix F contains two types of tables which may be used to determine the coded bytes required to define a given XY coordinate pair.

NOTE

The 4662 plotter accepts the EEB and EEEB bytes used for extra resolution in the 4663 plotter. However, neither of these bytes are used by the 4662 plotter. For increased resolution in the 4663 plotter, the order is: HIY EEEB EEB EB LOY HIX LOX.

Coordinate Resolution

If the full resolution of the plotter is not required, it is not necessary that all 12 bits be transmitted to the plotter. This results in increased transmission efficiency. A lower resolution of 10 bits can be transmitted for the XY coordinates by simply eliminating the extra byte. The optional byte permits coordinate resolutions as shown in Table 3-2.

Table 3-2

COORDINATE RESOLUTION

Coordinate Bits	Resolution (ADUs)	Extra Bytes Required
10	4	none
12	1	EB

NOTE

The mechanical resolution of the 4662 plotter is 1.37 ADUs, unless modified ADUs are selected by the SET PLOTTER UNITS command (available in plotters equipped with Firmware Level 5 or greater), in which case the resolution is 1 ADU. Normally, the coordinate sent is rounded to the closest possible mechanical position.

Optimized Coordinate Encoding

Further transmission efficiency may be achieved by optimizing coordinate transmission. When coordinate specifications are received by the plotter, the graphic bytes update two 12-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. Therefore, it is only necessary to transmit the portion of these coordinates that change from the previous "Move" or "Draw." Table 3-3 defines the minimum number of bytes which must be used to transmit the coordinate bits that changed from the previous coordinate specification.

Table 3-3

INSTRUCTIONS FOR OPTIMIZED BINARY COORDINATE CODING

Bytes to be Changed	Bytes Which Must Be Sent				
	HIY	EB	LOY	HIX	LOX
HIY	X				X
EB		X	X		X
LOY			X		X
HIX			X	X	X
LOX					X

NOTE

If more than one byte changes value, the bytes which must sent are the combination of all the bytes that must be sent for each byte changed. For example, if the HIX and EB bytes change, send only EB, LOY, HIX, and LOX.

If the specification of an XY 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, the restarted graphic sequence should be fully specified and not optimized to ensure that the graphic memory contains the desired values.

The graphic memory in the plotter is only reset to 0 during powerup. Therefore, a plot should normally begin with a full 12-bit coordinate specification, which sets all coordinate bits to known values even if less resolution is used for the remainder of the plot.

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

Refer back to Figure 3-3 and notice that two Attention Action modes exist: Alpha Attention and Graph Attention. Unless the Attention Action command causes a mode change, the current mode (Alpha or Graph) is reestablished after the Attention Action command process is completed.

Unless noted otherwise in the following discussion, the Attention Action commands perform the same functions when received during Alpha or Graph modes. 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 ^{E}C character and a specified action character.

NOTE

There must not be any other characters between the ^{E}C character and the following action character, except for the discarded characters, ^{N}L, ^{S}N, and ^{D}T (if DELETE IGNORE is set on the four rear-panel switches).

Table 3-4 lists the Attention Action commands recognized by the plotter and describes the resultant plotter action. Avoid using any ASCII characters not listed in these tables.

NOTE

Each Attention Action command must be completed before another is specified.

Table 3-4

ATTENTION ACTION COMMANDS AND THEIR EFFECTS

Attention Action Command	Action
^{E}C ?	Is converted to a ^{D}T character and interpreted as a LOY graphic byte.
$^{E}C\,^{B}L$	Bell sounds. May be used to ring bell without affecting the mode.
$^{E}C\,^{F}F$	Forces the Plotter to Alpha mode and performs the Move To Home function (see the PRINT command description in Section 4).
$^{E}C\,^{G}S$	Equivalent to ^{G}S
$^{E}C\,^{U}S$	Equivalent to ^{U}S
$^{E}C\,^{L}F$	Equivalent to ^{E}C
$^{E}C\,^{C}R$	Equivalent to ^{E}C
$^{E}C\,^{E}C$	Equivalent to ^{E}C
$^{E}C\,^{N}U$	Equivalent to ^{E}C

NOTE

Any character following the last four commands in Table 3-4 is interpreted as the second character of another Attention Action command.

MIXING STYLE I AND STYLE II COMMANDS

Style I commands may be viewed as a large set of multicharacter Attention Action commands. Therefore, Style I commands may appear within Style II commands wherever an Attention Action command is appropriate.

Take care in terminating Style I commands because any stray characters following these commands are interpreted as Style II print or graphic arguments, depending on the mode the plotter was in when the Style I command was received.

OUTPUT RESPONSES

Output messages consist of $^{D}{}_{1}$ and $^{D}{}_{3}$ control characters, control blocks, and response blocks. Output messages are transmitted from the plotter to the host in response to output-generating plotter commands. These paragraphs explain how output messages are coded and how the host decodes each response to retrieve the desired information. The $^{D}{}_{1}$ and $^{D}{}_{3}$ control character responses require no decoding. Refer to *Plotter-to-Device Transmission* in Section 2 for more information on how these blocks are transmitted to the host.

CONTROL BLOCKS

Control blocks are used 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*, in Section 2). These blocks are always one character in length and are encoded as shown in Figure 3-7.

The control bit (C) is encoded as either "0" (a positive block acknowledge, which is the ASCII character **A**) or as "1" (a negative block acknowledge, which is the ASCII character **I**). The "1 0" value for Bits 7 and 6 indicate that this is the last byte of a binary coded block. The "0 1" value for Bits 2 and 1 indicate that this is a control block. The parity bit (P) is determined by the setting of the four rear-panel switches.

RESPONSE BLOCKS

This format is used by the plotter when responding to all output-generating commands. The response block, which can range from 7 to 8 characters in length, consists of two data values and a value indicating the block type encoded in the last character of the block. The encoding of the two data values are slightly different for blocks containing plotter pen coordinates (generated by digitizing commands and operator digitizing) and other returned values such as plotter status. See Figures 3-8 and 3-9 for the encoding information.

Refer to the specific output-generating command for more information.

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

WHERE:
P = PARITY BIT
C = CONTROL BIT
O = POSITIVE ACKNOWLEDGE
I = NEGATIVE ACKNOWLEDGE

4164-12

Figure 3-7. Control Block Coding.

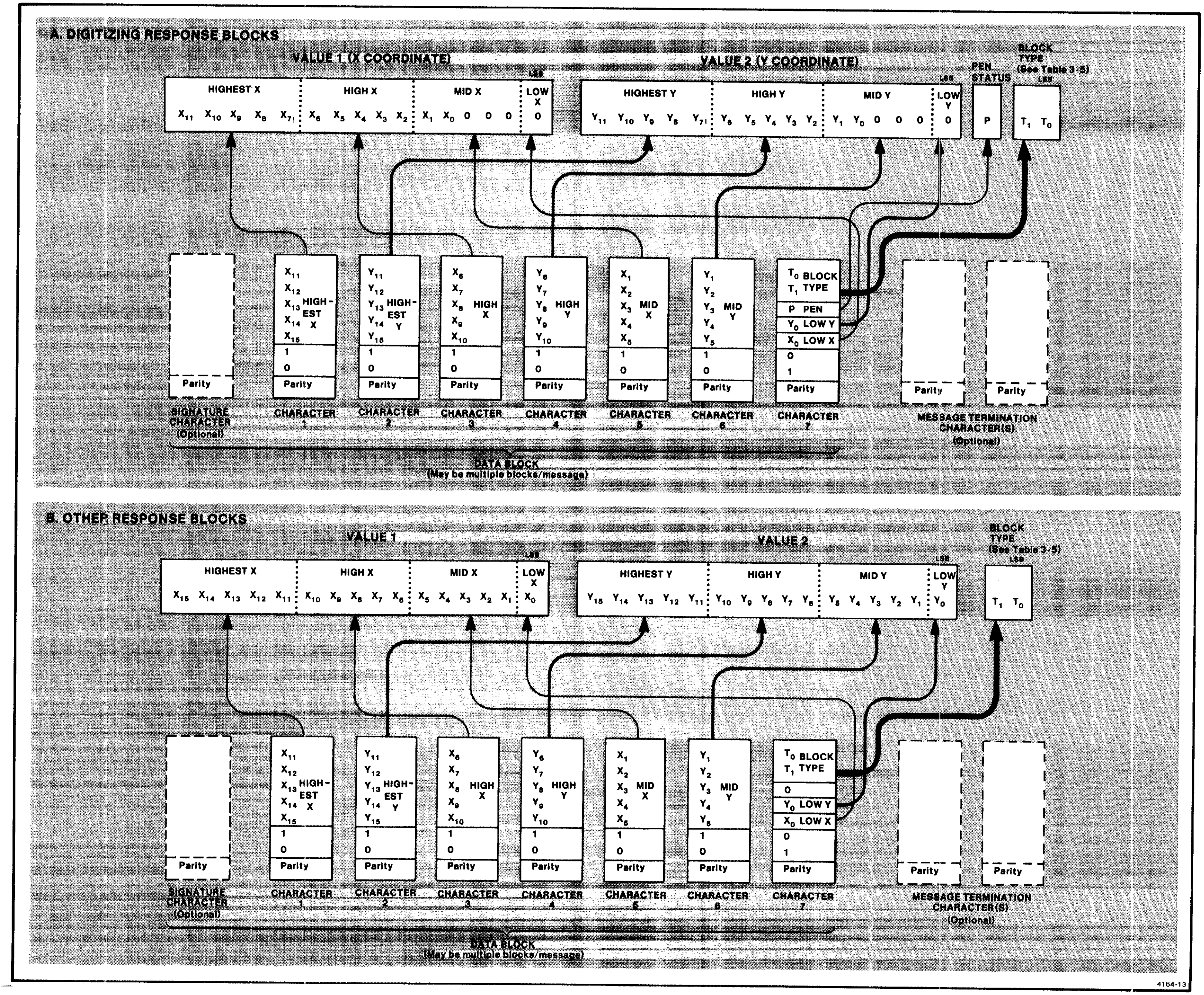

A. DIGITIZING RESPONSE BLOCKS
VALUE 1 (X COORDINATE)
VALUE 2 (Y COORDINATE)
PEN STATUS
BLOCK TYPE (See Table 3-5)
LSB
HIGHEST X
HIGH X
MID X
LOW X
X11 X10 X9 X8 X7
X6 X5 X4 X3 X2
X1 X0 0 0 0
0
HIGHEST Y
HIGH Y
MID Y
LOW Y
Y11 Y10 Y9 Y8 Y7
Y6 Y5 Y4 Y3 Y2
Y1 Y0 0 0 0
0
P
T1 T0
X11 X12 X13 X14 X15 HIGHEST X 1 0 Parity
Y11 Y12 Y13 Y14 Y15 HIGHEST Y 1 0 Parity
X6 X7 X8 X9 X10 HIGH X 1 0 Parity
Y6 Y7 Y8 Y9 Y10 HIGH Y 1 0 Parity
X1 X2 X3 X4 X5 MID X 1 0 Parity
Y1 Y2 Y3 Y4 Y5 MID Y 1 0 Parity
T0 BLOCK T1 TYPE
P PEN
Y0 LOW Y
X0 LOW X
0
1
Parity
SIGNATURE CHARACTER (Optional)
CHARACTER 1
CHARACTER 2
CHARACTER 3
CHARACTER 4
CHARACTER 5
CHARACTER 6
CHARACTER 7
MESSAGE TERMINATION CHARACTER(S) (Optional)
DATA BLOCK (May be multiple blocks/message)
B. OTHER RESPONSE BLOCKS
VALUE 1
VALUE 2
BLOCK TYPE (See Table 3-5)
LSB
HIGHEST X
HIGH X
MID X
LOW X
X15 X14 X13 X12 X11
X10 X9 X8 X7 X6
X5 X4 X3 X2 X1
X0
HIGHEST Y
HIGH Y
MID Y
LOW Y
Y15 Y14 Y13 Y12 Y11
Y10 Y9 Y8 Y7 Y6
Y5 Y4 Y3 Y2 Y1
Y0
T1 T0
X11 X12 X13 X14 X15 HIGHEST X 1 0 Parity
Y11 Y12 Y13 Y14 Y15 HIGHEST Y 1 0 Parity
X6 X7 X8 X9 X10 HIGH X 1 0 Parity
Y6 Y7 Y8 Y9 Y10 HIGH Y 1 0 Parity
X1 X2 X3 X4 X5 MID X 1 0 Parity
Y1 Y2 Y3 Y4 Y5 MID Y 1 0 Parity
T0 BLOCK T1 TYPE
0
Y0 LOW Y
X0 LOW X
0
1
Parity
SIGNATURE CHARACTER (Optional)
CHARACTER 1
CHARACTER 2
CHARACTER 3
CHARACTER 4
CHARACTER 5
CHARACTER 6
CHARACTER 7
MESSAGE TERMINATION CHARACTER(S) (Optional)
DATA BLOCK (May be multiple blocks/message)
4164-13

Figure 3-8. Packed Binary Response Message Format.

A. DIGITIZING RESPONSE BLOCKS

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_{11}	X_{10}	X_9 5 MSB of Value 1 (X Coordinate)	X_8	X_7
2	HIGHEST Y	0	1	Y_{11}	Y_{10}	Y_9 5 MSB of Value 2 (Y Coordinate)	Y_8	Y_7
3	HIGH X	0	1	X_6	X_5	X_4 5 Intermediate Bits of Value 1 (X Coordinate)	X_3	X_2
4	HIGH Y	0	1	Y_6	Y_5	Y_4 5 Intermediate Bits of Value 2 (Y Coordinate)	Y_3	Y_2
5	MID X	0	1	X_1	X_0	0 Next 5 Intermediate Bits of Value 1 (X Coordinate)	0	0
6	MID Y	0	1	Y_1	Y_0	0 Next 5 Intermediate Bits of Value 2 (Y Coordinate)	0	0
7	LOX, LOY, PEN STATUS, BLOCK TYPE	1	0	0	0	P	T_1	T_0

PEN STATUS   BLOCK TYPE

B. OTHER RESPONSE BLOCKS

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} 5 MSB of Value 1	X_{12}	X_{11}
2	HIGHEST Y	0	1	Y_{15}	Y_{14}	Y_{13} 5 MSB of Value 2	Y_{12}	Y_{11}
3	HIGH X	0	1	X_{10}	X_9	X_8 5 Intermediate Bits of Value 1	X_7	X_6
4	HIGH Y	0	1	Y_{10}	Y_9	Y_8 5 Intermediate Bits of Value 2	Y_7	Y_6
5	MID X	0	1	X_5	X_4	X_3 Next 5 Intermediate Bits of Value 1	X_2	X_1
6	MID Y	0	1	Y_5	Y_4	Y_3 Next 5 Intermediate Bits of Value 2	Y_2	Y_1
7	LOX, LOY, BLOCK TYPE	1	0	X_0	Y_0	T_2	T_1	T_0

LSB OF VALUE 1   LSB OF VALUE 2   BLOCK TYPE

4164-14

Figure 3-9. Packed Binary Response Format.

The last data character in the block (Character 7) encodes the low two bits to indicate the block type as shown in Table 3-5.

Table 3-5

BLOCK TYPE

T1	T0	Block Type
0	0	Digitizing Responses
0	1	Control Block (described earlier)
1	0	Status Response
1	1	Size Response

Following is a description of the data blocks.

The Packed Binary format, used for response blocks, consists of seven consecutive ASCII characters. These are encoded as shown in Figure 3-8 from two 16-bit numbers, a pen status bit, and two bits of block type coding. The coding (defined more formally in Figure 3-9) 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 to 65535. For coordinate values, the 12-bit coordinate value is returned using the highest 12 bits of the 16-bit value — see Figures 3-8 and 3-9 for these representations.

The pen status bit is valid only for digitizing response blocks. The pen status bit is set to zero if the pen is up and one if the pen is down on the plotting surface. It is set to 0 for all other block types.

Each response block (including control blocks) are preceded by the signature character if one has been defined by a SET SIGNATURE CHARACTER command.

Several response blocks may be transmitted to the host in a single output message. In this case, each response block in the message is preceded by a signature character if one is defined, but only one terminator sequence is appended to the end of the entire message.

The message termination characters depend on the setting of the rear-panel switches and may be CR, CR and ET, or none.

COMMAND NOTATIONS

The following paragraphs describe the conventions used in this manual to represent commands and their arguments.

CHARACTERS

ASCII Characters

The 94 printing ASCII characters (numbers, symbols, 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 in Appendix C).

NOTE

The ASCII space character is always shown as SP, and not as a "blank space" between printed characters.

Examples of characters: & 1 2 A B a b EC SP DT

The last three ASCII characters are Escape, Space, and Delete.

Special Characters

The AT symbol is always the EC ASCII character. This is the attention character for serial communications.

The AD symbol is replaced by one of the following ASCII characters. This is the plotter's address character.

A, B, C, or **D**

LITERAL VERSUS VARIABLE ELEMENTS

Boldface

Boldface is used to indicate a literal element, entered exactly as shown.

(The "at" signs in this manual are an exception to this rule; although the @ appears in regular type, enter the @ exactly as shown in the commands.)

Regular Type

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

Examples:
EC AD **R** rotation-angle

In this example, two elements (AD and rotation-angle) are to be replaced and EC and **R** are entered exactly as shown.

EC **A R 45**.

In this example, all five ASCII characters are entered exactly as shown.

COMMAND ELEMENTS

Element Types

The last word(s) of a 4662 plotter variable element name indicates the element type required by the 4662 as shown below:

Last Word(s)	Element Type
integer	integer
selector	selector
character	character

Other endings imply a "numeric" element.

For example:
- Pen-selector is a "selector" element.
- Checksum-value-integer is an "integer" element.
- Prompt-character is a "character" element.
- Rotation-angle is a "numeric" element.

Argument Separators

A comma (,) between elements can be replaced by either a comma (,) or one or more SP characters.

Command Terminations

Command terminators are generally NOT shown in the individual command descriptions. Refer to *Command Terminators* earlier in this section for instructions on how to terminate commands.

NOTATIONS

Brackets [ ]

An element inside brackets is OPTIONAL. Stacked elements within brackets indicate that you may select one or none of the elements.

Example:

[0]
[1]

Select 0, 1, or neither.

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 the next printed line, the additional line is indented.

Example:

ECAD **I**character-space,
    line-space

Section 4

SERIAL COMMAND DESCRIPTIONS

COMMAND DESCRIPTION OVERVIEW

Serial commands are divided into major groups according to similar functions. These groups are:

- Serial Interface Commands
- Device Commands
- Alpha Commands
- Graphic Commands
- Digitizing Commands
- Multiple Pen (Option 31) Commands

The discussion of each command group is divided into two parts: general concepts about the group, and specific command descriptions for each command in the group. Each command description follows a general format and contains all or some of the following:

- Purpose
- Syntax Box
- Parameters (This includes restrictions, meanings assigned to selector arguments, etc.)
- Outputs
- Comments
- References
- Examples

The syntax box shows the recommended command form that can be used under most conditions. The syntax box includes a specific example of the command. *Comments* and *References* are sometimes included to clarify specific entries in the syntax box.

Appendix H shows a simple plot using several of the commands described in this section.

SERIAL INTERFACE COMMANDS

INTRODUCTION

Serial Interface commands control the communication between a host computer and the plotter and are executed immediately upon receipt. Therefore, it is important not to include Serial Interface commands within Block mode blocks. (See Block mode description in Section 2.)

CONCEPTS

Refer to Section, *Serial Interface Concepts*, for more information.

SERIAL INTERFACE COMMAND DESCRIPTIONS

There are nine Serial Interface commands:

- PLOTTER ON
- PLOTTER OFF
- BLOCK START
- BLOCK END
- SET TURNAROUND DELAY
- SET BLOCK SIZE
- SET BYPASS CANCEL CHARACTER
- SET SIGNATURE CHARACTER
- SET PROMPT CHARACTER

Each command is described separately in the pages that follow.

PLOTTER ON

Purpose: This command turns the plotter logically on.

SERIAL

$^{E}{}_{C}$ $^{A}{}_{D}$ **E**

Example: $^{E}{}_{C}$ **A E**

Comments

The logical ON state means the plotter interprets as commands all characters received after a PLOTTER ON command. The TERMINAL MUTE function is enabled if it has been chosen by the rear-panel switch settings. This means that in a loop-through configuration, received characters, beginning with the character following the command character **E** of the PLOTTER ON command, are not sent to the terminal. This condition remains in effect until a PLOTTER OFF command is received.

A BLOCK START command also performs plotter-on functions.

NOTE

At power-up the plotter is in the logical OFF state except if it is in Copy mode. In Copy mode the plotter powers-up logically on and no PLOTTER ON command is necessary (see the 4662 Interactive Digital Plotter Operator's Manual). (In Copy mode, a PLOTTER OFF command may still be sent to turn the plotter logically OFF.)

PLOTTER OFF

Purpose: This command turns the plotter to the logical OFF state.

SERIAL

$^{E}_{C}$ $^{A}_{D}$ **F**

Example: $^{E}_{C}$ **A F**

Comments

The plotter's logical OFF state means that all characters received after a PLOTTER OFF command are not interpreted as commands (until either a PLOTTER ON or a BLOCK START command is received). The TERMINAL MUTE condition, if set by the rear-panel switches, is disabled. The terminal then receives all characters sent by the host computer following the **F** command character in the PLOTTER OFF command.

NOTE

The plotter is logically OFF at power-up except if the rear-panel switches select Copy mode, in which case no PLOTTER ON is necessary unless a PLOTTER OFF command is sent to turn the plotter logically OFF.

BLOCK START

Purpose: This command is used to begin a block in Block mode communications and also turns the plotter logically ON (this is equivalent to a PLOTTER ON command).

SERIAL

$^{E}{}_{C}$ $^{A}{}_{D}$ **(**

Example: $^{E}{}_{C}$ **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. Refer to Section 2, *Serial Interface Concepts*, for more information on checksum accumulation.

References

A complete description of Block mode communications is given in Section 2.

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

$^{E}_{C}$ $^{A}_{D}$ **)** checksum-value-integer

Example: $^{E}_{C}$ **A) 373**

Parameters

checksum-value
Block checksum value, 1-4095 (integer argument).

Outputs

One character (plus signature character and output termination character 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 begins with the **(** character of the BLOCK START command and ends with the **)** character of the BLOCK END command. Information concerning checksum calculations is presented in Section 2, *Serial Interface Concepts*.

References

See Section 2 for complete information on Block mode communication.

SET TURNAROUND DELAY

Purpose: This command sets the turnaround delay time.

SERIAL

EC AD **G** delay-time-integer

Example: EC **A G 200**

Parameters

delay-time

Delay time, in milliseconds, 0-65535 (integer argument). Resolution is approximately 8 milliseconds.

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 causes the plotter to wait the specified time before transmitting to the host computer. This delay is also used between consecutive plotter output messages.

The default value for the delay time is zero and is established at power-up or after receiving a DEVICE RESET command.

References

See *Communication With Other Devices* in Section 2 for a complete discussion of plotter-to-host communication.

SET BLOCK SIZE

Purpose: This command sets the maximum size of blocks that will be sent to the plotter.

SERIAL

```
          EC AD H maximum-block-size-integer
Example:  EC A H 132
```

Parameters

maximum-block-size
Maximum number of characters which will be sent in a block, 0-65535 (integer argument).

Comments

The maximum-block-size value should include room for the BLOCK START and BLOCK END commands (including the checksum and terminator).

The default value for block size is zero and is established at power-up or after a DEVICE RESET command.

References

See Section 2, *Serial Interface Concepts*, for a discussion of Block mode communications and recommendations for block size values.

SET BYPASS CANCEL CHARACTER

Purpose: This command sets the bypass cancel character (enables the Bypass function — see Appendix I, *Glossary*).

SERIAL

EC AD **U** bypass-cancel-character

Example: **EC A U** LF

Parameters

bypass-cancel-character
The bypass cancel character.

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, a typical situation is when the plotter output terminator is CR (selected by the four rear-panel switches), and the host computer echoes CR with CR LF. In this case, the bypass cancel character should be set to LF.

At power-up or after a DEVICE RESET command, no bypass cancel character is established — the plotter does not ignore echoed characters.

References

See Section 2, *Serial Interface Concepts*, for use of the bypass cancel character function.

SET SIGNATURE CHARACTER

Purpose: This command sets the signature character that the plotter prefixes to each output block.

SERIAL

$^{E}{}_{C}$ $^{A}{}_{D}$ **S** signature-character

Example: $^{E}{}_{C}$ **A S a**

Parameters

signature-character
The signature character.

Comments

This command causes output blocks from the plotter to be prefixed with the selected signature character. Some hosts use the signature character to identify the specific plotter originating an output block if more than one plotter is used in the same loop-through configuration.

At power-up or after a DEVICE RESET command, no signature character is established — output blocks will not be preceded by a signature character.

References

Refer to Sections 2 and 3, *Serial Interface Concepts* and *Introduction to Serial Commands and Responses*, for more information.

SET PROMPT CHARACTER

Purpose: This command sets the prompt character that the plotter must receive before any output message is sent to the host computer.

SERIAL

^{E}C ^{A}D **R** prompt-character

Example: ^{E}C **A R ?**

Parameters

prompt-character
The prompt character.

Comments

Some host computers can accept input from a device, such as a terminal or plotter, only at certain times. A prompt character (for example, **?**) is sent to the device to indicate input is expected (and will be accepted). The SET PROMPT CHARACTER command prevents the plotter from transmitting output messages to the host computer until the host computer sends the selected prompt character. The plotter must receive a prompt character for each output message.

At power-up or after a DEVICE RESET command, no prompt character is established — prompt mode is disabled.

References

See Section 2, *Serial Interface Concepts*, for a complete description of Prompt mode.

DEVICE COMMANDS

CONCEPTS

Some of the Device commands generate output messages. Therefore, it is important to understand the information about output messages in *Serial Interface Concepts* (Section 2) and *Introduction to Serial Commands and Responses* (Section 3).

DEVICE COMMAND DESCRIPTIONS

There are three Device commands:

- DEVICE RESET
- READ STATUS
- SIZE

Each command is described separately in the paragraphs that follow.

DEVICE RESET

Purpose: This command resets graphic, alpha, and interface parameters to power-up (default) values.

SERIAL

$^{E}{}_{C}$ $^{A}{}_{D}$ **N**

Example: $^{E}{}_{C}$ **A N**

Comments

The following is a list of actions that occur as a result of this command:

- Alpha parameters are reset to default values. This is equivalent to an ALPHA RESET command. See the description of the ALPHA RESET command under *Alpha Commands* later in this section for a complete list of actions.
- The PROMPT light is turned off (equivalent to receiving a PROMPT LIGHT OFF command).
- The output buffer is cleared.
- The PEN SPEED is reset to FULL SPEED (equivalent to a PEN SPEED command with an argument of 570 mm/second) in Option 31 equipped plotters.
- The ERROR light is turned off (if on) and the error bits in the device status register are cleared.
- The following actions also occur:
  - Prompt mode is disabled.
  - The bypass function is disabled.
  - No signature characters are sent.
  - Turnaround delay time is set to zero.
  - Block mode BLOCK SIZE is set to zero.
  - Plotter units are set to normal ADUs in plotters equipped with a Firmware Level 5 or above (see the *SELECT PLOTTER UNITS* command).

READ STATUS

Purpose: This command causes the plotter to generate an output message containing plotter status information.

SERIAL

(One Output Message Block Produced)

$^{E}_{C}$ $^{A}_{D}$ **O** status-register-selector

Example: $^{E}_{C}$ **A O 1**

Parameters

status-register-selector
Status register selector, 0-3 (see *Comments*).

Outputs

value 1
status register number, 0-3.

value 2
status information (see *Comments*).

block type
status response (see Section 3, *Introduction to Commands and Responses*, for more information).

Comments

The legal status-register-selector range is 0-3. Plotter responses are defined in Table 4-1.

Table 4-1

STATUS INFORMATION (OUTPUT VALUE 2)

Status-Register-Selector	Status Information
0	Device status
1	Interface status
2	Current selected pen (Option 31 only, 0 otherwise)
3	Available buffer space (Firmware Level 4 and up only; 0 otherwise)

Device Status

This is a 16-bit integer value in which each binary digit indicates if a particular condition is true or false. Unused bit positions are currently zero (see Table 4-2), but these bits are reserved for future use and may not always be zero. The "1" binary value for a specified position indicates that the state associated with that position is true. A zero indicates the state is not true. Table 4-2 describes each bit position.

Table 4-2

DEVICE STATUS BITS

Bit[a]	Description
15	This bit is not currently used (set to zero).
14	This bit is not currently used (set to zero).
13	Pen Select Error. This bit is set to "1" by a PEN SELECT command (either front-panel or host command) if the requested pen position is empty or if there is no empty pen position to store the pen currently in the plotter's pen carriage. This bit is reset by a READ STATUS 0 or DEVICE RESET command, or by pressing the front-panel LOAD switch.
12	Plotting Speed. This bit is set to "1" if Slow Plotting Speed is selected by either the front-panel (Option 31 equipped plotters) or rear-panel switches (plotters without Option 31). The bit is 0 otherwise.
11	$^{C}_{R}$ action. This bit is set to "1" if the rear-panel $^{C}_{R}$ action switch is set to $^{C}_{R}$ implies $^{C}_{R}$ + $^{L}_{F}$. The bit is 0 otherwise.
10	Pen Status. This bit is set to "1" if the plotter's pen is down on the paper or "0" if the pen is up.

[a] Bit-0 is the least-significant bit.

(continued)

Table 4-2 (cont)

Bit[a]	Description
9	Load mode. This bit is set to "1" if the plotter is in Load mode (the front-panel LOAD switch is pressed down and locked). The bit is "0" if the LOAD switch is up (released).
8	Page X Mirrored. This bit is set to "1" 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 "0" otherwise.
7	Page Y Mirrored. This bit is set to "1" if the page lower left y-coordinate has been defined above the page upper right y-coordinate. Bit-7 is set to "0" otherwise.
3-6	Bit-6 through Bit-3 indicate when the current pen position is outside the current page boundaries as follows: Bit-6 X Right Bit-5 X Left Bit-4 Y Above Bit-3 Y Below Left, right, above, and below are defined with respect to the horizontally oriented platen surface. These directions are not altered by a mirrored page.
2	Communication Error. A communications error, such as parity or framing error, sets this bit to "1". This bit is cleared by pressing the front-panel LOAD switch, by sending a READ STATUS WORD 0 command, or by sending a DEVICE RESET command.
1	Command/Response Error. Command/response errors occur when an invalid command, argument, or format is received. This bit is cleared by pressing the front-panel LOAD switch, by sending a READ STATUS WORD 0 command, or by sending a DEVICE RESET command.
0	Internal Error. A internal error condition sets this bit to "1".

[a] Bit-0 is the least-significant bit.

Interface Status

This is a 16-bit integer value in which Bit-11 through Bit-15 are binary digits indicating true or false. A "1" binary value for a position indicates that the state associated with that position is true. A "0" indicates the state is not true. Bit-0 through Bit-10 encode the amount of available memory in the input buffer. Table 4-3 shows the meanings of the bits.

Table 4-3

INTERFACE STATUS

Bit	Description If RS-232-C Interface Is Selected
15	Bit is not currently used (set to 0). It is reserved for future expansion and may not always be zero.
14	"1" if rear-panel COPY MODE switch is set; "0" otherwise.
13	"1" if the rear-panel DEL IGNORE switch is set; "0" otherwise.
11-12	Set according to rear-panel GIN termination selection as shown (Bit 12 is the first bit and Bit 11 is the second bit): **Bit-12 Bit-11 Termination** 0 0 none 0 1 CR 1 0 CR + ET 1 1 invalid
0-10	Indicates the amount of available buffer space (in bytes).

[a] Bit-0 is the LSB (least-significant bit).

NOTE

In plotters with Version 4 and higher, if more than 2047 bytes of space are available, this field is forced to 2047. In these versions, the actual number of free bytes is indicated in Status Word 3.

Currently Selected Pen

If Option 31 is installed, this integer value indicates the pen number currently in the pen carriage (1 to 8). If no pen is currently selected, a value of 0 is returned.

Zero is always returned if Option 31 is not installed on the plotter.

Available Buffer Space

If the plotter's firmware is Version 4 or higher, this integer value indicates the number of available bytes in the input buffer. In plotters with Version 3 and below, this value is always zero.

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

(One Output Message Block Produced)

$^{\mathrm{E}}\mathrm{C}$ $^{\mathrm{A}}\mathrm{D}$ **Q**

Example: $^{\mathrm{E}}\mathrm{C}$ **A Q**

Outputs

value 1
3850 (represents x-size x 256 + y-size where x-size equals 15 inches and y-size equals 10 inches).

value 2
Identification word (see Table 4-4).

Table 4-4

IDENTIFICATION WORD

Bit[a]	Information
0-3	Firmware release number.
4-5	See Table 4-5.
6	In plotters with Version 4 or higher, this bit is set to "1" if Option 20 (RAM expansion) is installed. This bit is zero in earlier versions.
7-13	Set to "0" (reserved for future use and must not always be assumed to be zero).
14	Set to "0" for versions earlier than Version 4. In later versions, this bit is set to "1" if the new (4663-compatible) character font is selected by the OLD-NEW strap on the System Memory circuit board, or "0" if the old 4662 character font is selected.
15	Set to "1" if Option 31 is installed; "0" otherwise.

[a] Bit 0 is the LSB (least significant bit).

References

See Section 3, *Introduction to Serial Commands and Responses*, for information on decoding output responses from the plotter.

Table 4-5

IDENTIFICATION WORD BITS 4 AND 5

Bit 4	Bit 5	Maximum Buffer Space Available in Bytes
0	0	Less than or equal to 600 bytes
0	1	Less than or equal to 1100 bytes
1	0	Less than or equal to 1600 bytes
1	1	Greater than 1600 bytes

ALPHA COMMANDS

INTRODUCTION

The plotter has an internal "alpha" (alphanumeric character) generator that enables the user to print alphanumeric characters and symbols that are defined in nine resident fonts.

The alpha generator prints the 95 ASCII characters SP through ~ and responds to the control characters BL through VT and CR.

CONCEPTS

The following 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

PRINT Command Fundamentals

For each SP through ~ character received in Alpha mode, 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 DIMENSION command).

The alpha generator prints characters from the active font according to alpha setup commands for alpha dimension and rotation. The alpha generator also processes control characters BL through VT and CR.

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 by a joystick action. If this expectation is not met, the plotter precedes the PRINT command with a MOVE-TO-HOME command (*home* is defined in the next paragraph). The automatic MOVE-TO-HOME function also occurs on the first PRINT command after pressing a front panel LOAD, LOCATE LOWER-LEFT, or LOCATE UPPER-RIGHT switch.

The Home Position

The *home* position is a point located one character height below the upper-left corner of the page. (If the alpha rotation angle is non-zero, the home position is rotated by this angle about the upper-left corner of the page.)

Setting the Size and Spacing of Printed Characters

The SET ALPHA DIMENSION command is used to specify the character space and line space dimensions. These dimensions are directly related to character width and character height, as illustrated for the PRINT command in Figure 4-1. Default dimensions are described under the SET ALPHA DIMENSION command.

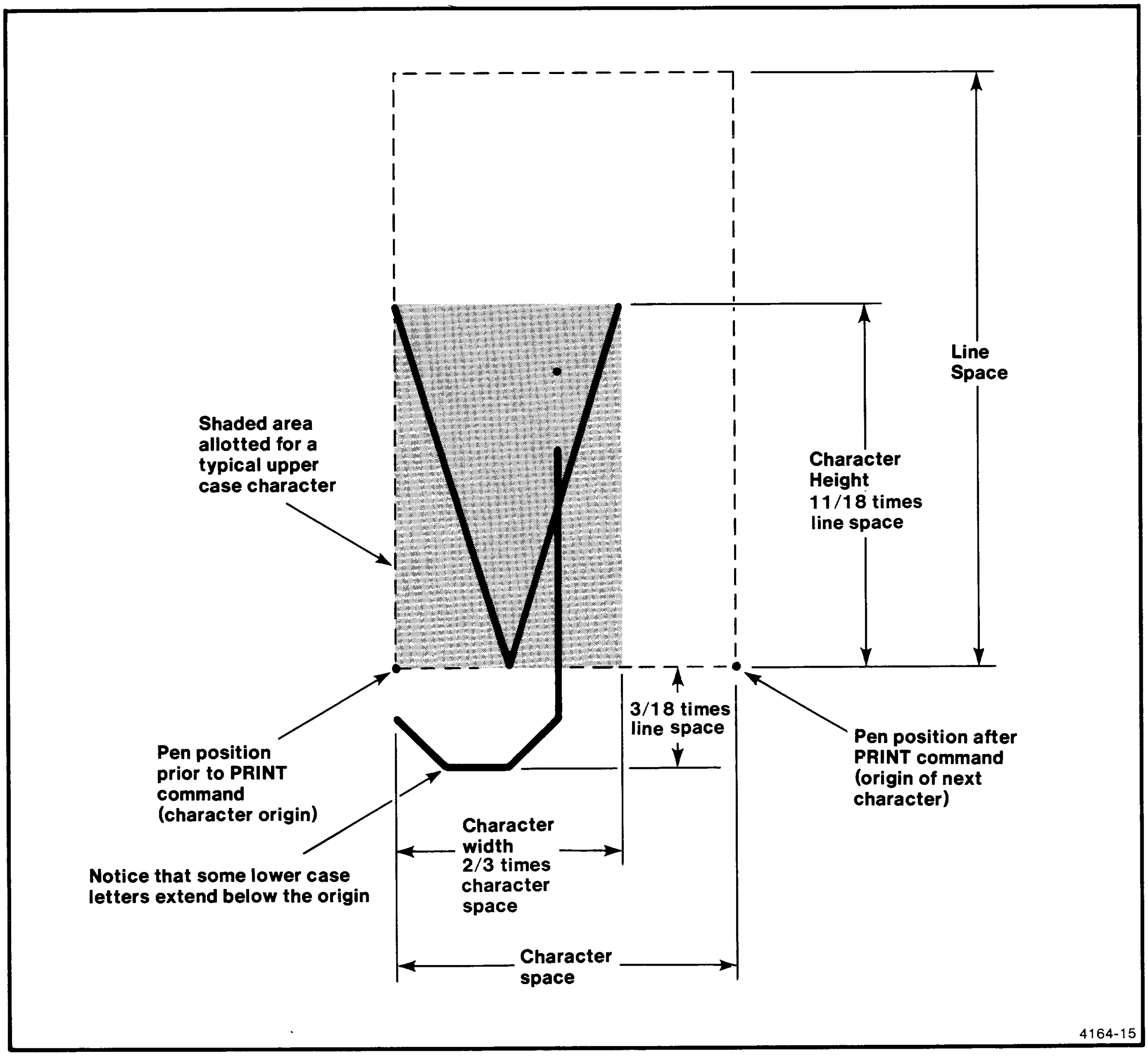
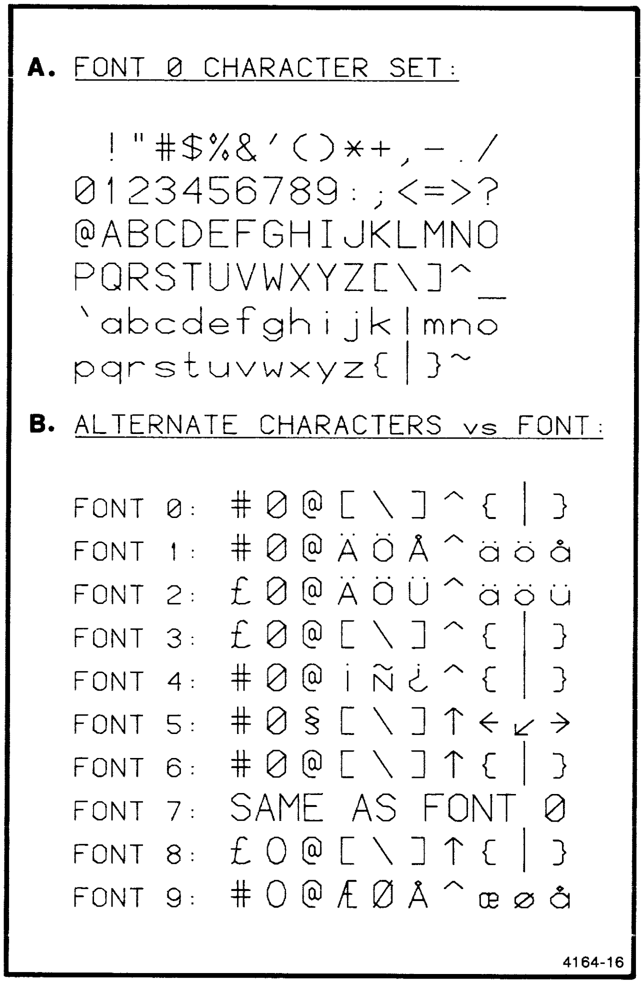

Line Space
Character Height 11/18 times line space
Shaded area allotted for a typical upper case character
3/18 times line space
Pen position after PRINT command (origin of next character)
Pen position prior to PRINT command (character origin)
Notice that some lower case letters extend below the origin
Character width 2/3 times character space
Character space
4164-15

Figure 4-1. Alpha Size and Spacing for the PRINT Command.

Resident Alpha Fonts

The alpha generator has resident character definitions for ASCII characters ˢᴾ through ~ in nine fonts.

Figure 4-2A shows the entire Font 0 (default) character set. All other fonts are identical to Font 0 except for the character substitutions shown in Figure 4-2B.



A. FONT Ø CHARACTER SET:

! "#$%&'()*+,-./
Ø123456789:;<=>?
@ABCDEFGHIJKLMNO
PQRSTUVWXYZ[\]^_
`abcdefghijklmno
pqrstuvwxyz{|}~

B. 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 ^ œ ø å

4164-16

Figure 4-2. Resident Alpha Fonts.

NOTE

Plotters with a Firmware Level of 3 or less have slightly different character substitutions in Fonts 1 through 6 as shown in Figure 4-3.

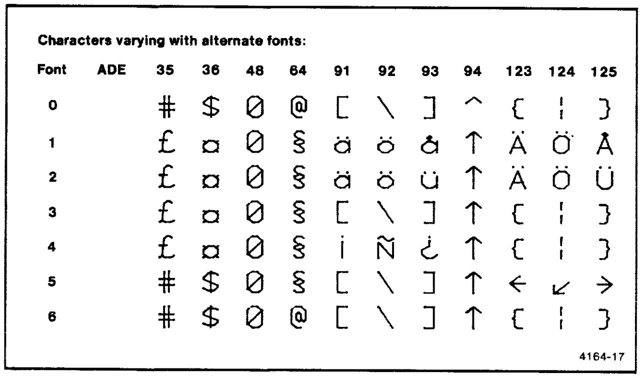
An OLD-NEW strap is located on the System Memory circuit board in newer plotters to obtain Fonts 1 through 6 used in early model plotters. Figure 4-3 shows the old fonts.

ALPHA COMMAND DESCRIPTIONS

There are six Alpha commands:

- SET ALPHA DIMENSION
- SET ALPHA ROTATION
- SELECT ALPHA FONT
- PRINT
- RESET ALPHA PARAMETERS
- MOVE-TO-HOME

Each is described separately in the following pages.



Characters varying with alternate fonts:

Font	ADE	35	36	48	64	91	92	93	94	123	124	125
0		#	$	Ø	@	[	\	]	^	{	¦	}
1		£	¤	Ø	§	ä	ö	å	↑	Ä	Ö	Å
2		£	¤	Ø	§	ä	ö	ü	↑	Ä	Ö	Ü
3		£	¤	Ø	§	[	\	]	↑	{	¦	}
4		£	¤	Ø	§	¡	Ñ	¿	↑	{	¦	}
5		#	$	Ø	§	[	\	]	↑	←	↙	→
6		#	$	Ø	@	[	\	]	↑	{	¦	}

4164-17

Figure 4-3. "Old" Style Resident Alpha Fonts.

SET ALPHA DIMENSION

Purpose: This command specifies the character space and line space.

SERIAL

EC AD **I** character-space,line-space

Example: EC **A I 112,264**

Parameters

character-space
X-axis distance from the beginning of a letter to the beginning of the next (in ADUs).

line-space
Y-axis distance from the baseline for one line of printed characters to the next baseline above it (in ADUs).

Comments

Default values are shown in Table 4-6. These occur at power-up or after a DEVICE RESET or RESET ALPHA PARAMETERS command.

Examples

The example in the syntax box causes all subsequent printing and the associated spacing to be twice as wide and three times as high as the default character dimensions.

Table 4-6

DEFAULT ALPHA DIMENSIONS (SERIAL INTERFACE)

Character Width	Character Height	Character Space	Line Space
37 1/3 ADUs	53 7/9 ADUs	56 ADUs	88 ADUs

References

Refer to Figure 4-1. The width and height of a character is 2/3 times and 11/18 times the character space and line space values respectively.

SET ALPHA ROTATION

Purpose: This command (1) sets the alpha rotation angle specified in degrees and (2) sets the left margin (as a line through the current pen carriage position perpendicular to the printing direction) used by subsequent alpha commands.

SERIAL

EC AD **J** rotation-angle

Example: EC **A J 30**

Parameters

rotation-angle
The angle of rotation, counterclockwise, in degrees.

Comments

The default rotation angle is 0 degrees, and the default left margin is the left side of the page. The left margin is used for PRINT CR.

Examples

The example in the syntax box shows how to specify a 30° counterclockwise rotation angle, and how to set the left margin at the current pen carriage position. Figure 4-4 shows the effect of a 30° alpha rotation on subsequent PRINT commands.

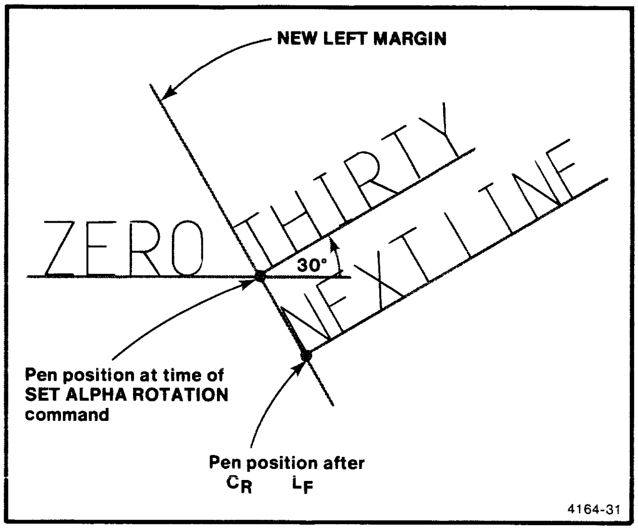


Figure 4-4. Alpha Rotation of 0° and 30°.

SELECT ALPHA FONT

Purpose: This command selects the font used by subsequent PRINT commands.

SERIAL

$^{E}_{C}$ $^{A}_{D}$ **T** font-selector

Example: $^{E}_{C}$ **A T 1**

Parameters

font-selector
font selector, 0-15

References

Refer to *Resident Alpha Fonts* earlier in this section.

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

(Use Style II Alpha Mode)

Example: **ABCDefg**

Comments

The Alpha mode (Style II command form) is used to print ASCII characters SP through ~, BL through VT and CR. Use US to enter Alpha mode. The alpha generator prints each character SP through ~ according to current ALPHA DIMENSION, ROTATION, and FONT command parameters. Control characters are processed as listed in Table 4-7 (other control characters are ignored).

Examples

The example in the syntax box shows how to print the first seven letters in the alphabet: ABCDefg, including lowercase.

References

See Section 3, *Introduction to Serial Commands and Responses*, for more information on printing characters.

Table 4-7

CONTROL CHARACTER RESPONSES

Character	Action
BL	Rings the bell.
BS	Moves the pen carriage left one character space.
HT	Treated as a SP.
LF	Moves the pen carriage down one line space.
VT	Moves the pen carriage up one line space.
ECFF	See the MOVE-TO-HOME command.
	Individual FF characters received in Alpha mode are ignored.
CR	Moves the pen carriage to the left margin (and generates a LF if CR GENERATES LF is selected by the four rear-panel switches).

RESET ALPHA PARAMETERS

Purpose: This command resets alpha parameters to their default (power-up) values.

SERIAL

$^{E}_{C}$ $^{A}_{D}$ **V**

Example: $^{E}_{C}$ **A V**

Comments

The RESET ALPHA PARAMETER command establishes default parameters for the alpha setup comands:

- Sets alpha rotation to zero degrees
- Selects alpha font 0
- Resets alpha dimensions as shown in Table 4-6
- Sets the left margin to the left page boundary.

MOVE-TO-HOME

Purpose: This command moves the pen carriage to the home position and sets the plotter in Style II Alpha mode. The home position is a point located one character height below the upper-left corner of the page. (If the alpha rotation angle is non-zero, the home position is rotated by this angle about the upper-left corner of the page.)

SERIAL

(Style II Command)

$^{E}_{C}$ $^{F}_{F}$

Comments

This command moves the pen carriage to the home position (the height of a typical uppercase character from the upper-left corner of the page, perpendicular to the printing direction).

GRAPHICS COMMANDS

INTRODUCTION

The following pages include information about standard Graphics commands, including:

- Graphics concepts
- the MOVE command
- the DRAW command

CONCEPTS

The MOVE and DRAW commands are sent to the plotter using Style II Graph mode coding. The overall plotter command coding scheme (including Graph mode) is described fully in Section 3 and briefly here.

$^{G}_{S}$ selects Graph mode. When Graph mode is selected, the first XY coordinate pair represents a "move" and subsequent XY coordinate pairs represent "draws." (Refer to Section 3 for details on encoding or decoding the XY coordinates.) Move and draw coordinates are expressed in ADUs.

In using graphics commands, you may want to refer to Appendix E, which provides additional information on area fill patterns.

GRAPHICS COMMAND DESCRIPTIONS

There are two Graphics commands:

- MOVE
- DRAW

The following pages describe these commands separately.

MOVE

Purpose: This command causes the pen carriage to move (with the pen up) to the specified position.

SERIAL

(Use Style II Graph Mode Coding)

Comments

The pen moves to the point specified. If either coordinate is outside the page boundary (or both coordinates), the pen moves to the point on the page boundary that is nearest to the specified point.

Use Style II Graph mode. Use ^{G}s to enter Graph mode.

The first point (coordinate pair) after the ^{G}s character specifies a Move. Coordinates are specified in ADUs (see Appendix F for a coordinate conversion chart).

DRAW

Purpose: The DRAW command causes the plotter to draw a straight line from the current position to the specified position.

SERIAL

(Use Style II Graph mode coding)

Comments

The pen draws a line from the current pen position to the point specified. If either the new or previously specified point is outside the page boundary, the DRAW command is treated as a MOVE command. If either coordinate is outside the page boundary (or both coordinates), the pen moves to the point on the page boundary that is nearest to the specified point.

Use Style II Graph mode. All coordinate pairs after the first are draws. Coordinates are specified in ADUs (see Appendix F for a coordinate conversion chart.

NOTE

The first XY coordinate pair after the $^{G}\mathrm{S}$ *ASCII character can be used for a draw if a* $^{B}\mathrm{L}$ *ASCII character follows the* $^{G}\mathrm{S}$.

DIGITIZING COMMANDS

Introduction

The plotter's digitizing commands allow the host computer to request and obtain coordinates of (1) the current pen position or, (2) points designated by the operator using the front-panel CALL switch.

Current pen position information is obtained using the DIGITIZE command. Operator designated points are obtained using the "call digitize" procedures described later. Remember that the CALL switch is permanently enabled and, therefore, the operator must be instructed when to use the CALL switch.

Concepts

Call digitizing is the process by which the operator presses the front-panel CALL switch to send pen coordinates to the host.

A call digitizing process has these steps:

1. The PROMPT LIGHT ON command turns on the PROMPT light, which indicates to the operator that digitizing is permitted.
2. a. For each point, the operator uses the joystick to position the digitizing reticle (used in place of a pen in the pen carriage) over the desired point. The operator then presses the front-panel CALL switch. This instructs the plotter to store the point.

   b. (Optional) If the operator chooses, he or she may press and hold the CALL switch until the plotter's bell rings (which sends a "status block" to the host) and indicates the end of digitizing. (This action assumes that the host knows the meaning of the status block.)
3. For each point, the plotter sends one digitizing output block of information consisting of X and Y position values and a value indicating whether the pen is up or down.

   If Step 2b is used, the plotter sends a status block rather than position information.

   If this is NOT the last point, the process continues at Step 2.
4. The PROMPT LIGHT OFF command turns off the PROMPT light, thus telling the operator to stop digitizing.

This completes the call digitizing process.

Digitizing Command Descriptions

There are three Digitizing commands:

- DIGITIZE
- PROMPT LIGHT ON
- PROMPT LIGHT OFF

Each command is described separately in the paragraphs that follow.

DIGITIZE

Purpose: This command causes the plotter to send one output message block containing current X and Y position values and a pen up/down status value.

SERIAL

(One Output Message Block Produced)

$^{E}{}_{C}$ $^{A}{}_{D}$ **M**

Example: $^{E}{}_{C}$ **A M**

Outputs

value 1
X-position

value 2
Y-position

pen-status
0 = pen up
1 = pen down

References

Refer to Section 3, *Introduction to Serial Commands and Responses*, for more information on decoding output responses from the plotter.

PROMPT LIGHT ON

Purpose: This command turns on the front-panel PROMPT light.

SERIAL

$^{E}{}_{C}$ $^{A}{}_{D}$ **K**

Example: $^{E}{}_{C}$ **A K**

Comments

When using the Serial Interface, 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 *Communicating with Other Devices* in Section 2, *Serial Interface Concepts*.

On Option 31 equipped plotters, the blinking PAUSE light (set by pressing the front-panel PAUSE switch) overrides the PAUSE light established by this command. The PAUSE function causes the PROMPT light to blink at a one-second rate. The PROMPT light returns to the correct condition when RESUME is pressed.

The PROMPT light is turned off and Prompt mode is exited by a PROMPT LIGHT OFF or a DEVICE RESET command.

References

See the PROMPT LIGHT OFF command description.

PROMPT LIGHT OFF

Purpose: This command turns the front-panel PROMPT light off.

SERIAL

$^{E}{}_{C}$ $^{A}{}_{D}$ **L**

Example: $^{E}{}_{C}$ **A L**

Comments

Refer to the PROMPT LIGHT ON command.

References

See the PROMPT LIGHT ON command description.

MULTIPLE PEN (OPTION 31) COMMANDS

INTRODUCTION

The Multiple Pen option (Option 31) adds an eight-pen capability to the plotter. With this option, you can make multi-colored plots. Although the front-panel switches allow the operator to select pens, the host can also select pens under program control. In addition, the host can program the plotter to plot at the best speed for a pen to match its ink flow characteristics to the media used. The programmed speed is the pen's terminal velocity on long vectors (see Performance Specifications in the *4662 Interactive Digital Plotter Operator's Manual*) and is measured in mm/second.

The SELECT PLOTTER UNITS command allows the host to address the plotter's mechanical resolution. This allows improved solid area fills on overhead transparency films in critical applications.

CONCEPTS

The SELECT PEN command permits the host to select one of eight pens stored in the rotary pen turret of Option 31 equipped plotters.

The PEN SPEED command permits the host to select a terminal velocity for the current pen. (This terminal velocity is in effect after the pen's initial acceleration and prior to its deceleration at the end of a vector.) The pen's terminal velocity may be selected to match the particular ink flow characteristics of the pen/media used. In plotters with Firmware Levels of 5 and less, host-selected terminal speeds of less than 220 mm/second automatically set the plotter to SLOW acceleration regardless of the current front-panel SLOW or FAST (acceleration) switch settings. On the other hand, if the host-selected terminal speed is greater than 220 mm/second, the current front-panel SLOW or FAST (acceleration) switch setting becomes active. Therefore:

- For speeds less than 220 mm/second, only SLOW acceleration is used (available in plotters with Firmware Level 5 and above).
- For speeds greater than 220 mm/second, SLOW or FAST acceleration can be chosen.

The SELECT PLOTTER UNITS command (available in Option 31 equipped plotters with Firmware Versions 5 or greater) is primarily designed for plots using solid-area fill patterns. Selecting the plotter units allows the host to chose the default plotter coordinate system or a modified plotter coordinate system.

The default addressing units for the plotter are 0 to 4095 ADUs over the length of the X-axis of the page and 0 to 2730 ADUs over the length of the Y-axis of the page. This ratio matches a 10x15 inch ratio of the Y-axis to the X-axis. However, the mechanical resolution is 0.005 inches, making 3000 addressable points along the 15-inch X-axis and 2000 addressable points along the 10-inch Y-axis. The uneven mapping of the addressing units into the mechanical resolution units of the plotter makes it difficult to step the pen in small, even increments that are necessary to perform high-quality solid area fills on transparency film.

The modified plotter coordinate system allows the host to address each of the 3000 points in the plotter's X-axis exactly. Addressing each of the 3000 points (or the 2000 points in the Y-axis) eliminates the very slight shift of some area fill lines when the plotter addresses the normal 2730 by 4096 ADU coordinate system. In the normal 2730 by 4096 ADU coordinate system, the plotter rounds off calculations to plot some of the 4096 points in its normal X-axis (the same is true in the Y-axis). The result is that sometimes unwanted shading patterns develop in solid-area fill patterns. The effect is not noticeable in other plots. See Appendix E for more information on area fill patterns.

NOTE

The ability of the host to exactly address the 3000 points of the plotter's maximum resolution only occurs for a default page size. If the page has been altered by the front-panel SET PAGE switches, the uniform solid-area fill effect can not be guaranteed.

MULTIPLE PEN (OPTION 31) COMMAND DESCRIPTIONS

Three Option 31 commands are described here:

- SELECT PEN
- PEN SPEED
- SELECT PLOTTER UNITS

SELECT PEN

Purpose: This command causes the plotter to select the specified pen from the rotary pen turret or to store the current pen in the rotary pen turret.

SERIAL

EC AD **BP** pen-selector

Example: EC **A BP 1**

Parameters

pen-selector

pen position in the rotary pen turret 1 through 8, or 0, which means store the pen presently in the pen carriage in the rotary pen turret.

Comments

The pen position (selector) matches the pen numbers printed on the rotary pen turret cover. Out-of-range arguments generate a command/response error.

Examples

The example in the syntax box causes the pen carriage to deposit any pen it has and to pick up the pen located in Pen Position 1 of the rotary pen turret.

PEN SPEED

Purpose: This command causes the plotter to limit its terminal velocity to a value less than or equal to the specified value.

SERIAL

EC AD **BY** pen-speed

Example: EC **A BY 280**

Parameters

pen-speed

terminal velocity of the pen, in mm/second. Ranges from 10 to 570 mm/second, in steps of 10 mm/second.

Comments

Pen speeds are rounded to the nearest 10 mm/second value. This gives 57 different speeds ranging from 10 mm/second to 570 mm/second. Negative and zero values generate a command/response error and cause the command to be ignored.

Values greater than 570 mm/second are treated as 570 mm/second.

In plotters with a Firmware Level 5 or higher, the plotter's acceleration is reduced to one-half of normal to improve line quality whenever speeds of less than 220 mm/second are programmed.

The default pen speed value at power-up or after a DEVICE RESET command is 570 mm/second.

Examples

The example in the syntax box causes the plotter to operate at approximately one-half speed.

SELECT PLOTTER UNITS

Purpose: This command selects the plotter coordinate system.

SERIAL

EC AD **BV** units-selector

Example: EC **A BV 3**

Parameters

units-selector: (0 or 3)

- 0 selects the normal 4096 by 2730 ADUs (X-axis by Y-axis) plotter coordinate system. In copy mode, the coordinate system is 4096 by 3124 ADUs.
- 3 selects the modified 3000 by 2000 ADUs (X-axis by Y-axis) plotter coordinate system (3000 by 2288 ADUs in Copy mode).

Comments

The physical size of a single modified ADU is 0.00366 inches (.093 mm) using a default page size.

The default plotter corrdinate system selector is 0 and is set at power-up or after a DEVICE RESET command.

Examples

The example in the syntax box causes the plotter to select the modified plotter coordinate system.

References

See *Concepts* (under *Multiple Pen Commands*) for a description of the use of this command.

Section 5

GPIB INTERFACE CONCEPTS

INTRODUCTION

This section describes five aspects of the GPIB Interface which must be considered before GPIB devices can communicate with the plotter. These are:

- Connecting the plotter to other GPIB devices,
- Setting the four rear-panel switches to allow other GPIB devices to communicate with the plotter,
- Controlling the transmission of information to and from the plotter,
- Formatting the various plotter commands,
- 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-l978 for programmable measuring apparatus. This standard defines the mechanical and electrical characteristics of the bus, as well as procedural characteristics (such as how to send data bytes to and from the interface). Standardization allows the plotter to be easily installed in systems with various devices communicating through the GPIB. However, the GPIB information (consisting of commands and responses that are transferred 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 Section 6.

NOTE

The plotter's GPIB Interface does exhibit some exceptions to the IEEE Standard 488-1978. However, the plotter is designed to work with other devices that are manufactured by Tektronix's Information Display Division. Proper operation cannot be assured with other GPIB devices except those manufactured by Tektronix's Information Display Division.

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

- The total length of GPIB cable in the system must either be 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 should 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.

Figure 5-1 illustrates a typical system configuration showing the flexibility of GPIB. More than one plotter may be connected in the system if desired.

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

SETTING THE FOUR REAR-PANEL SWITCHES

Settings on the four rear-panel switches (associated with the GPIB Interface) permit the plotter to operate in a variety of communications environments. See the discussions *Rear Panel Controls (GPIB Switch Settings)* and *Understanding Operation With the GPIB Interface* located in the 4662 Interactive Digital Plotter Operator's Manual.

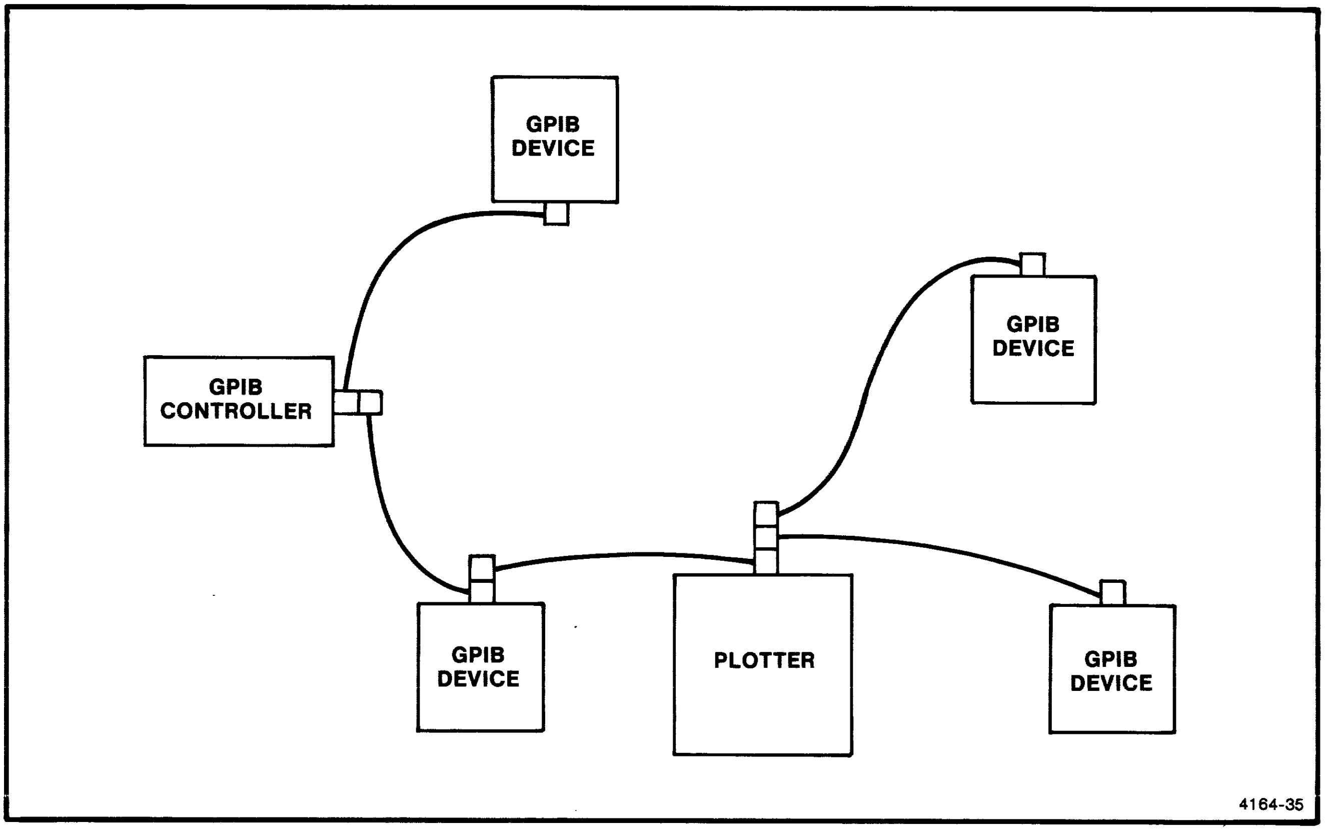


Figure 5-1. Typical GPIB System Configuration.

COMMUNICATING WITH OTHER DEVICES

The following describes how other devices instruct the plotter to transmit or receive data bytes, and how the plotter interacts with a GPIB controller. These actions are described both in terms of the GPIB functions required by the plotter interface and, for the example of TEKTRONIX 4050 Series controllers, the typical commands the controller uses to command the required GPIB functions. Refer to the controller operator's manual to see how GPIB functions are commanded.

GPIB OPERATION

The GPIB Interface allows data bytes, consisting 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.

The designation of the talker and listener devices is done under control of a system controller. The controller utilizes special bus functions to designate the desired talker and listener(s), to initiate the transmission of data bytes between devices, and to determine the status of the devices in the system. Examples of devices with this controller capability are the TEKTRONIX 4050 Series Graphic Computing Systems. For systems that do not have a controller, the plotter's four rear-panel switches manually designate the plotter as a talker and listener and initiate data byte transfers.

LOCAL SWITCHING FUNCTION

The LOCAL switch controls the plotter's response to messages on the GPIB. The choices provided by this switch are:

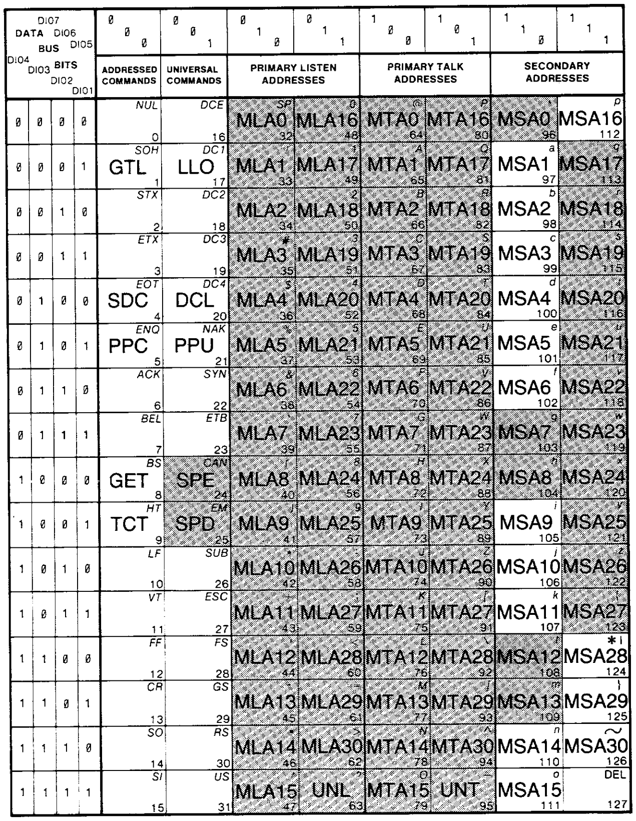
- LOCAL Switch UP (On-Line Remote) — This is the normal mode of operation. The plotter receives all bus messages and executes received commands. Any action of the LOCAL switch clears the output buffer.
- LOCAL Switch Down (Off-Line Remote) — The plotter ignores all GPIB messages. This mode is logically equivalent to removing the GPIB cable from the plotter.

DEVICE-TO-PLOTTER TRANSMISSION

The following pages 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 primary address which must correspond to the primary address selected from the four rear-panel switches. Refer to Figure 5-2 for a chart of GPIB message bytes.



DATA BUS BITS DI07											
DI06											
DI05											
DI04	DI03	DI02	DI01	0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1
				ADDRESSED COMMANDS	UNIVERSAL COMMANDS	PRIMARY LISTEN ADDRESSES		PRIMARY TALK ADDRESSES		SECONDARY ADDRESSES	
0	0	0	0	NUL 0	DCE 16	SP MLA0 32	0 MLA16 48	@ MTA0 64	P MTA16 80	` MSA0 96	p MSA16 112
0	0	0	1	SOH GTL 1	DC1 LLO 17	! MLA1 33	1 MLA17 49	A MTA1 65	Q MTA17 81	a MSA1 97	q MSA17 113
0	0	1	0	STX 2	DC2 18	" MLA2 34	2 MLA18 50	B MTA2 66	R MTA18 82	b MSA2 98	r MSA18 114
0	0	1	1	ETX 3	DC3 19	# MLA3 35	3 MLA19 51	C MTA3 67	S MTA19 83	c MSA3 99	s MSA19 115
0	1	0	0	EOT SDC 4	DC4 DCL 20	$ MLA4 36	4 MLA20 52	D MTA4 68	T MTA20 84	d MSA4 100	t MSA20 116
0	1	0	1	ENQ PPC 5	NAK PPU 21	% MLA5 37	5 MLA21 53	E MTA5 69	U MTA21 85	e MSA5 101	u MSA21 117
0	1	1	0	ACK 6	SYN 22	& MLA6 38	6 MLA22 54	F MTA6 70	V MTA22 86	f MSA6 102	v MSA22 118
0	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	0	0	0	BS GET 8	CAN SPE 24	(MLA8 40	8 MLA24 56	H MTA8 72	X MTA24 88	h MSA8 104	x MSA24 120
1	0	0	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	0	1	0	LF 10	SUB 26	* MLA10 42	: MLA26 58	J MTA10 74	Z MTA26 90	j MSA10 106	z MSA26 122
1	0	1	1	VT 11	ESC 27	+ MLA11 43	; MLA27 59	K MTA11 75	[MTA27 91	k MSA11 107	{ MSA27 123
1	1	0	0	FF 12	FS 28	, MLA12 44	< MLA28 60	L MTA12 76	\ MTA28 92	l MSA12 108	*| MSA28 124
1	1	0	1	CR 13	GS 29	- MLA13 45	= MLA29 61	M MTA13 77	] MTA29 93	m MSA13 109	} MSA29 125
1	1	1	0	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 4662 Plotter.

- 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

- KEY

CAN	ASCII Character
SPE	GPIB Code
24	Decimal

*| on some keyboards or systems

(3806) 4164-18

Figure 5-2. GPIB Code Chart.

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. Data bytes (DABs) containing the command code and arguments are transmitted from the controller 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 instructs the 4050 Series controller to send the ATN and MLA 1 bus messages at the same time, which configures a plotter having device address 1 to be the listener. The controller then automatically becomes the talker when the data bytes **D50,50** are transmitted through the GPIB to the plotter. As long as the plotter remains enabled as a listener, several commands in succession may be transmitted from the talker 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 talker could be a device other than the 4050 Series Graphic System, such as a GPIB mass storage device. In this case, the controller enables the mass storage device as the talker and commands it to transfer a plot to the plotter. When the transfer is completed, the controller cancels the talker and listener functions.

NOTE

The 4050 Series Graphic Computing System functions as both a controller and as a talker. Refer to the Print and WByte statements and to the Interfacing Information appendix (located in the 4050 Series Graphic Computing System Reference Manual) for a further discussion of these actions. The actual transmitted data bytes that define a specific command are discussed in Section 6, Introduction to GPIB Commands and Responses.

MSA Commands From a Controller. An alternative for transmitting the DRAW command from the 4050 Series Graphic Computing System is:

DRAW@1:50,50

This statement instructs the 4050 Series controller to send the ATN and MLA 1 bus messages at the same time, followed by the MSA 20 (my secondary address) message while the ATN message is still true. The MLA 1 message configures 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 in the previous example. This non-standard use of the MSA message is unique to the 4050 Series Graphic Computing System. Refer to Section 6, *Introduction to GPIB Commands and Responses*, for a description of the plotter commands having equivalent MSA definitions. The remainder of the statement is processed the same as the previous **PRINT@1,32:** statement, which transferred the DRAW arguments of **50,50** to the plotter.

Off-Line Plotting. A plot can also be transferred to a plotter without needing a controller connected to the GPIB. A typical system involving a TEKTRONIX 4924 Digital Cartridge Tape Recorder and a plotter is shown in Figure 5-3. To transfer a plot, the plotter is enabled as a listener by selecting LISTEN ONLY on the four rear-panel switches. Then, the desired tape file is 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 ON LINE button should be in the released or up position). The contents of the desired tape cartridge file are transferred to the plotter.

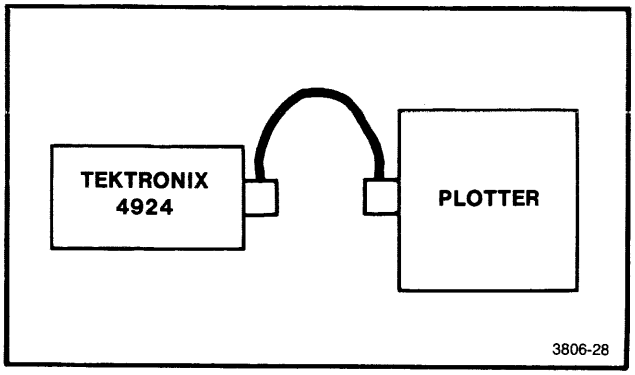


Figure 5-3. Typical OFF-LINE Configuration.

NOTE

When the plotter is in LISTEN ONLY mode, it does not recognize MLA or MTA messages. The plotter will, therefore, 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.

NOTE

Only DAB command forms may be sent to the plotter in LISTEN ONLY mode. MSA forms are not allowed.

Input Buffering

As commands are received from the talker, they are placed temporarily in an input buffer in 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 pages describe:

- How the input buffering function operates,
- How much memory is needed to store a plotter command, and
- How much memory is available under various conditions to store commands.

Buffer Operation. As commands are received from the talker, they are not executed immediately, but are stored in the plotter's input buffer. The only exception to this is the DEVICE RESET (IFC) command, which as well as being stored in the input, also immediately resets the plotter's GPIB Interface functions.

To maximize the data transfer rate on the bus, the plotter transfers data received from the GPIB bus into its input buffer any time it is not actually processing stored commands. If the buffer becomes full (i.e., commands are transferred to the plotter faster than the plotter can execute them), data transfer on the GPIB bus pauses until the input buffer space becomes available.

Storage Requirements for Commands. All commands are received by the Plotter in the various command forms described in Section 6, *Introduction to GPIB Commands and Responses.*

Commands are normally stored internally in the same form they are received (including separators and terminators), using one byte of storage for each received character. Exceptions to this are:

- Listen Address, Talk Address, Unlisten, and Untalk commands are processed directly by the interface and are not stored.
- A character sent with EOI true requires an extra byte of storage for a termination command which is generated internally.
- Each secondary address command requires one byte of storage.
- An Interface Clear (IFC) command requires one byte of storage for an internally generated command.

Memory Limits for the Input Buffer.

NOTE

The amount of memory available to store input data can be determined by using the READ STATUS command. In the standard plotter with 2K bytes of memory, about 1400 bytes are available to store received data. Option 20, RAM Memory Expansion, increases the input memory by 6000 bytes for a total of 7400 bytes.

NOTE

The plotter's input buffer is cleared by pressing the LOCAL or LOAD switches.

PLOTTER-TO-DEVICE TRANSMISSION

The plotter transmits several types of information to other devices. The following topics discuss this 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 (responses to plotter commands) is transmitted in the form of output blocks. Each output block contains one or three numeric values.

One output block is generated for each command that causes the plotter to generate an output. Only one output can be buffered by the plotter at a time. If a second command is received that causes the plotter to generate an output before the host can read the first output, the plotter will overwrite the second output over the first, thereby losing the first output. The only exception to this rule is call digitizing (CALL GIN), where up to two or three points can be digitized and stored. This is because the output generated by call digitizing is buffered separately from other host-requested output. Therefore, operator-generated output is not destroyed by host-generated output or vice-versa. Call digitizing output can only be read by the controller-generated CALL DIGITIZE command.

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 which must correspond to the plotter's primary address set by the four rear-panel switches. Refer to Figure 5-2 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 data bus bytes. The plotter transmits one output block from the output buffer followed by an output termination sequence of $^{C}_{R}$ + $^{L}_{F}$ (Carriage Return and Line Feed). In addition, if call digitizing is taking place and the front-panel CALL switch is held down until the plotter's bell sounds once, a status message of a single data value is sent instead of three values, which normally indicate pen coordinates. This difference can be used to indicate the end of a set of digitized points to the controller. Transmission then stops until another MTA bus message is received. If the output buffer is empty, the bus pauses until output is generated.

The output termination sequence tells 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 processing the data).

One difference occurs if the plotter is set for talk-only mode. In this case, if the CALL switch is held until the bell sounds once, an internal flag is set which causes the plotter to send the EOI message true along with the last byte of next digitized point, instead of the status block. This is used to indicate the completion of a digitizing task to the listening device so that the listening device can perform any end of transmission tasks (such as closing files on a Tektronix 4924).

On-Line Digitizing Example. To cause a single digitize response to be generated and transferred to a 4050 Series Graphic Computing System, the following BASIC statements would be executed by the Graphic System:

```
PRINT@1,32:"G";
INPUT@1,32:X,Y,Z
```

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. Then, 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 current pen status (up or down).

The three values in the output block are converted into ASCII Decimal Floating Point numbers and are entered into the output buffer. The 4050 Series controller reads these data bytes and assigns values of the numbers to the three target variables (X, Y, and Z) in the INPUT statement. The 4050 Series controller then sends the UNT and the ATN bus message to cancel the plotter talk function. Manually digitized coordinates are returned in a similar fashion by pressing the plotter's CALL switch and having the host retrieve the pen's coordinates by a CALL DIGITIZE command.

Off-Line Digitizing Example. Manually digitized coordinates could also be transferred from the plotter to an off-line storage device 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 5-3. To transfer digitized coordinates, the plotter is enabled as a talker by selecting TALK ONLY by the four rear-panel switches. 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 ON LINE button should be in the up or released position). Using the plotter's front-panel CALL switch, digitized coordinates are generated, transferred to the 4924, and stored on the tape cartridge. Before the last coordinate is digitized, the front-panel CALL switch is held down until the plotter's bell sounds once. The CALL switch then is pressed again momentarily to digitize the last point. This causes the plotter to send EOI true at the end of the output block, causing the 4924 to close the tape file.

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 controller functions that set up and terminate these exchanges.

Interrupting Data Exchanges

The 4662 plotter does not allow interruption of data transfers as described in the IEEE 488-1978 GPIB specification. Once the plotter is addressed as a listener or talker, a complete command or response message must be transferred before the controller can again assume control of the GPIB bus.

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 non-active 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. IFC also causes the plotter to internally generate a DEVICE RESET command.

The plotter requires about 1 ms to respond to an IFC message, which is somewhat longer than the maximum response time specified in the IEEE 488-1978 specification. No other messages should be sent during this time.

DCL/SDL Actions (Device Clear/Selected Device Clear)

The 4662 does not respond to DCL and SCL messages.

SRQ (Service Request) and Serial Poll Actions

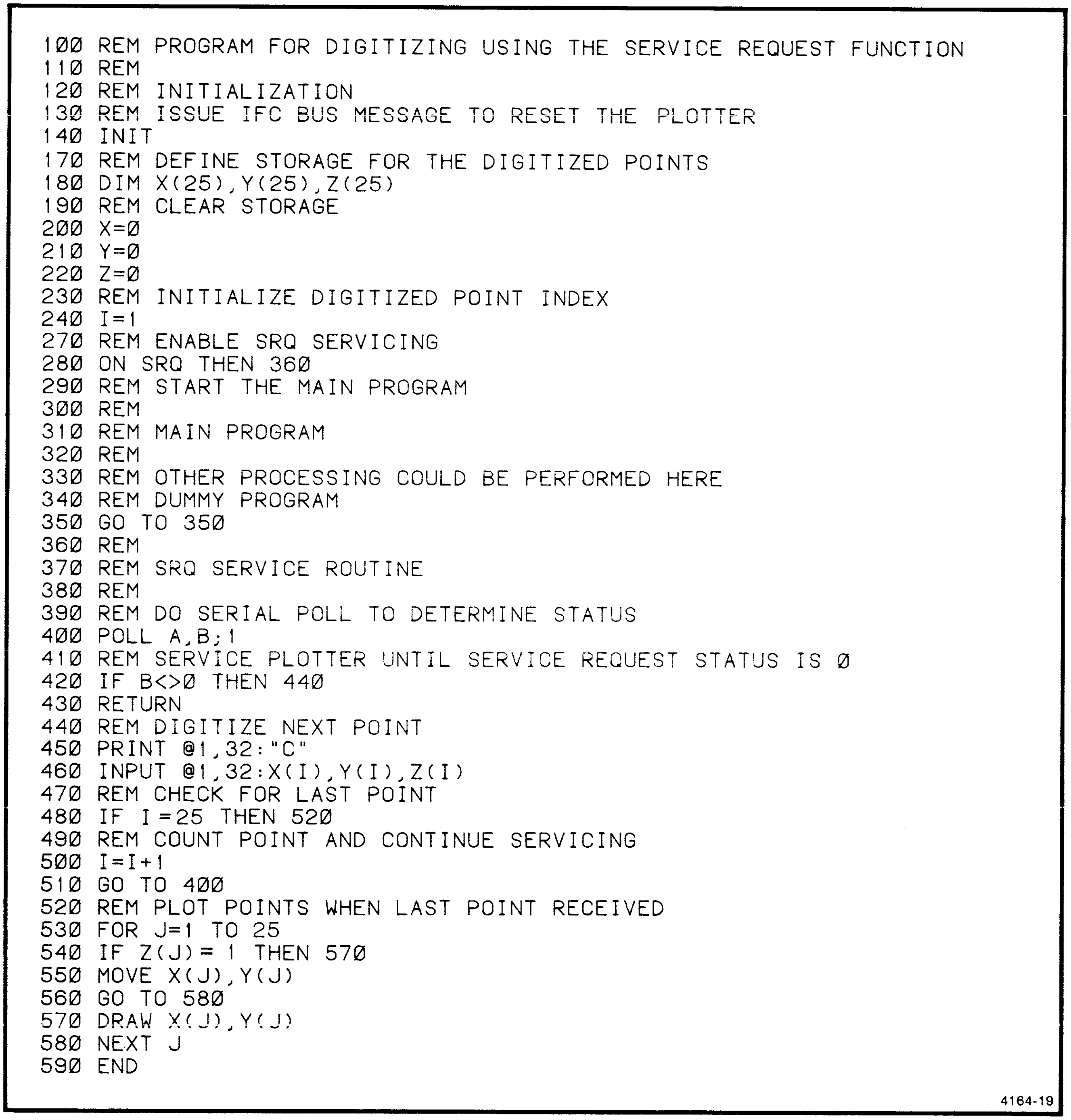
Another general function of the GPIB controller is to respond to asynchronous requests for service from devices on the GPIB. Devices on the GPIB can request service by sending a SRQ message. If the controller has enabled this function, the SRQ message interrupts the controller's current task. The controller 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 may be sent to indicate what type of service the device is requesting. As soon as a serial poll response byte containing the active service request bit is read, the controller stops SERIAL POLL commands. When the plotter receives a SERIAL POLL command, it resets its SRQ message and the service request bit in its serial poll response byte. If, after the plotter has transmitted the data block which generated the SRQ message, more data remains to be transmitted, the plotter again asserts SRQ. It also sets the service request bit in its serial poll response byte.

NOTE

This function is slightly different than specified in IEEE Standard 488-1978, which indicates that a device should continue to send SRQ and keep its service request bit in the serial poll response until the condition generating the service request is cleared, not until the device is simply polled.

For the plotter, the only condition that causes a service request is "Operator Digitize Output Block Ready." The value of the serial poll response byte can be either 0 (decimal), which indicates that the plotter is not requesting service, or 64, which indicates that it is. The remaining bits in the serial poll response byte are not used. Therefore, the SRQ message is transmitted when the front-panel CALL switch is pressed. Refer to *Digitizing Commands* in Section 7 for more information on the Digitize function. The plotter must not be sent a SERIAL POLL command when in Listen-Only or Talk-Only modes.

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 shown in Figure 5-4 for a 4050 Series Graphic Computing System uses the SRQ function to read digitized points into the graphic system while the main program performs other processing. When 25 points are digitized, all previously digitized points are displayed.



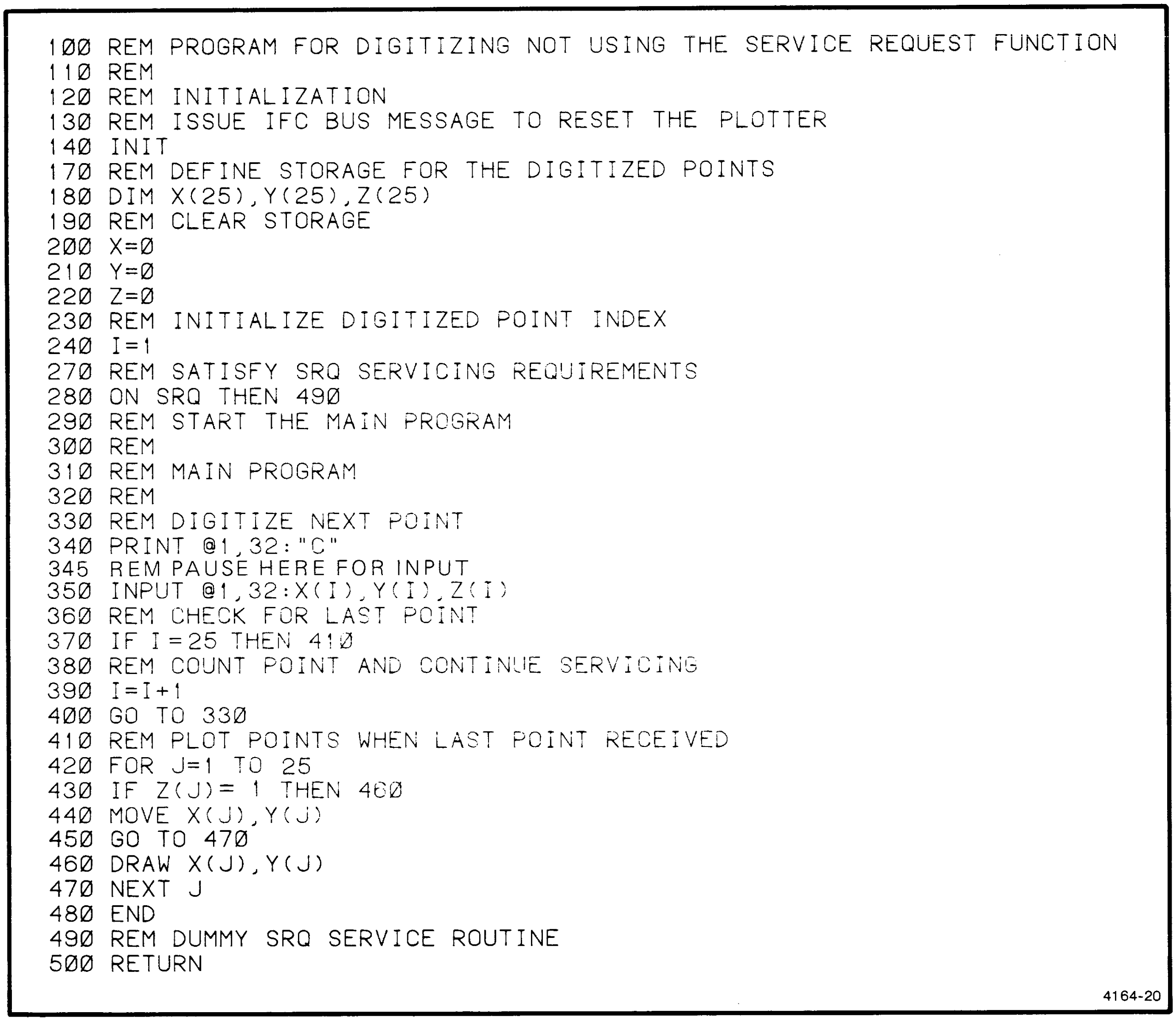
```
1ØØ REM PROGRAM FOR DIGITIZING USING THE SERVICE REQUEST FUNCTION
11Ø REM
12Ø REM INITIALIZATION
13Ø REM ISSUE IFC BUS MESSAGE TO RESET THE PLOTTER
14Ø INIT
17Ø REM DEFINE STORAGE FOR THE DIGITIZED POINTS
18Ø DIM X(25),Y(25),Z(25)
19Ø REM CLEAR STORAGE
2ØØ X=Ø
21Ø Y=Ø
22Ø Z=Ø
23Ø REM INITIALIZE DIGITIZED POINT INDEX
24Ø I=1
27Ø REM ENABLE SRQ SERVICING
28Ø ON SRQ THEN 36Ø
29Ø REM START THE MAIN PROGRAM
3ØØ REM
31Ø REM MAIN PROGRAM
32Ø REM
33Ø REM OTHER PROCESSING COULD BE PERFORMED HERE
34Ø REM DUMMY PROGRAM
35Ø GO TO 35Ø
36Ø REM
37Ø REM SRQ SERVICE ROUTINE
38Ø REM
39Ø REM DO SERIAL POLL TO DETERMINE STATUS
4ØØ POLL A,B;1
41Ø REM SERVICE PLOTTER UNTIL SERVICE REQUEST STATUS IS Ø
42Ø IF B<>Ø THEN 44Ø
43Ø RETURN
44Ø REM DIGITIZE NEXT POINT
45Ø PRINT @1,32:"C"
46Ø INPUT @1,32:X(I),Y(I),Z(I)
47Ø REM CHECK FOR LAST POINT
48Ø IF I =25 THEN 52Ø
49Ø REM COUNT POINT AND CONTINUE SERVICING
5ØØ I=I+1
51Ø GO TO 4ØØ
52Ø REM PLOT POINTS WHEN LAST POINT RECEIVED
53Ø FOR J=1 TO 25
54Ø IF Z(J)= 1 THEN 57Ø
55Ø MOVE X(J),Y(J)
56Ø GO TO 58Ø
57Ø DRAW X(J),Y(J)
58Ø NEXT J
59Ø END
```

4164-19

Figure 5-4. Operator-Digitized Input Using the SRQ Function.

Alternatively, digitized input can be read directly by the main program. An example of this operation appears in the short program shown in Figure 5-5 for a 4050 Series Graphic Computing System. Here, the GPIB halts until coordinates are generated. For controllers which do recognize SRQ messages, this function must be disabled or a dummy service routine added as shown to reset the response to the SRQ message (if this mode of coordinate input is desired).

In either of these cases, if the number of points to be digitized is not known in advance, the program can also input the response from the plotter as an ASCII string and then examine it to determine if the response contains three values or one value. (Three values indicate that the response is a digitized point, while one value indicates a status response.) The program terminates its input when the status response (caused by the operator pressing the CALL switch until the bell sounds) is received.



```
1ØØ REM PROGRAM FOR DIGITIZING NOT USING THE SERVICE REQUEST FUNCTION
11Ø REM
12Ø REM INITIALIZATION
13Ø REM ISSUE IFC BUS MESSAGE TO RESET THE PLOTTER
14Ø INIT
17Ø REM DEFINE STORAGE FOR THE DIGITIZED POINTS
18Ø DIM X(25),Y(25),Z(25)
19Ø REM CLEAR STORAGE
2ØØ X=Ø
21Ø Y=Ø
22Ø Z=Ø
23Ø REM INITIALIZE DIGITIZED POINT INDEX
24Ø I=1
27Ø REM SATISFY SRQ SERVICING REQUIREMENTS
28Ø ON SRQ THEN 49Ø
29Ø REM START THE MAIN PROGRAM
3ØØ REM
31Ø REM MAIN PROGRAM
32Ø REM
33Ø REM DIGITIZE NEXT POINT
34Ø PRINT @1,32:"C"
345 REM PAUSE HERE FOR INPUT
35Ø INPUT @1,32:X(I),Y(I),Z(I)
36Ø REM CHECK FOR LAST POINT
37Ø IF I=25 THEN 41Ø
38Ø REM COUNT POINT AND CONTINUE SERVICING
39Ø I=I+1
4ØØ GO TO 33Ø
41Ø REM PLOT POINTS WHEN LAST POINT RECEIVED
42Ø FOR J=1 TO 25
43Ø IF Z(J)= 1 THEN 46Ø
44Ø MOVE X(J),Y(J)
45Ø GO TO 47Ø
46Ø DRAW X(J),Y(J)
47Ø NEXT J
48Ø END
49Ø REM DUMMY SRQ SERVICE ROUTINE
5ØØ RETURN
```

4164-20

Figure 5-5. Operator-Digitized Input Without the SRQ Function.

NOTE

SRQs are not generated if the CALL switch is held down until the plotter's bell sounds when sending status.

If the CALL DIGITIZE command has already been issued to address the plotter as a talker when the operator-digitized coordinates are generated, then the response is transmitted without the SRQ message being sent. However, if a coordinate is generated before the plotter is addressed as a talker, the SRQ message is sent.

CONTROLLING THE PLOTTER FROM 4050 SERIES GRAPHIC COMPUTING SYSTEMS

The following examples use the 4050 Series Graphic Computing System to illustrate how a GPIB device is programmed to send commands to and receive responses from the plotter. Although the examples are written using the 4050 Series BASIC language, most controllers have similar capabilities or functions. Refer to the controller's programming manual for instructions on how to perform these same functions. For information on the 4050 Series 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 Computing System Reference Manual*

These examples describe three different methods for programming plotter communications. These methods, which are briefly explained in the remainder of this section, cover a broad range of capability, and one method should be appropriate in most situations.

NOTE

In the following examples, the plotter's device address is asssumed to be 1.

Plotter Communication One Byte at a Time

The first method uses a 4050 Series RByte or WByte function to command Move, Print, and Digitize operations by programming the communications "one byte at a time." Although the RByte and WByte 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.

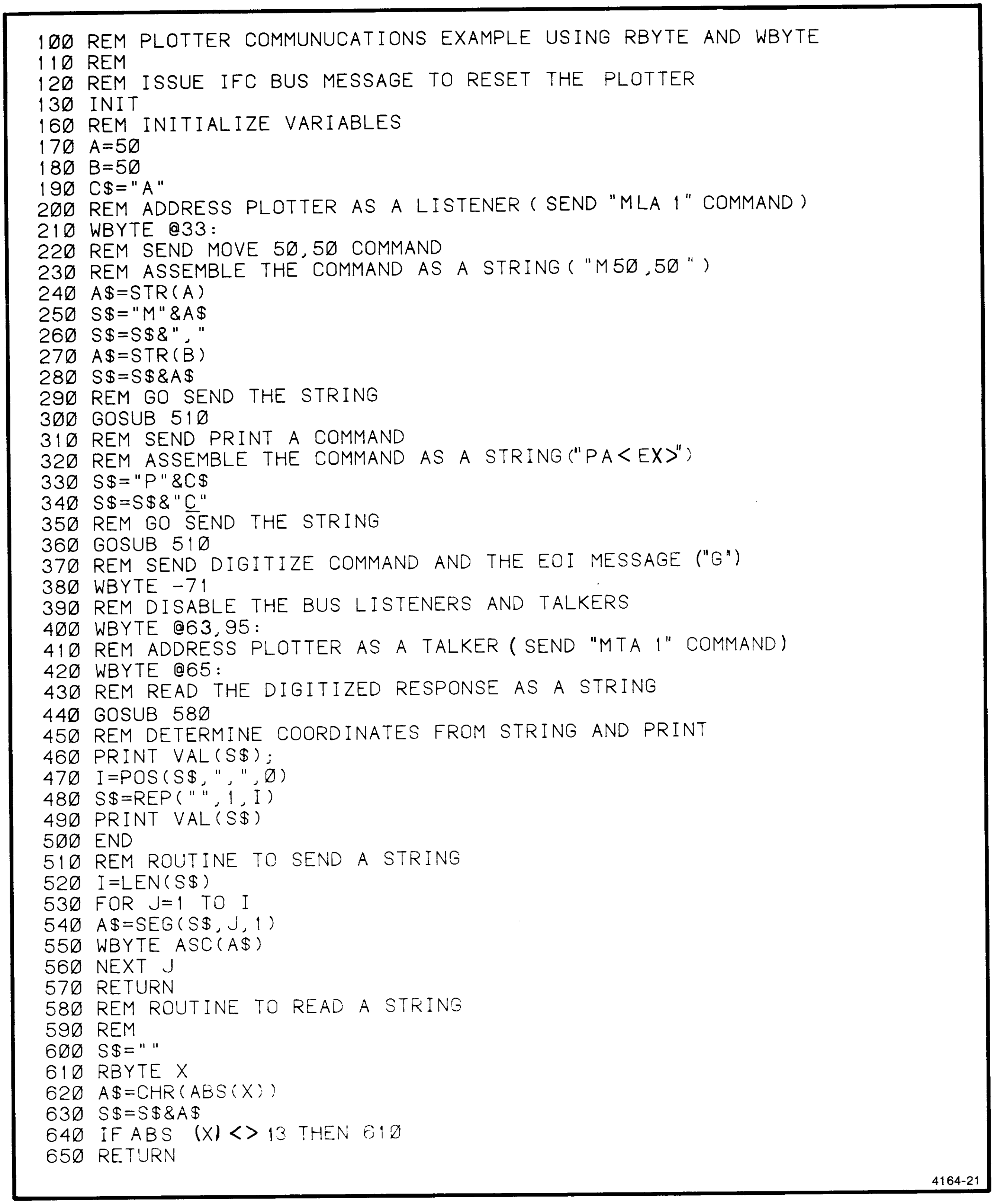
Figure 5-6 illustrates a short program which sends commands in data byte (DAB) form to have the plotter move to 50,50, print an "A", and then digitize the pen position.

The following paragraphs describe the GPIB actions involved. The two-digit numbers in the WByte commands are the ASCII decimal equivalents (ADEs) of DABs to be transmitted over the GPIB. The ADE value for each DAB character is shown previously in Figure 5-2. The @ character causes the GPIB's ATN bus message to be sent, and the : character causes the ATN bus 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 Computing System issues a RByte or WByte command, it automatically assigns itself as a listener or talker respectively 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 PRINT command is terminated with an ᴱx 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 unlisten message, the plotter is enabled as a talker so it can transmit the digitized response. As the number of characters of the response is not predefined, the program reads response bytes until the plotter sends the ᶜʀ character terminating the response. The desired coordinates are then extracted from the string of response bytes and printed.



```
1ØØ REM PLOTTER COMMUNUCATIONS EXAMPLE USING RBYTE AND WBYTE
11Ø REM
12Ø REM ISSUE IFC BUS MESSAGE TO RESET THE PLOTTER
13Ø INIT
16Ø REM INITIALIZE VARIABLES
17Ø A=5Ø
18Ø B=5Ø
19Ø C$="A"
2ØØ REM ADDRESS PLOTTER AS A LISTENER ( SEND "MLA 1" COMMAND )
21Ø WBYTE @33:
22Ø REM SEND MOVE 5Ø,5Ø COMMAND
23Ø REM ASSEMBLE THE COMMAND AS A STRING ( "M5Ø,5Ø " )
24Ø A$=STR(A)
25Ø S$="M"&A$
26Ø S$=S$&","
27Ø A$=STR(B)
28Ø S$=S$&A$
29Ø REM GO SEND THE STRING
3ØØ GOSUB 51Ø
31Ø REM SEND PRINT A COMMAND
32Ø REM ASSEMBLE THE COMMAND AS A STRING("PA<EX>")
33Ø S$="P"&C$
34Ø S$=S$&"C"
35Ø REM GO SEND THE STRING
36Ø GOSUB 51Ø
37Ø REM SEND DIGITIZE COMMAND AND THE EOI MESSAGE ("G")
38Ø WBYTE -71
39Ø REM DISABLE THE BUS LISTENERS AND TALKERS
4ØØ WBYTE @63,95:
41Ø REM ADDRESS PLOTTER AS A TALKER ( SEND "MTA 1" COMMAND)
42Ø WBYTE @65:
43Ø REM READ THE DIGITIZED RESPONSE AS A STRING
44Ø GOSUB 58Ø
45Ø REM DETERMINE COORDINATES FROM STRING AND PRINT
46Ø PRINT VAL(S$);
47Ø I=POS(S$,",",Ø)
48Ø S$=REP(" ",1,I)
49Ø PRINT VAL(S$)
5ØØ END
51Ø REM ROUTINE TO SEND A STRING
52Ø I=LEN(S$)
53Ø FOR J=1 TO I
54Ø A$=SEG(S$,J,1)
55Ø WBYTE ASC(A$)
56Ø NEXT J
57Ø RETURN
58Ø REM ROUTINE TO READ A STRING
59Ø REM
6ØØ S$=""
61Ø RBYTE X
62Ø A$=CHR(ABS(X))
63Ø S$=S$&A$
64Ø IF ABS (X) <> 13 THEN 61Ø
65Ø RETURN
```

4164-21

Figure 5-6. A GPIB Communications Example Using RByte and WByte to Send DAB Commands.

Plotter Communications Using DAB Commands

Higher-level 4050 Series BASIC commands control the plotter more easily than the RByte and WByte functions. The higher-level commands are in DAB form, which may be generated by most GPIB devices.

Figure 5-7 shows the same program as that shown in Figure 5-6, 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 Computing System not to send the MSA command form (described later in this section). The PRINT statement automatically causes device 1 (the plotter) to be addressed as a listener before the data bytes are transmitted. The GPIB is cleared with the untalk and unlisten messages after the data bytes are transmitted. The semicolon (;) (after the PRINT statement) suppresses a CR character, which is otherwise always added to the data bytes transmitted. The input command causes the plotter to be addressed as a talker and the plotter output to be accepted by the 4050 Series controller.

The semicolon (;) used in place of the comma (,) as an argument delimiter suppresses formatting spaces, which are otherwise automatically added between output variables (resulting in less efficient transmission to the plotter).

The 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 5-7:

PRINT @1,32:"M50,50"

or

C$="M50SP50"
PRINT@1,32:C$

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 statement.

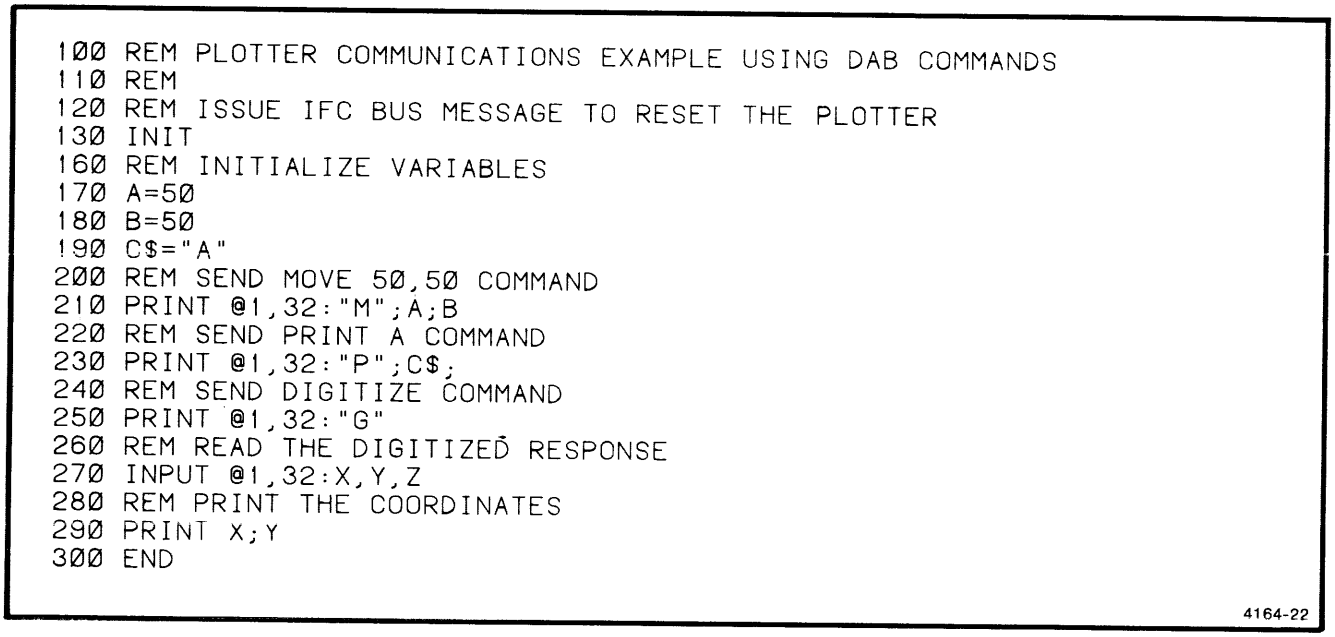
PRINT @1,32:"M";50;50

or

A=50
B=50
PRINT @1,32:"M";A;B

or

A=50
B=50
C$="M"
PRINT @ 1,32:C$;A;B



```
100 REM PLOTTER COMMUNICATIONS EXAMPLE USING DAB COMMANDS
110 REM
120 REM ISSUE IFC BUS MESSAGE TO RESET THE PLOTTER
130 INIT
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
```

4164-22

Figure 5-7. GPIB Communications Example Using PRINT to Send DAB Commands.

In these forms, the command code is expressed as a string constant or string variable. Arguments are numeric constants or numeric variables. The semicolon (;) suppresses formatting spaces that would otherwise be automatically inserted by the 4050 Series Graphic Computing 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

Therefore, 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 therefore 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 may be combined on the same line, as shown in the following examples using the **MOVE 50,50** and the **PRINT A** commands from Figure 5-7:

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 (in the previous example, the EOI, which is automatically generated by the 4050 at the end of a print string, terminates the end of a print string argument). The ; is used to suppress the $^{C}_{R}$ action.

PRINT @1,32:"M50$^{S}_{P}$50PA$^{E}_{X}$G"

For clarity, these commands could be separated by a command terminator as shown next:

PRINT @1,32:"M50$^{S}_{P}$50;PA$^{E}_{X}$;G"

Plotter Communications Using MSA Commands

Figure 5-8 shows the same example using the MSA command forms. This example illustrates how 4050 Series BASIC statements may be used to generate the MSA form of the plotter commands using the PRINT and INPUT keywords. Refer to the command descriptions in Section 7 to determine if a MSA equivalent form is given for the desired plotter command.

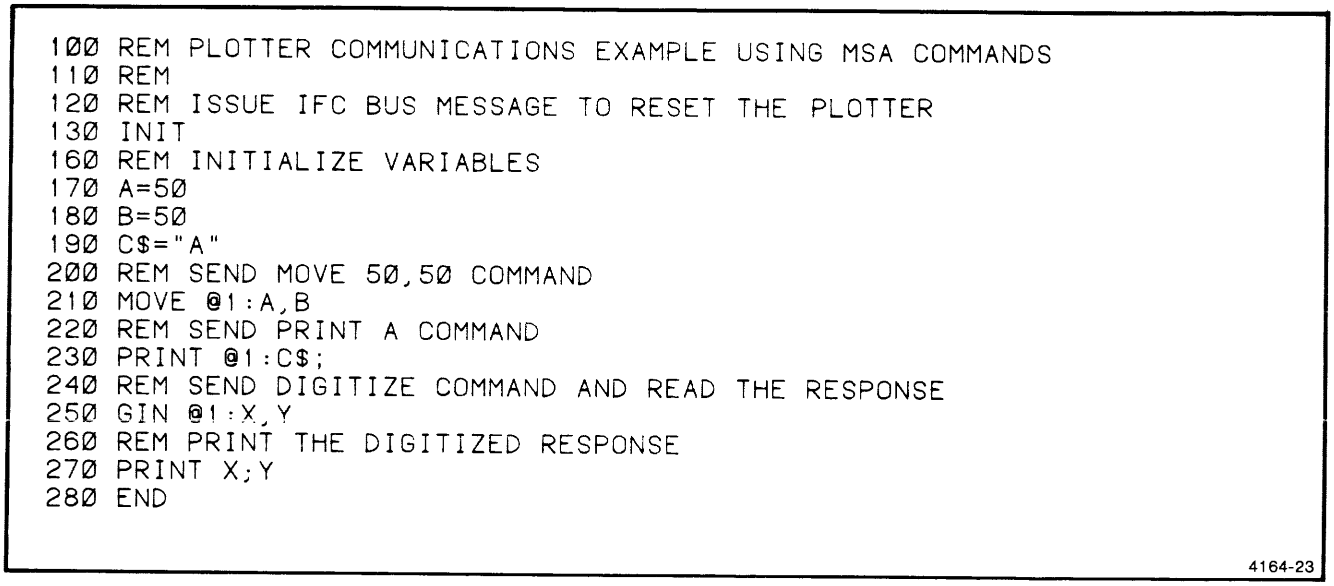
NOTE

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. The output from a 4050 Series Graphic Computing System keyword command may be directed to either the plotter or to the display by simply changing the device address from 1 = plotter to 32 = display.

NOTE

The INPUT statement is not required in the MSA form of the example, because the GIN statement automatically does an input function.

The plotter accepts the MSA commands shown in Table 5-1 for compatibility with 4050 Series Graphic Computing Systems. All MSA commands not listed here are ignored. Table 5-1 also indicates any corresponding 4050 Series BASIC language keywords which are used to generate these commands.



```
1ØØ REM PLOTTER COMMUNICATIONS EXAMPLE USING MSA COMMANDS
11Ø REM
12Ø REM ISSUE IFC BUS MESSAGE TO RESET THE PLOTTER
13Ø INIT
16Ø REM INITIALIZE VARIABLES
17Ø A=5Ø
18Ø B=5Ø
19Ø C$="A"
2ØØ REM SEND MOVE 5Ø,5Ø COMMAND
21Ø MOVE @1:A,B
22Ø REM SEND PRINT A COMMAND
23Ø PRINT @1:C$;
24Ø REM SEND DIGITIZE COMMAND AND READ THE RESPONSE
25Ø GIN @1:X,Y
26Ø REM PRINT THE DIGITIZED RESPONSE
27Ø PRINT X;Y
28Ø END
```

Figure 5-8. GPIB Communications Example for MSA Command Form.

Table 5-1
MSA COMMANDS

MSA	Equivalent 4662 Command	Corresponding 4050 Series Keyword
00	READ STATUS WORD	[a]
07	RESET ALPHA PARAMETERS	[a]
08	PEN SELECT	[a]
12	PRINT	PRINT
13[b]	SIZE	[a]
17	SET ALPHA SCALE	[a]
18	SELECT ALPHA FONT	[a]
19	PRINT	LIST
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 Computing System sends MOVE and DRAW commands to the plotter to draw the axis. No corresponding MSA command is generated.

[c] The INPUT keyword is used instead of the PRINT keyword to generate the plotter command for these secondary addresses.

Other plotter commands listed in Table 5-1 can be generated in the same way as the commands shown in Figure 5-8 (for example, using the PRINT and INPUT keywords). Alternatively, the other listed keywords can be used to generate the commands. For example:

PRINT @1,20:"50,50"

and

DRAW @1:50,50

both express a plotter DRAW command. However, the 4050 Series Graphic Computing System performs additional processing of commands produced with the MOVE, RMOVE, DRAW, and RDRAW keywords (this processing does not occur if the commands are generated using the PRINT keyword). For example, the Window and Viewport established by the 4050 Series Graphic Computing System transforms commands (containing MOVE and DRAW keywords) before they are sent to the plotter. To avoid this transformation in the 4050 Series Graphic Computing System, set the 4050 Series Viewport and Window to the same values. Then, commands generated by 4050 Series BASIC statements that contain the MSA or DAB plotter command forms are not transformed before being sent to the plotter.

If plotter DRAW commands generated by DRAW or RDRAW keywords contain coordinates outside the 4050 Series Graphic Computing System's Window, the coordinates are "scissored". This "scissoring" action causes the graphic data to be clipped even if the 4050 Series Graphic Computing System's Window is larger than the plotter's Page.

Section 6

INTRODUCTION TO GPIB COMMANDS AND RESPONSES

INTRODUCTION

This section describes how to form GPIB commands that control the plotter and how to interpret any responses sent by the plotter. A wide range of command and argument forms are available to permit communication with devices having various output capabilities.

The following pages describe the structure of the actual commands sent over the GPIB to the plotter. The user must refer to the controller's manual to determine how these command forms are transmitted over the GPIB. Some examples show commands generated by a 4050 Series Graphic Computing System and may give some insights on how other devices generate plotter commands.

Appendix H shows an example of a simple plot using GPIB commands.

PLOTTER COMMANDS

Refer to Figure 6-1. The general command form recognized by the GPIB Interface is:

<**Command Code**> <**Arguments**>

This section discusses each portion of this command along with any separators and terminators required to correctly punctuate the commands. Also presented is the reaction of the plotter to incomplete or unrecognized commands.

COMMAND CODE

The command code consists of one or two uppercase or lowercase ASCII alphabetic characters that identify the command. The command code for each plotter operation is defined in Section 7.

NOTE

Only uppercase ASCII alphabetic characters are allowed in plotters containing Firmware Version 3 or lower.

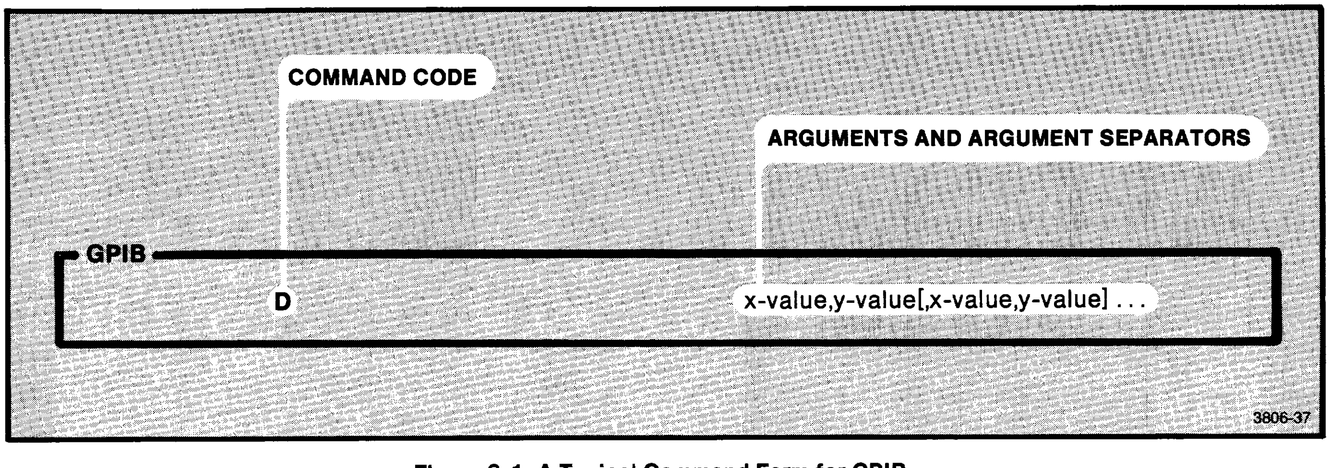

COMMAND CODE
ARGUMENTS AND ARGUMENT SEPARATORS
GPIB
D
x-value,y-value[,x-value,y-value] . . .
3806-37

Figure 6-1. A Typical Command Form for GPIB.

ARGUMENTS

Any arguments defined for a particular command follow the command code. Specific arguments for each command are defined in the description of each command (Section 7). Command arguments can be of two general types: numeric or string. Individual arguments are separated from each other by argument separators, and the end of the argument is marked by a command terminator. Refer to *Argument Separators* and *Command Terminators* described 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)

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 is expressed in an integer or floating point form.
- The E is uppercase (lowercase is accepted, but not recommended), and the exponent is expressed as one or two 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 which is not part of a specified numeric form (for example, a character which is not one of the digits 0 through 9 or the characters + , −, e, E, and .).

These numeric argument forms accommodate 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 description of each command in Section 7.

A leading **?** or = character before a numeric argument is ignored.

String Arguments

Print string arguments begin with the first character after the command code and may contain any ASCII character except ^{E}x. The string is terminated by an ^{E}x character, an EOI bus message, or by receiving a MLA, MTA, MSA, UNL, or UNT command.

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 is the most common argument separator. However, an argument separator may also be expressed in one of these more general forms:

- A sequence of one or more space characters.
- A comma, followed by another sequence of zero or more space characters.

NOTE

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

COMMAND TERMINATORS

A command terminator is a character which 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 types of command terminators: discarded and reprocessed. Discarded command terminators explicitly delimit the command, and after they are recognized, they are discarded without further interpretation. (EOI is a discarded command terminator). Reprocessed command terminators terminate a command but are also then reprocessed as part of a following command. An example is when the command code terminates one command and starts the next command.

The last argument of a command must be properly terminated. For example, the next command code terminates a numeric argument but not a string argument.

NOTE

Commands which have no arguments require no command terminators.

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 command descriptions in Section 7 indicate which commands allow multiple argument groups to be specified.

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.

COMMAND INTERPRETATION

Errors in forming or communicating commands may cause the plotter to receive commands unrecognizable as legal plotter commands. The following outlines the plotter's response to these commands:

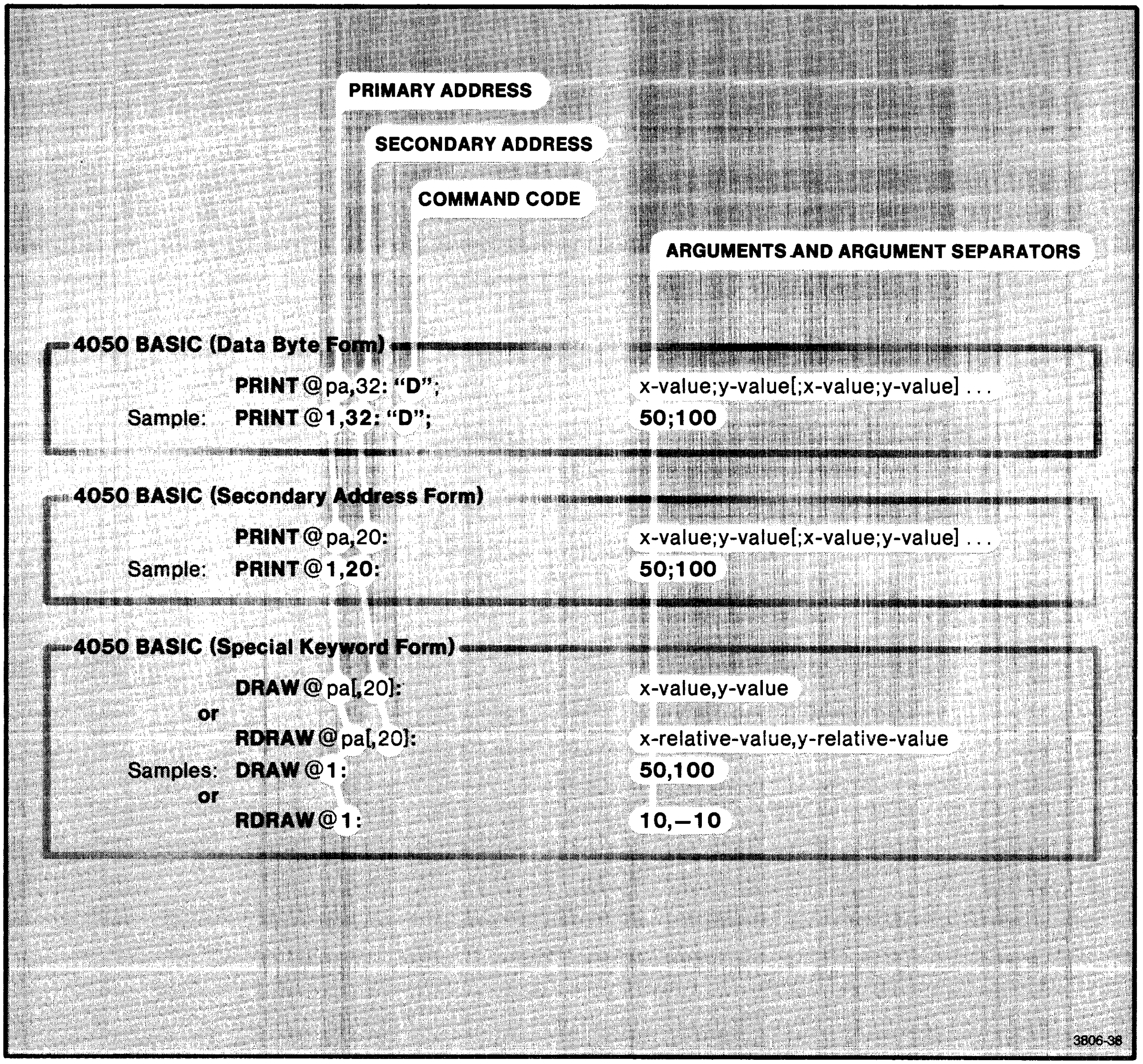
- If the command code is not recognized, it is ignored (characters in the arguments may be interpreted as command codes).
- If there is an error in a command argument, a COMMAND/RESPONSE error is generated. The command may be ignored or may be processed with undefined arguments.

COMMAND FORMAT FOR 4050 SERIES BASIC GPIB APPLICATIONS

In the second "syntax box" for each command description in Section 7, 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.) Refer to Figure 6-2.

Primary Address and Secondary Address

All three GPIB forms require a GPIB Primary Address value which matches the selection made by the four rear-panel switches. 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 numeric expression. For example, a primary address of one may be represented by "1" or by "D" where D= 1.



```
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 6-2. The Three Command Forms for 4050 BASIC Applications.

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 SP before the second argument is suppressed. The plotter requires a separator — for example, either a SP or a comma (,). Programming actions that avoid the problem include:

- *Use a comma (,) instead of a semicolon (;) between the string constant or string variable and the next 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 next argument if the next argument is a 4050 BASIC array variable.

For a print-string argument, a 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 (EX) whenever a print string is followed by another command in the same BASIC statement. Remember that a semicolon (;) at the end of the statement supresses the automatic CR produced at the end of a BASIC print statement.

OUTPUT RESPONSES

Output-generating plotter commands cause an output block to be created. The listening device decodes these output blocks to obtain the desired information.

Each output block consists of one or three numeric values. For most commands, the 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 descriptions that follow.

In forming output blocks, the data values are converted to ASCII decimal numbers in scientific notation, with each value separated by commas and the complete block terminated by a CRLF sequence. The values are then transmitted through the output buffer. Refer to *Plotter-to-Device Transmission* in Section 5 for more information.

COMMAND NOTATIONS

CHARACTERS

ASCII Characters

The 94 printing ASCII characters (numbers, symbols, 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 in Appendix C.

NOTE

> *The ASCII space character is always shown as* SP*, and not as a "blank space" between printed characters.*

Examples of characters: & 1 2 A B a b EC SP DT

The last three ASCII characters are escape, space, and delete.

LITERAL VERSUS VARIABLE ELEMENTS

Boldface

Boldface is used to indicate a literal element, entered exactly as shown.

(The "at" signs in this manual are an exception to this rule; although the @ appears in regular type, enter the @ exactly as shown in the commands.)

Regular Type

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

Examples:

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

In this example, two elements (pa and rotation-angle) are to be replaced, and **PRINT** @ and **,32:"R";** are entered exactly as shown.

PRINT @ pa**,32:"R";**rotation-angle

In this case, all ASCII characters are entered exactly as shown.

COMMAND ELEMENTS

Element Types

The last word(s) of a 4662 plotter variable element name indicates the element type required by the 4662 as shown below:

Last Word(s)	Element Type
selector	selector
print-string	print string
pa	primary address
target-variable	target variable

Other endings imply a "numeric" element.

For example:

- pen-selector is a "selector" element
- print-character-string is a "print string" element
- rotation-angle is "numeric" element

Argument Separators

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

Command Terminations

Command terminators are generally NOT shown in the individual command descriptions. Refer to *Command Terminators* earlier in this section for instructions on how to terminate commands.

NOTATIONS

Brackets [ ]

An element inside brackets is OPTIONAL. Stacked elements within brackets indicate that you may select one or none of the elements.

Example:

[0]
[1]

Select 0, 1, or neither.

Dots ...

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

Example:

$^{S}{}_{P}$...

One or more $^{S}{}_{P}$ characters.

Indented for Continuation

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

Example:

```
INPUT @pa,13:value-1-target-variable,
             value-2-target-variable,
             value-3-target-variable
```

Section 7

GPIB COMMAND DESCRIPTIONS

COMMAND DESCRIPTION OVERVIEW

GPIB commands are divided into major groups according to similar functions. These groups are:

- Device Commands
- Alpha Commands
- Graphics Commands
- Digitizing Commands
- Multiple Pen (Option 31) Commands

The discussion of each command group is divided into two parts: general concepts about the group and specific command descriptions for each command in the group. Each command description follows a general format and contains all or some of the following:

- Purpose
- Syntax Box
- Parameters (This includes restrictions, meanings assigned to selector arguments, etc.)
- Outputs
- Comments
- References
- Examples

The syntax boxes show a recommended command form that can be used under most conditions. The syntax boxes usually include specific examples of the commands. *Comments* are sometimes included in the command description to clarify specific entries in the syntax boxes.

Appendix H shows a simple plot made using several of the commands described in this section.

DEVICE COMMANDS

CONCEPTS

Some of the Device commands generate output messages. Therefore, it is important to understand the information on output messages in Section 6, *Introduction to GPIB Commands and Responses*.

DEVICE COMMAND DESCRIPTIONS

There are three Device commands:

- DEVICE RESET
- READ STATUS
- SIZE

Each command is described separately in the pages that follow.

DEVICE RESET

Purpose: This command resets graphic, alpha, and interface parameters to power-up (default) values.

GPIB

IFC (GPIB bus message)

4050 BASIC (Special Keyword Form)

INITIALIZE

Comments

These actions occur as a result of this command:

- The GPIB Interface hardware is reset.
- Alpha parameters are reset to default values. This is equivalent to an ALPHA RESET command. See the description of ALPHA RESET under *Alpha Commands* later in this section for a complete list of actions.
- The PROMPT light is turned off (equivalent to receiving a PROMPT LIGHT OFF command).
- The output buffer is cleared.
- The PEN SPEED is reset to FULL SPEED (equivalent to a PEN SPEED command with an argument of 570 mm/second) in Option 31 equipped plotters.
- The ERROR light is turned off (if on) and the error bits in the device status register are cleared.

References

Refer to *Interface Clear* in Section 5, *GPIB Interface Concepts*.

READ STATUS

Purpose: This command causes the plotter to generate an output message containing plotter status information.

GPIB

```
          V status-register-selector
```

4050 BASIC (Data Byte Form)

```
          PRINT@pa,32:"V";status-register-selector
          INPUT@pa,32:value-1-target-variable
Example:  PRINT@1,32:"V";0
          INPUT @I,32:V1
```

4050 BASIC (Secondary Address Form)

```
          PRINT@pa,0:status-register-selector
          INPUT@pa,32:value-1-target-variable
Example:  PRINT@1,0:0
          INPUT@1,32:V1
```

Parameters

status-register-selector

Status register selector, 0 — 3 (Status Information — Value 1, see *Comments* below)

Outputs

See *Comments*.

Comments

The legal status-register-selector range is 0 — 3. Plotter responses are defined in Table 7-1.

Table 7-1

STATUS INFORMATION

Status-Register Selector	Status Information
0	Device status
1	Interface status
2	Current selected pen (Option 31 only, 0 otherwise)
3	Available buffer space (Firmware Level 4 and up only; 0 otherwise)

Device Status

This is a 16-bit integer value in which each binary digit indicates if a particular condition is true or false. Unused bit positions are currently zero, but are reserved for future use and must not always be assumed to be zero (see Table 7-2). The "1" binary value for a specified position indicates that the state associated with that position is true. A zero indicates the state is not true. Table 7-2 describes each bit position.

Table 7-2

DEVICE STATUS BITS

Bit[a]	Description
15	This bit is not currently used (set to zero).
14	This bit is not currently used (set to zero).
13	Pen Select Error. This bit is set to "1" by a PEN SELECT command (either front-panel or host command) if the requested pen position is empty or if there is no empty pen position to store the pen currently in the plotter's pen carriage. This bit is reset by a READ STATUS 0 or DEVICE RESET command, or by pressing the front-panel LOAD switch.
12	Plotting Speed. This bit is set to "1" if Slow Plotting Speed is selected by either the front-panel (Option 31 equipped plotters) or rear-panel switches (plotters without Option 31). The bit is 0 otherwise.
11	CR Action. This bit is set to "1" if the rear-panel CR action switch is set to CR implies CR + LF. The bit is 0 otherwise.
10	Pen Status. This bit is set to "1" if the plotter's pen is down on the paper or "0" if the pen is up.
9	Load Mode. This bit is set to "1" if the plotter is in Load mode (the front-panel LOAD switch is pressed down and locked). The bit is "0" if the LOAD switch is up (released).
8	Page X Mirrored. This bit is set to "1" 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 "0" otherwise.
7	Page Y Mirrored. This bit is set to "1" if the Page lower-left y-coordinate has been defined above the Page upper-right y-coordinate. Bit-7 is set to "0" otherwise.
3 – 6	Bit-6 through Bit-3 indicate when the current pen position is outside the current page boundaries as follows: Bit-6 X Right Bit-5 X Left Bit-4 Y Above Bit-3 Y Below Left, right, above, and below are defined with respect to the horizontally oriented platen surface. These directions are not altered by a mirrored page.
2	Communications Error. A communications error, such as a parity or framing error, sets this bit to "1". This bit is cleared by pressing the front-panel LOAD switch, by sending a READ STATUS WORD 0 command, or by sending a DEVICE RESET command.
1	Command/Response Error. Command/response errors occur when an invalid command, argument, or format is received. This bit is cleared by pressing the front-panel LOAD switch, by sending a READ STATUS WORD 0 command, or by sending a DEVICE RESET command.
0	Internal Error. A internal error condition sets this bit to "1".

[a] Bit-0 is the least-significant bit.

Interface Status

This is a 16-bit integer value in which Bit-11 through Bit-15 are binary digits indicating true or false. A "1" binary value for a position indicates that the state associated with that position is true. A "0" indicates the state is not true. Bit-0 through Bit-10 encode the amount of available memory in the input buffer. Table 7-3 shows the meanings of the bits.

Table 7-3

INTERFACE STATUS

Bit[a]	Description If GPIB Interface Selected
15	Bit is not currently used (set to 0)
14	"1" if rear-panel CR action switch is set to CR implies CR + LF; "0" otherwise
13	Bit is not used (set to 0)
12	"1" if rear-panel switch is set to inhibit MSA commands; "0" otherwise
11	"1" if rear-panel switch is set to inhibit DAB commands; "0" otherwise
0 – 10	Indicates amount of available buffer space

[a] Bit-0 is the least-significant bit.

NOTE

In plotters with Firmware Version Level 4 and higher, if more than 2047 bytes of space are available, this field will be forced to 2047. In these firmware versions, the actual number of free bytes is indicated in Status Word 3.

Currently Selected Pen

If Option 31 is installed, this integer value indicates the pen number currently in the pen carriage (1 to 8). If no pen is currently selected, a value of 0 is returned.

Zero is always returned if Option 31 is not installed on the plotter.

Available Buffer Space

If the plotter's firmware is Version 4 or higher, this integer value indicates the number of available bytes in the input buffer. In plotters with Version 3 Firmware and below, this value is always zero.

References

Refer to Section 6, *Introduction to GPIB Commands and Responses*, for further information on decoding output responses from the plotter.

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.

GPIB

```
I
```

4050 BASIC (Data Byte Form)

```
          PRINT@pa,32:"I"
          INPUT@pa,32:x-size-target-variable,
                      y-size-target-variable,
                      identification-word-target-variable
Example:  PRINT@1,32:"I"
          INPUT@1,32:X,Y,I
```

4050 BASIC (Secondary Address Form)

```
          INPUT@pa,13:x-size-target-variable,
                      y-size-target-variable,
                      identification-word-target-variable
Example:  INPUT@1,13:X,Y,I
```

Outputs

value 1

- 150 (represents X-size in GDUs).

value 2

- 100 (represents Y-size in GDUs).

value 3

- Identification word (see Table 7-4 under *Comments*).

Comments

The identification word is encoded as a two's-complement number and, therefore, if Bit-15 (Option 31) is set to 1, the returned value is negative. To decode the various bit positions, if the decimal number returned is negative, add 65536 to the number returned to convert it into the range of positive 1 to 65,535.

Table 7-4

IDENTIFICATION WORD

Bit[a]	Information
0 – 3	Firmware release number.
4 – 5	See Table 7-5.
6	In plotters with Firmware Version Levels 4 or higher, this bit is set to "1" if Option 20 (RAM expansion) is installed. This bit is zero in earlier versions.
7 – 13	Currently set to "0" (reserved for future use).
14	Set to "0" for versions earlier than Version 4. In later versions, this bit is set to "1" if the new (4663 compatible) character font is selected by the OLD-NEW strap on the System Memory circuit board, or "0" if the old 4662 character font is selected.
15	Set to "1" if Option 31 is installed; "0" otherwise.

[a] Bit-0 is the least-significant bit.

References

See *Introduction to GPIB Commands and Responses*, Section 6, for information on decoding output responses from the plotter.

Table 7-5

IDENTIFICATION WORD BITS 4 AND 5

Bit 4	Bit 5	Maximum Buffer Space Available in Bytes
0	0	Less than or equal to 600 bytes
0	1	Less than or equal to 1100 bytes
1	0	Less than or equal to 1600 bytes
1	1	Greater than 1600 bytes

ALPHA COMMANDS

INTRODUCTION

The plotter has an internal "alpha" generator that enables the user to print alphanumeric characters and symbols that are defined in nine resident fonts.

The alpha generator prints the 95 ASCII characters $^{S}{}_{P}$ through ~ and responds to the control characters $^{B}{}_{L}$ through $^{C}{}_{R}$.

CONCEPTS

The following 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

PRINT Command Fundamentals

For each $^{S}{}_{P}$ through ~ character in 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 DIMENSION command).

The alpha generator prints characters from the active font according to alpha setup commands for alpha dimension and rotation. The alpha generator also processes control characters $^{B}{}_{L}$ through $^{C}{}_{R}$.

The Automatic MOVE-TO-HOME Function

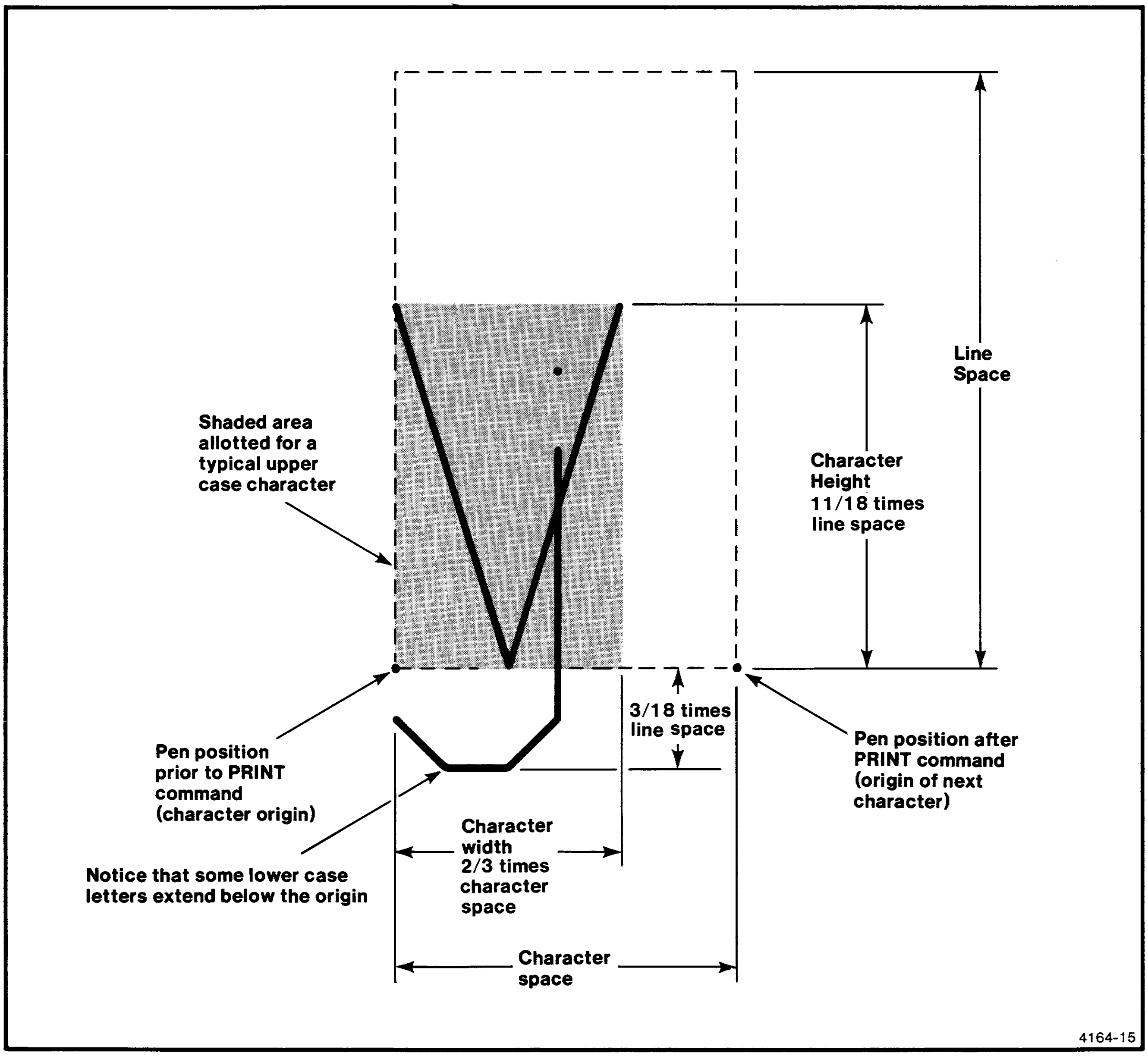
The plotter expects the first PRINT command after power-up to be preceded by a MOVE or DRAW command or a joystick action. If this expectation is not met, the plotter precedes the PRINT command with a MOVE-TO-HOME command (*home* is defined in the next paragraph). The automatic MOVE-TO-HOME function also occurs on the first PRINT command after pressing a front-panel LOAD, LOCATE LOWER-LEFT, or LOCATE UPPER-RIGHT switch.

The Home Position

The *home* position is a point located one character height below the upper-left corner of the page. (If the alpha rotation angle is non-zero, the home position is rotated by this angle about the upper-left corner of the page.)

Setting the Size and Spacing of Printed Characters

The SET ALPHA DIMENSION command is used to specify the character space and line space dimensions. These dimensions are directly related to character width and character height, as illustrated for the PRINT command in Figure 7-1. Default dimensions are described under the SET ALPHA DIMENSION command.



Line Space
Character Height 11/18 times line space
Shaded area allotted for a typical upper case character
3/18 times line space
Pen position prior to PRINT command (character origin)
Pen position after PRINT command (origin of next character)
Notice that some lower case letters extend below the origin
Character width 2/3 times character space
Character space
4164-15

Figure 7-1. Alpha Size and Spacing for the PRINT Command.

Resident Alpha Fonts

The alpha generator has resident character definitions for ASCII characters ˢᴾ through ~ in nine fonts.

Figure 7-2A shows the entire Font 0 (default) character set. All other fonts are identical to Font 0 except for the character substitutions shown in Figure 7-2B.

NOTE

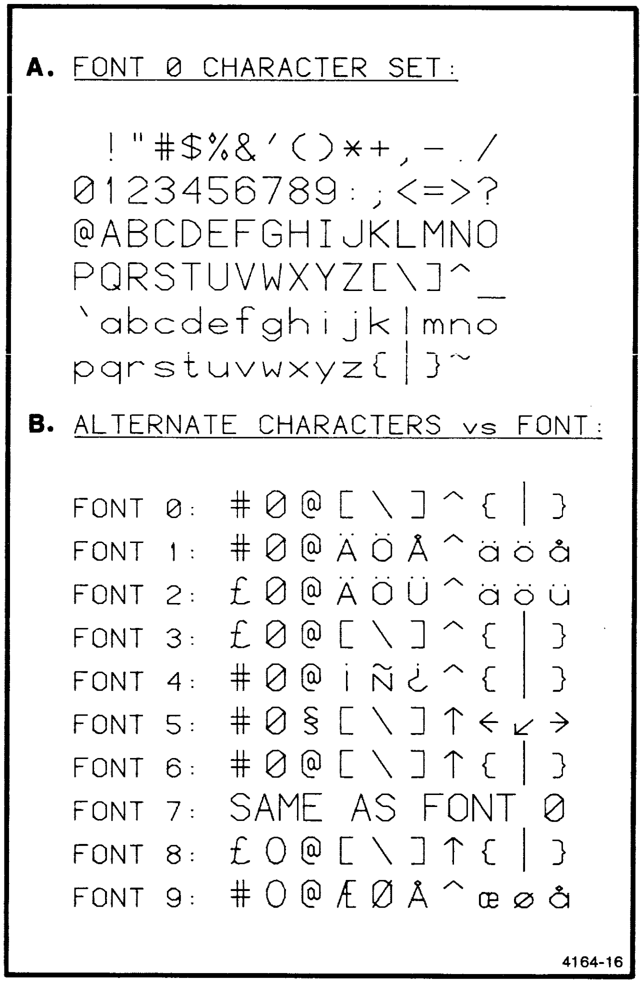
> *Plotters with Firmware Level 3 or less have slightly different character substitutions in Fonts 1 through 6, as shown in Figure 7-3.*
>
> *An OLD-NEW strap is located on the System Memory circuit board in newer plotters to obtain Fonts 1 through 6, which are used in early model plotters. Figure 7-3 shows the old fonts.*

ALPHA COMMAND DESCRIPTIONS

There are six Alpha commands:

- SET ALPHA DIMENSION
- SET ALPHA ROTATION
- SELECT ALPHA FONT
- PRINT
- RESET ALPHA PARAMETERS
- MOVE-TO-HOME

The following pages describe each command separately.



A. FONT 0 CHARACTER SET:

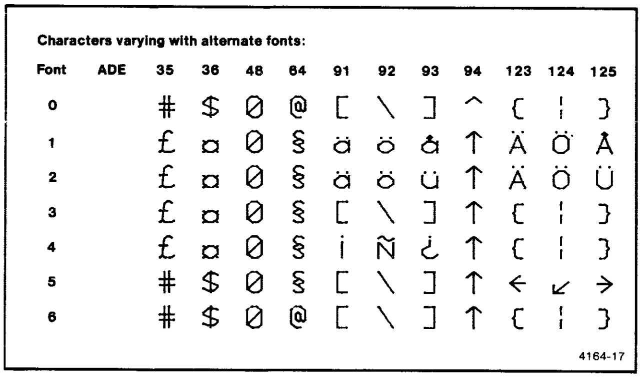
! "#$%&'()*+,-./
0123456789:;<=>?
@ABCDEFGHIJKLMNO
PQRSTUVWXYZ[\]^_
`abcdefghijklmno
pqrstuvwxyz{|}~

B. ALTERNATE CHARACTERS vs FONT:

FONT 0: # 0 @ [\ ] ^ { | }
FONT 1: # 0 @ Ä Ö Å ^ ä ö å
FONT 2: £ 0 @ Ä Ö Ü ^ ä ö ü
FONT 3: £ 0 @ [\ ] ^ { | }
FONT 4: # 0 @ ¡ Ñ ¿ ^ { | }
FONT 5: # 0 § [\ ] ↑ ← ↙ →
FONT 6: # 0 @ [\ ] ↑ { | }
FONT 7: SAME AS FONT 0
FONT 8: £ O @ [\ ] ↑ { | }
FONT 9: # O @ Æ Ø Å ^ æ ø å

4164-16

Figure 7-2. Resident Alpha Fonts.



Characters varying with alternate fonts:

Font	ADE	35	36	48	64	91	92	93	94	123	124	125
0		#	$	0	@	[	\	]	^	{	¦	}
1		£	¤	0	§	ä	ö	å	↑	Ä	Ö	Å
2		£	¤	0	§	ä	ö	ü	↑	Ä	Ö	Ü
3		£	¤	0	§	[	\	]	↑	{	¦	}
4		£	¤	0	§	¡	Ñ	¿	↑	{	¦	}
5		#	$	0	§	[	\	]	↑	←	↙	→
6		#	$	0	@	[	\	]	↑	{	¦	}

4164-17

Figure 7-3. "Old" Style Resident Alpha Fonts.

SET ALPHA DIMENSION

Purpose: This command specifies the character space and line space.

GPIB

```
S character-space,line-space
```

4050 BASIC (Data Byte Form)

```
          PRINT@pa,32:"S";character-space;line-space
Example:  PRINT@1,32:"S";3.584;8.448
```

4050 BASIC (Secondary Address Form)

```
          PRINT@pa,17:character-space;line-space
Example:  PRINT@1,17:3.584;8.448
```

Parameters

character-space
X-axis distance from the beginning of a typical uppercase letter to the beginning of the next, in GDUs.

line-space
Y-axis distance from the baseline for one line of printed characters to the next baseline above it, in GDUs.

Comments

Default values are shown in Table 7-6. The width and height of a typical uppercase letter is 2/3 and 11/18 times the character space and line space values respectively (see Figure 7-1).

Examples

The examples in the syntax boxes cause all subsequent printing and the associated spacing to be twice as wide and three times as high as the default character dimensions.

Table 7-6

DEFAULT ALPHA DIMENSIONS (GPIB INTERFACE)

Character Width	Character Height	Character Space	Line Space
1.195 GDUs	1.721 GDUs	1.792 GDUs	2.816 GDUs

SET ALPHA ROTATION

Purpose: This command sets (1) the alpha rotation angle (specified in degrees) and (2) the left margin (as a line through the current pen carriage position perpendicular to the printing direction).

GPIB

```
                R rotation-angle
```

4050 BASIC (Data Byte Form)

```
                PRINT@pa,32:"R";rotation-angle
      Example:  PRINT@1,32:"R";30
```

4050 BASIC (Secondary Address Form)

```
                PRINT@pa,25:rotation-angle
      Example:  PRINT@1,25:30
```

Parameters

rotation-angle
The angle of rotation, counterclockwise, in degrees.

Comments

The default rotation angle is 0 degrees, and the default left margin is the left side of the page. (See Figure 7-4.) The left margin is used for PRINT CR.

Examples

The examples 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 7-4 shows the effect of a 30° alpha rotation on subsequent PRINT commands.

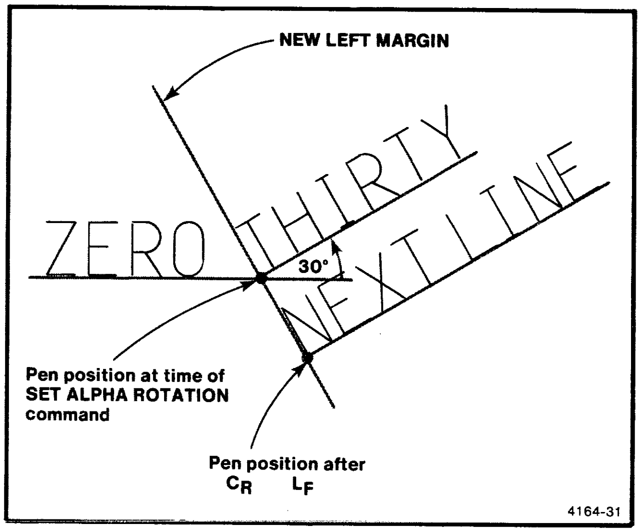


Figure 7-4. Alpha Rotation of 0° and 30°.

SELECT ALPHA FONT

Purpose: This command selects the font used by subsequent PRINT commands.

GPIB

```
F font-selector
```

4050 BASIC (Data Byte Form)

```
          PRINT@pa,32:“F”;font-selector
Example:  PRINT@1,32:“F”;1
```

4050 BASIC (Secondary Address Form)

```
          PRINT@pa,18:font-selector
Example:  PRINT@I,18:1
```

Parameters

font-selector
font selector, 0 — 15

References

Refer to *Resident Alpha Fonts*, under *Concepts* for Alpha commands.

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 of the next character.

GPIB

```
          P print-string
```

4050 BASIC (Data Byte Form)

```
          PRINT@pa,32:"Pprint-string";
Example:  PRINT@1,32:"PABCDefg";
```

4050 BASIC (Secondary Address Form)

```
          PRINT@pa,12:"print-string";
Example:  PRINT@I,12:"ABCDefg";
```

4050 BASIC (Special Keyword Form)

```
          PRINT@pa:"print-string";
Example:  PRINT@1:"ABCDefg";
```

Parameters

print-string

Represents one or more ASCII characters ${}^{B}{}_{S}$ through ${}^{C}{}_{R}$ and ${}^{S}{}_{P}$ through ~. Other characters are ignored, except ${}^{E}{}_{X}$ which is a terminator.

Comments

The alpha generator prints each character ${}^{S}{}_{P}$ through ~ according to current ALPHA DIMENSION, ALPHA ROTATION, and ALPHA FONT command parameters. Control characters are processed as listed in Table 7-7 (other control characters are ignored).

4050 BASIC PRINT statements generate a ${}^{C}{}_{R}$ as the last character transmitted unless the statement ends with a semicolon (;).

Special considerations are necessary to ensure that print strings are properly terminated (see *Print Strings* in Section 5).

Table 7-7

CONTROL CHARACTER RESPONSES

Character	Action
${}^{B}{}_{L}$	Rings the bell.
${}^{B}{}_{S}$	Moves the pen carriage left one character space.
${}^{H}{}_{T}$	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.
${}^{F}{}_{F}$	Moves the pen carriage to the home position (the height of a typical uppercase character from the upper-left corner of the Page, perpendicular to the current printing direction).
${}^{C}{}_{R}$	Moves the pen carriage to the left margin (and generates a ${}^{L}{}_{F}$ if CR GENERATES LF is selected by the four rear-panel switches).
${}^{E}{}_{X}$	Print string terminator.

Examples

The examples in the syntax boxes show how to print the first seven letters in the alphabet: ABCDefg, including lowercase.

RESET ALPHA PARAMETERS

Purpose: This command resets alpha parameters to their default (power-up) values.

GPIB

```
            A
```

4050 BASIC (Data Byte Form)

```
            PRINT@pa,32:"A"
Example:    PRINT@1,32:"A"
```

4050 BASIC (Secondary Address Form)

```
            PRINT@pa,7:
Example:    PRINT@1,7:
```

Comments

The RESET ALPHA PARAMETERS command establishes the default parameters (shown in Table 7-6) for the alpha setup commands. The RESET ALPHA PARAMETERS command also:

- Sets alpha rotation to zero degrees.
- Selects alpha font 0.
- Resets alpha dimensions as shown in Table 7-6.
- Sets the left margin to the left edge of the page.

MOVE-TO-HOME

Purpose: This command moves the pen carriage to the home position.

GPIB

```
            H
```

4050 BASIC (Data Byte Form)

```
            PRINT@pa,32:"H"
  Example:  PRINT@1,32:"H"
```

4050 BASIC (Secondary Address Form)

```
            PRINT@pa,23:
  Example:  PRINT@I,23:
```

4050 BASIC (Special Keyword Form)

```
            HOME@pa:
         or
            PAGE@pa:
 Examples:  HOME@1:
         or
            PAGE@1:
```

Comments

On the plotter, the PAGE and HOME keywords cause the same action.

Examples

The examples in the syntax boxes show how to move the pen carriage to the home position. The home position is a point located one character height below the upper-left corner of the page. (If the alpha rotation angle is non-zero, the home position is rotated by this angle about the upper-left corner of the page.)

GRAPHICS COMMANDS

INTRODUCTION

The following pages include information about standard Graphics commands, including:

- Concepts
- MOVE
- DRAW

CONCEPTS

The MOVE and DRAW commands are sent to the plotter in either of two forms:

- data byte form (DAB)
- secondary address form (MSA)

In addition, another command form is available when using the TEKTRONIX 4050 Series Graphic Computing System. This is the special keyword form shown in the syntax boxes for some commands. This special keyword command form (e.g., MOVE, DRAW, RMOVE, and RDRAW) allows data to be plotted on the 4050 Series screen or on the plotter, using the 4050 Series window and clipping features. (The other two command forms, DAB and MSA, do not use the 4050 Series window and clipping features.)

Move and draw coordinates are expressed in GDUs.

While using the Graphics commands, you may want to refer to Appendix E, which provides additional information on area fill patterns.

GRAPHICS COMMAND DESCRIPTIONS

There are two Graphics commands:

- MOVE
- DRAW

The following pages describe each separately.

MOVE

Purpose: This command causes the pen carriage to move (with the pen up) to the specified position.

GPIB

```
          M x-value,y-value
```

4050 BASIC (Data Byte Form)

```
          PRINT@pa,32:"M";x-value;y-value
Example:  PRINT@1,32:"M";50;100
```

4050 BASIC (Secondary Address Form)

```
          PRINT@pa,21:x-value;y-value
Example:  PRINT@1,21:50;100
```

4050 BASIC (Special Keyword Form)

```
           MOVE@pa:x-value,y-value
      or
           RMOVE@pa:x-relative-value,y-relative-value
Examples:  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 pen moves to the point specified. If either coordinate (or both coordinates) is outside the page boundary, the pen moves to the point on the page boundary that is nearest to the specified point.

Coordinates are specified in GDUs.

The 4050 BASIC keyword forms (MOVE and RMOVE) require special attention when used to produce plotter move commands:

- The 4050 Series Graphic Computing System automatically converts each RMOVE statement to a MOVE statement by adding the current position information, and then sends a MOVE command to the plotter.
- If the 4050 BASIC WINDOW and VIEWPORT are numerically equal in range, MOVE and RMOVE data is sent to the plotter without being scaled. (This is the default 4050 situation.) If the 4050 BASIC WINDOW and VIEWPORT are not identical, the 4050 automatically scales MOVE data before sending it to the plotter.

DRAW

Purpose: The DRAW command causes the plotter to draw a straight line from the current position to the specified position.

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]...
Example:  PRINT@1,32:"D";50;100
```

4050 BASIC (Secondary Address Form)

```
          PRINT@pa,20:x-value;y-value[;x-value;y-value]...
Example:  PRINT@1,20:50;100
```

4050 BASIC (Special Keyword Form)

```
          DRAW@pa:x-value,y-value
       or
          RDRAW@pa:x-relative-value,y-relative-value
Examples: 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 pen draws a line from the current pen position to the point specified. If either the new or previously specified point is outside the page boundary, the DRAW command is treated as a MOVE command. If either coordinate (or both coordinates) is outside the page boundary, the pen moves to the point on the page boundary that is nearest to the specified point.

Coordinates are specified in GDUs.

The 4050 BASIC keyword forms (DRAW and RDRAW) require special attention when used to produce plotter DRAW commands:

- The 4050 Series Graphic Computing System automatically converts each RDRAW statement to a DRAW statement by adding the current position information, and then sends a DRAW command to the plotter.
- The 4050 Series VIEWPORT controls the range of X and Y data sent to the plotter in response to a 4050 BASIC DRAW or RDRAW statement. The 4050 VIEWPORT should be set to 0,150,0,100 to allow full access to the plotter page. (The default 4050 VIEWPORT of 0,130,0,100 does not allow access to the right-most 13% of the plotter page.) The 4050 automatically clips any portion of a DRAW or RDRAW line that is outside the 4050 viewport.

NOTE

The 4050 clipping control applies to plotter commands produced with the special keyword command form (DRAW, RDRAW, MOVE, and RMOVE), but does not apply to plotter commands produced with a DAB or MSA command form. To obtain 4050 clipping control, use the special keyword form for both MOVE and DRAW.

- If the 4050 BASIC WINDOW and VIEWPORT are numerically equal in range, DRAW and RDRAW data are sent to the plotter without being scaled. (This is the default 4050 situation.) If the 4050 BASIC WINDOW and VIEWPORT are different, the 4050 automatically scales DRAW information before sending it to the plotter. For example, for user data expressed in inches, use a 4050 WINDOW of 0,15,0,10 and a VIEWPORT of 0,150,0,100.

DIGITIZING COMMANDS

INTRODUCTION

The plotter's Digitizing commands allow the host computer to request and obtain coordinates of: (1) the current pen position or, (2) points digitized by the operator using the front-panel CALL switch.

Current pen position information is obtained using the DIGITIZE command. Operator-designated points are obtained using the "call digitize" procedures described later. Remember that the CALL switch is permanently enabled and, therefore, the operator must be instructed when to use the CALL switch.

CONCEPTS

Two operator- digitizing procedures are discussed in the following text. The procedures are:

- Call Digitizing via GPIB Interface (With SRQ Messages)
- Call Digitizing via GPIB Interface (without SRQ Messages)

Call Digitizing via GPIB Interface (With SRQ Messages)

The call digitizing operations discussed here involve GPIB "SRQ messages" used to inform the GPIB Controller that the plotter has data available.

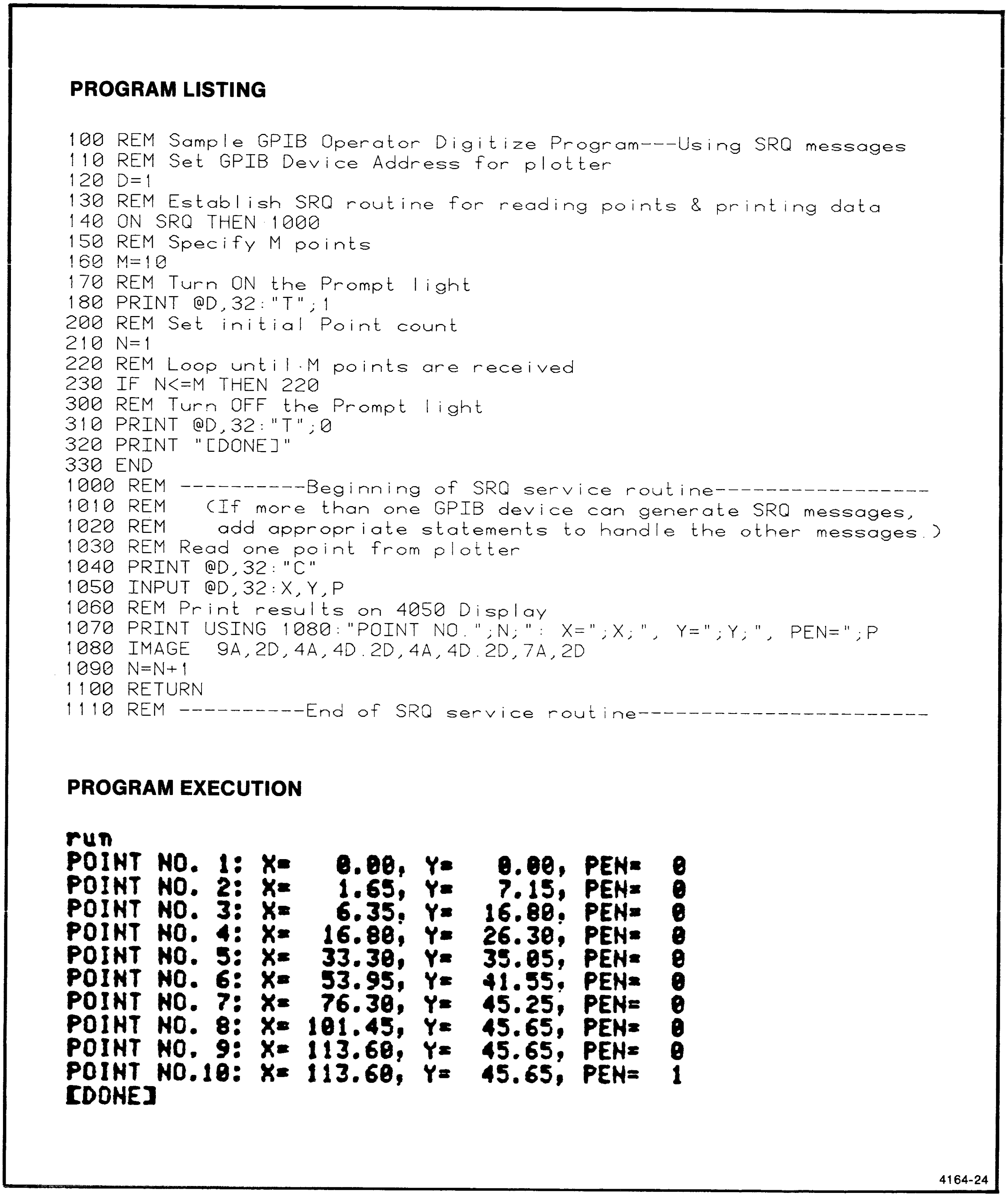
A typical call digitizing process with SRQ messages has these steps:

1. The PROMPT LIGHT ON command is used to turn on the PROMPT light, telling the plotter operator that digitizing is permitted.
2 For each point, the operator uses the joystick to position the digitizing reticle (used in place of a pen in the pen carriage) over the desired point and then presses the front-panel CALL switch. This instructs the plotter to store the point.
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 value indicating whether the pen is up or down.

   If this is NOT the last point to be digitized, the process repeats, starting back at Step 2.
6. The PROMPT LIGHT OFF command turns off the PROMPT light, telling the operator to stop digitizing.

This completes the call digitizing process.

The sample 4050 Series program in Figure 7-5 uses this procedure to store points and to display the results on the 4050 Series display screen.



PROGRAM LISTING

```
100 REM Sample GPIB Operator Digitize Program---Using SRQ messages
110 REM Set GPIB Device Address for plotter
120 D=1
130 REM Establish SRQ routine for reading points & printing data
140 ON SRQ THEN 1000
150 REM Specify M points
160 M=10
170 REM Turn ON the Prompt light
180 PRINT @D,32:"T";1
200 REM Set initial Point count
210 N=1
220 REM Loop until M points are received
230 IF N<=M THEN 220
300 REM Turn OFF the Prompt light
310 PRINT @D,32:"T";0
320 PRINT "[DONE]"
330 END
1000 REM ----------Beginning of SRQ service routine-----------------
1010 REM   (If more than one GPIB device can generate SRQ messages,
1020 REM    add appropriate statements to handle the other messages.)
1030 REM Read one point from plotter
1040 PRINT @D,32:"C"
1050 INPUT @D,32:X,Y,P
1060 REM Print results on 4050 Display
1070 PRINT USING 1080:"POINT NO.";N;": X=";X;", Y=";Y;", PEN=";P
1080 IMAGE  9A,2D,4A,4D.2D,4A,4D.2D,7A,2D
1090 N=N+1
1100 RETURN
1110 REM ----------End of SRQ service routine----------------------
```

PROGRAM EXECUTION

```
run
POINT NO. 1: X=   0.00, Y=   0.00, PEN=  0
POINT NO. 2: X=   1.65, Y=   7.15, PEN=  0
POINT NO. 3: X=   6.35, Y=  16.80, PEN=  0
POINT NO. 4: X=  16.80, Y=  26.30, PEN=  0
POINT NO. 5: X=  33.30, Y=  35.05, PEN=  0
POINT NO. 6: X=  53.95, Y=  41.55, PEN=  0
POINT NO. 7: X=  76.30, Y=  45.25, PEN=  0
POINT NO. 8: X= 101.45, Y=  45.65, PEN=  0
POINT NO. 9: X= 113.60, Y=  45.65, PEN=  0
POINT NO.10: X= 113.60, Y=  45.65, PEN=  1
[DONE]
```

4164-24

Figure 7-5. Sample GPIB CALL Digitize Program with SRQ Messages.

Call Digitizing via GPIB Interface (Without SRQ Messages)

Call digitizing via the GPIB can be performed without the need for or generation of SRQ messages. (If a CALL DIGITIZE command is sent prior to the operator pressing the CALL switch for each point, the plotter responds with data without sending an SRQ message.)

A typical call digitizing process without SRQ messages has these steps:

1. The PROMPT LIGHT ON command is used to turn on the PROMPT light, telling the plotter operator that digitizing is permitted.
2. For each point, the GPIB controller sends a CALL DIGITIZE command to request data.

3a. For each point, the operator uses the joystick to position the digitizing reticle (used in place of a pen in the pen carriage) over the desired point and then presses the front-panel CALL switch. This instructs the plotter to store the point.

3b. (Optional) The plotter operator may press and hold the CALL switch until the plotter's bell rings (which produces a "status block") to indicate the end of digitizing. However, this action assumes that the host knows the meaning of the "status block." (For some controllers special handling of data is required, since the "status block" contains only one value rather than three.) The "status block" is distinquished from a digitizing response block because the status block has only one value rather than three.

4. For each point, the plotter sends one output block of information consisting of X and Y position values and a value indicating whether the pen is up or down. If Step 3b is used, the plotter sends a status block, consisting of one value rather than three.

   If this is NOT the last point to be digitized, the process repeats, starting back at Step 2.

5. The PROMPT LIGHT OFF command turns off the PROMPT light telling the operator to stop digitizing.

This completes the call digitizing process.

The sample 4050 Series program in Figure 7-6 uses this procedure to store points and to display the results on the 4050 Series display screen.

NOTE

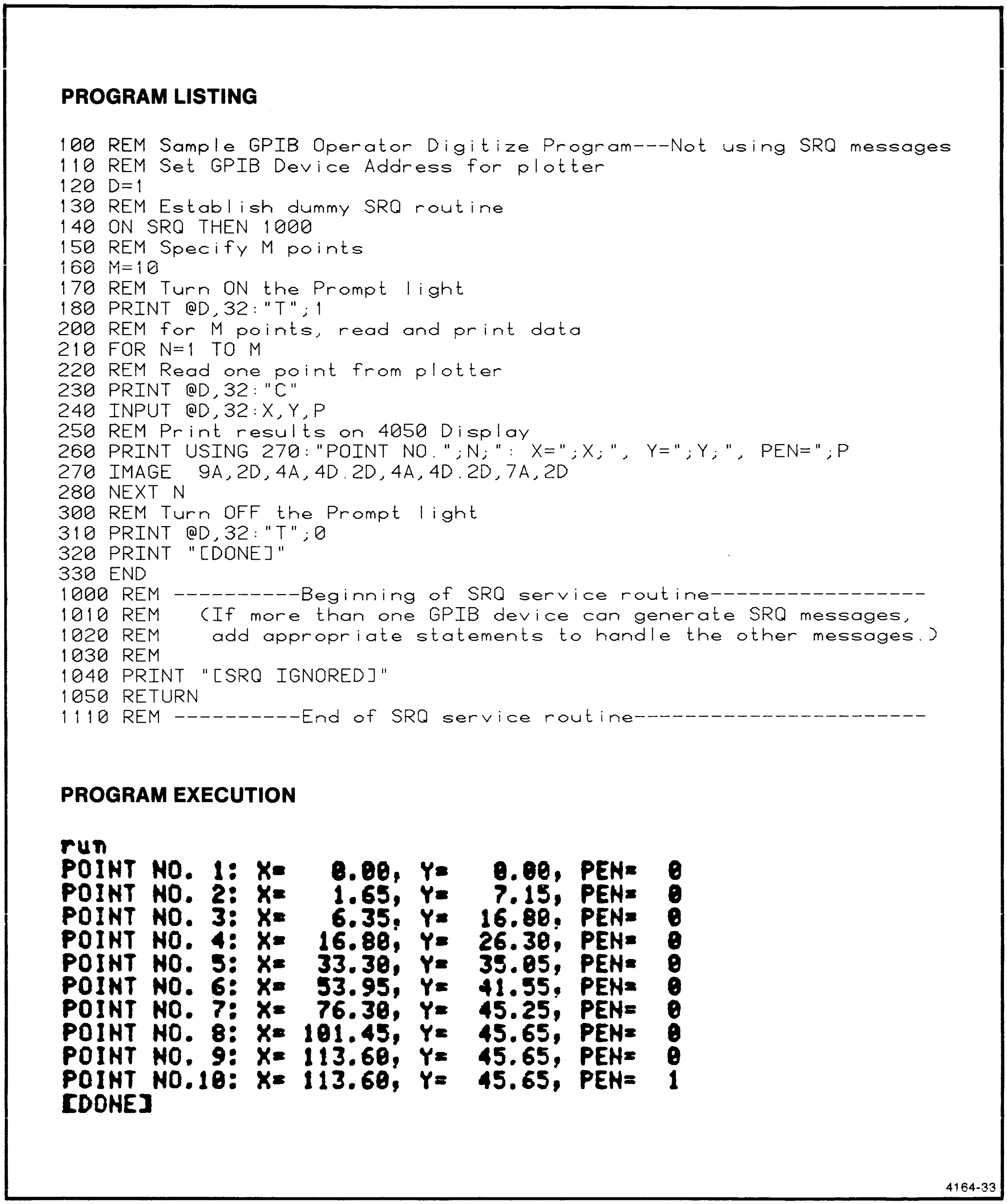
If the plotter operator pushes the CALL switch again before the GPIB controller has been able to send the next CALL DIGITIZE command, an SRQ message results. If you are using a TEKTRONIX 4050 Series controller or any controller that recognizes SRQs, a "do nothing" SRQ service routine is required to satisfy the unexpected SRQs.

DIGITIZING COMMAND DESCRIPTIONS

There are three Digitizing commands:

- DIGITIZE
- CALL DIGITIZE
- PROMPT LIGHT ON/OFF

Each command is described separately in the pages that follow.



PROGRAM LISTING

```
100 REM Sample GPIB Operator Digitize Program---Not using SRQ messages
110 REM Set GPIB Device Address for plotter
120 D=1
130 REM Establish dummy SRQ routine
140 ON SRQ THEN 1000
150 REM Specify M points
160 M=10
170 REM Turn ON the Prompt light
180 PRINT @D,32:"T";1
200 REM for M points, read and print data
210 FOR N=1 TO M
220 REM Read one point from plotter
230 PRINT @D,32:"C"
240 INPUT @D,32:X,Y,P
250 REM Print results on 4050 Display
260 PRINT USING 270:"POINT NO.";N;": X=";X;", Y=";Y;", PEN=";P
270 IMAGE  9A,2D,4A,4D.2D,4A,4D.2D,7A,2D
280 NEXT N
300 REM Turn OFF the Prompt light
310 PRINT @D,32:"T";0
320 PRINT "[DONE]"
330 END
1000 REM ----------Beginning of SRQ service routine-----------------
1010 REM   (If more than one GPIB device can generate SRQ messages,
1020 REM    add appropriate statements to handle the other messages.)
1030 REM
1040 PRINT "[SRQ IGNORED]"
1050 RETURN
1110 REM ----------End of SRQ service routine-----------------------
```

PROGRAM EXECUTION

```
run
POINT NO. 1: X=   0.00, Y=   0.00, PEN=  0
POINT NO. 2: X=   1.65, Y=   7.15, PEN=  0
POINT NO. 3: X=   6.35, Y=  16.80, PEN=  0
POINT NO. 4: X=  16.80, Y=  26.30, PEN=  0
POINT NO. 5: X=  33.30, Y=  35.05, PEN=  0
POINT NO. 6: X=  53.95, Y=  41.55, PEN=  0
POINT NO. 7: X=  76.30, Y=  45.25, PEN=  0
POINT NO. 8: X= 101.45, Y=  45.65, PEN=  0
POINT NO. 9: X= 113.60, Y=  45.65, PEN=  0
POINT NO.10: X= 113.60, Y=  45.65, PEN=  1
[DONE]
```

4164-33

Figure 7-6. Sample GPIB CALL Digitize Program Without SRQ Messages.

DIGITIZE

Purpose: This command causes the plotter to send one output message block containing current X and Y position values and a pen status value indicating whether the pen is up or down. Position values are expressed in GDUs.

GPIB

(One Output Message Block Produced)

```
G
```

4050 BASIC (Data Byte Form)

```
          PRINT@pa,32:"G"
          INPUT@pa,32:x-value-target-variable,
                      y-value-target-variable,
                      pen-status-target-variable
Examples: PRINT@1,32:"G"
          INPUT@1,32:X,Y,P
```

4050 BASIC (Secondary Address Form)

```
          PRINT@pa,24:
          INPUT@pa,32:x-value-target-variable,
                      y-value-target-variable,
                      pen-status-target-variable
        or
          INPUT@pa,24:x-value-target-variable,
                      y-value-target-variable,
                      pen-status-target-variable
Example:  PRINT@I,24:
          INPUT@1,32:X,Y,P
        or
          INPUT@1,24:X,Y,P
```

4050 BASIC (Special Keyword Form)

```
          GIN@pa:x-value-target-variable,
                 y-value-target-variable
Example:  GIN@1:X,Y
```

Outputs

x-value
X-position, in GDUs

value 2
Y-position, in GDUs

pen-status
0 = pen up
1 = pen down

References

Refer to Section 6, *Introduction to GPIB Commands and Responses*, for information on decoding output responses from the plotter.

Examples

The examples in the syntax boxes show how to obtain the current pen position values and pen status value. Note that for the GIN keyword, only two target variables are allowed; the pen status value is ignored.

CALL DIGITIZE

Purpose: The CALL DIGITIZE command is used to request position data stored by the front-panel CALL switch function during operator-digitizing actions. The plotter responds to the CALL DIGITIZE command by sending one output message block containing X and Y position values and a pen status value. Position values are expressed in GDUs.

GPIB

```
(One Output Message Block is Produced)
C
```

4050 BASIC (Data Byte Form)

```
          PRINT@pa,32:"C"
          INPUT@pa,32:x-value-target-variable,
                      y-value-target-variable,
                      pen-status-target-variable
Example:  PRINT@1,32:"C"
          INPUT@1,32:X,Y,P
```

4050 BASIC (Secondary Address Form)

```
          PRINT@pa,27:
          INPUT@pa,32:x-value-target-variable,
                      y-value-target-variable,
                      pen-status-target-variable
      or
          INPUT@pa,27:x-value-target-variable,
                      y-value-target-variable,
                      pen-status-target-variable
Examples: PRINT@I,27:
          INPUT@1,32:X,Y,P
      or
          INPUT@1,27:X,Y,P
```

Outputs

x-value
X-position, in GDUs

y-value
Y-position, in GDUs

pen-status
0 = pen up
1 = pen down

Comments

The CALL DIGITIZE command may be used in two ways (the two methods differ in regard to generation and use of GPIB SRQ messages):

- If the CALL DIGITIZE command is sent *AFTER* the CALL switch is pushed, the plotter generates an SRQ message when the CALL switch is pressed. The controller then sends a CALL DIGITIZE command to obtain the one block output message in response to the SRQ message.
- If the CALL DIGITIZE command is sent *BEFORE* the CALL switch is pressed, the plotter sends one output block message as soon as data is available. No SRQ message is sent. (Note: If the operator reacts too quickly and presses the CALL switch storing another point before the next CALL DIGITIZE command, the plotter sends an SRQ message. Therefore, a "do nothing" SRQ service routine may be needed for controllers that respond to SRQ messages.)

References

Refer to Section 6 for information on decoding output responses from the plotter.

Examples

The examples in the syntax boxes show how to acquire data produced by pressing the front-panel CALL switch.

PROMPT LIGHT ON/OFF

Purpose: This command turns the front-panel PROMPT light on or off.

GPIB

```
T selector
```

4050 BASIC (Data Byte Form)

```
          PRINT@pa,32:"T";selector
Example:  PRINT@1,32:"T";1
```

4050 BASIC (Secondary Address Form)

```
          PRINT@pa,26:selector
Example:  PRINT@1,26:1
```

Parameters

selector
Prompt light selector, 0 — 1:

- 1 = ON
- 0 = OFF

Comments

On Option 31 equipped plotters, the front-panel PAUSE switch overrides the PROMPT light indication set by this command. The PAUSE function causes the PROMPT light to blink at a one-second rate. The PROMPT light returns to the correct condition when RESUME is pressed.

The PROMPT light is turned off by the DEVICE RESET (IFC) command.

Examples

The examples in the syntax boxes turn the PROMPT light on.

MULTIPLE PEN (OPTION 31) COMMANDS

INTRODUCTION

The Multiple Pen Option (Option 31) adds an eight-pen capability to the plotter. With this option, multi-colored plots can be made. Although the front-panel switches allow the operator to select pens, the host can also select pens under program control. In addition, the host can also program the plotter to plot at the best speed for a pen in order to match its ink flow characteristics to the media used. The programmed speed is the pen's terminal velocity on long vectors (see the specifications in the *4662 Interactive Digital Plotter Operator's Manual*) and is measured in mm/second.

CONCEPTS

The SELECT PEN command permits the host to select one of eight pens stored in the rotary pen turret of Option 31 equipped plotters.

The PEN SPEED command permits the host to select a terminal velocity for the current pen (this velocity is in effect after the pen's initial acceleration and prior to its deceleration at the end of a vector). The pen's terminal velocity may be selected to match the particular ink flow characteristics of that pen/media used. In plotters with Firmware Level 5 and above, host-selected terminal speeds of less than 220 mm/second automatically set the plotter to SLOW acceleration regardless of the current front-panel SLOW or FAST (acceleration) switch settings. On the other hand, if the host-selected terminal speed is greater than 220 mm/second, the current front-panel SLOW or FAST switch setting becomes active. Therefore:

- For speeds less than 220 mm/second, only SLOW acceleration is used (available in plotters with Firmware Level 5 and above).
- For speeds greater than 220 mm/second, SLOW or FAST acceleration can be chosen.

MULTIPLE PEN (OPTION 31) COMMAND DESCRIPTIONS

Two Option 31 commands are described here:

- SELECT PEN
- PEN SPEED

SELECT PEN

Purpose: This command causes the plotter to select the specified pen from the rotary pen turret or to store the current pen in the rotary pen turret.

GPIB

```
BPpen-selector
```

4050 BASIC (Data Byte Form)

```
          PRINT@pa,32:“BP”;pen-selector
Example:  PRINT@1,32:“BP”;1
```

4050 BASIC (Secondary Address Form)

```
          PRINT@pa,8:pen-selector
Example:  PRINT@1,8:1
```

Parameters

pen-selector

Pen position in the rotary pen turret 1 to 8, or 0, which means store the pen presently in the pen carriage in the rotary pen turret.

Comments

The pen position (selector) matches the pen numbers printed on the rotary pen turret cover. Out-of-range arguments generate a command/response error.

Examples

The examples in the syntax boxes cause the pen carriage to deposit any pen it has and pick up the pen located in Pen Position 1 of the rotary pen turret.

PEN SPEED

Purpose: This command causes the plotter to limit its terminal velocity to a value less than or equal to the specified value.

GPIB

```
               BYpen-speed
```

4050 BASIC (Data Byte Form)

```
               PRINT@pa,32:"BY";pen-speed
     Example:  PRINT@1,32:"BY";280
```

Parameters

pen-speed

Terminal velocity of the pen, in mm/second. Ranges from 10 to 570 mm/second, in steps of 10 mm/second.

Comments

Pen speeds are rounded to the nearest 10 mm/second. This gives 57 different speeds ranging from 10 mm/second to 570 mm/second. Negative or zero values generate a command/response error and cause the command to be ignored.

Values greater than 570 mm/second are treated as 570 mm/second.

In plotters with Firmware Level 5 or higher, the plotter's acceleration is reduced to one-half of normal to improve line quality whenever speeds of less than 220 mm/second are programmed.

Default value at power-up or after a DEVICE RESET command is 570 mm/second.

Examples

The examples in the syntax boxes cause the plotter to operate at approximately one-half speed.

Appendix A

SERIAL COMMAND SUMMARY

Command	**Command Format***
PLOTTER ON	${}^{E}{}_{C}{}^{A}{}_{D}$**E**
PLOTTER OFF	${}^{E}{}_{C}{}^{A}{}_{D}$**F**
BLOCK START	${}^{E}{}_{C}{}^{A}{}_{D}$**(**
BLOCK END	${}^{E}{}_{C}{}^{A}{}_{D}$**)**checksum-value-integer
SET TURNAROUND DELAY	${}^{E}{}_{C}{}^{A}{}_{D}$**G**delay-time-integer
SET BLOCK SIZE	${}^{E}{}_{C}{}^{A}{}_{D}$**H**maximum-block-size-integer
SET BYPASS CANCEL CHARACTER	${}^{E}{}_{C}{}^{A}{}_{D}$**U**bypass-cancel-character
SET SIGNATURE CHARACTER	${}^{E}{}_{C}{}^{A}{}_{D}$**S**signature-character
SET PROMPT CHARACTER	${}^{E}{}_{C}{}^{A}{}_{D}$**R**prompt-character
DEVICE RESET	${}^{E}{}_{C}{}^{A}{}_{D}$**N**
READ STATUS	${}^{E}{}_{C}{}^{A}{}_{D}$**O**status-register-selector
SIZE	${}^{E}{}_{C}{}^{A}{}_{D}$**Q**
ALPHA MODE	${}^{U}{}_{S}$
SET ALPHA DIMENSION	${}^{E}{}_{C}{}^{A}{}_{D}$**I**character-space,line-space
SET ALPHA ROTATION	${}^{E}{}_{C}{}^{A}{}_{D}$**J**rotation-angle
SELECT ALPHA FONT	${}^{E}{}_{C}{}^{A}{}_{D}$**T**font-selector
PRINT	${}^{U}{}_{S}$text
RESET ALPHA PARAMETERS	${}^{E}{}_{C}{}^{A}{}_{D}$**V**
MOVE-TO-HOME	${}^{E}{}_{C}{}^{F}{}_{F}$
GRAPH MODE	${}^{G}{}_{S}$
MOVE [DRAW . . .]	${}^{G}{}_{S}$encoded-style-II-XY-move-coordinate-pair [encoded-style-II-XY-draw-coordinate-pair] . . .
or DRAW	${}^{G}{}_{S}{}^{B}{}_{L}$encoded-style-II-XY-draw coordinate-pair . . .
DIGITIZE	${}^{E}{}_{C}{}^{A}{}_{D}$**M**
PROMPT LIGHT ON	${}^{E}{}_{C}{}^{A}{}_{D}$**K**
PROMPT LIGHT OFF	${}^{E}{}_{C}{}^{A}{}_{D}$**L**
SELECT PEN	${}^{E}{}_{C}{}^{A}{}_{D}$**BP**pen-selector
PEN SPEED	${}^{E}{}_{C}{}^{A}{}_{D}$**BY**pen-speed-in-mm/second
SET PLOTTER UNITS	${}^{E}{}_{C}{}^{A}{}_{D}$**BV**units-selector (0 or 3)

*where ${}^{A}{}_{D}$ is **A, B, C,** or **D.**

Appendix B

GPIB COMMAND SUMMARY

READ STATUS Command

GPIB	**V** status-register-selector
DAB	**PRINT**@pa,**32:"V"**;status-register-selector **INPUT**@pa,**32:**status-information-target-variable
MSA	**PRINT**@pa,**0:**status-register-selector **INPUT**@pa,**32:**status-information-target-variable

SIZE Command

GPIB	**I**
DAB	**PRINT**@pa,**32:"I"** **INPUT**@pa,**32:**x-size-target-variable, y-size-target-variable, identification-word-target-variable
MSA	**INPUT**@pa,**13:**x-size-target-variable, y-size-target-variable, identification-word-target-variable
KEYWORD	**INPUT**@pa:x-size-target-variable, y-size-target-variable, identification-word-target-variable

SET ALPHA DIMENSION Command

GPIB	**S** character-space,line-space
DAB	**PRINT**@pa,**32:"S"**;character-space,line-space
MSA	**PRINT**@pa,**17:**character-space,line-space

SET ALPHA ROTATION Command

GPIB	**R** rotation-angle
DAB	**PRINT**@pa,**32:"R"**;rotation-angle
MSA	**PRINT**@pa,**25:**rotation-angle

SELECT ALPHA FONT Command

GPIB	**F** rotation-angle
DAB	**PRINT**@pa,**32:"F"**;font-selector
MSA	**PRINT**@pa,**18:**font-selector

PRINT Command

GPIB	**P** print-string
DAB	**PRINT** pa,**32:"P**print-string**"**;
MSA	**PRINT**@pa,**12:**"print-string";
KEYWORD	**PRINT** @pa:"print-string";

RESET ALPHA PARAMETERS Command

GPIB	**A**
DAB	**PRINT** pa,**32:"A"**
MSA	**PRINT**@pa,**7:**

MOVE-TO-HOME Command

GPIB	**H**
DAB	**PRINT**@pa,**32:"H"**
MSA	**PRINT**@pa,**23:**
KEYWORD	**HOME**@pa: or **PAGE**@pa:

MOVE Command

GPIB	**M** x-value,y-value
DAB	**PRINT**@pa,**32:"M"**;x-value,y-value
MSA	**PRINT**@pa,**21:**x-value,y-value
KEYWORD	**MOVE**@pa:x-value,y-value or **RMOVE**@pa:x-relative-value,y-relative-value

DRAW Command

GPIB	**D** x-value,y-value[,x-value,y-value] . . .
DAB	**PRINT**@pa,**32:‘D‘**;x-value;y-value [;x-value;y-value] . . .
MSA	**PRINT**@pa,**20:**x-value;y-value [;x-value;y-value] . . .
KEYWORD	**DRAW**@pa:x-value,y-value or **RDRAW**@pa:x-relative-value,y-relative-vaiue

DIGITIZE Command

GPIB	**G**
DAB	**PRINT**@pa,**32:“G”** **INPUT**@pa,**32:**x-value-target-variable, y-value-target-variable, pen-status-target-variable
MSA	**PRINT**@pa,**24:** **INPUT**@pa,**32:**x-value-target-variable, y-value-target-variable, pen-status-target-variable or **INPUT** pa,**24:**x-value-target-variable, y-value-target-variable, pen-status-target-variable
KEYWORD	**GIN**@pa:x-value-target-variable, y-value-target-variable

CALL DIGITIZE Command

GPIB	**C**
DAB	**PRINT**@pa,**32:“C”** **INPUT**@pa,**32:**x-value-target-variable, y-value-target-variable, pen-status-target-variable
MSA	**PRINT**@pa,**27:** **INPUT**@pa,**32:**x-value-target-variable, y-value-target-variable, pen-status-target-variable or **INPUT**@pa,**27:**x-value-target-variable, y-value-target-variable, pen-status-target-variable

PROMPT LIGHT ON/OFF Command

X GPIB	**T1** (for PROMPT light on) **T0** (for PROMPT light off)
DAB	**PRINT**@pa,**32:“T”;1** (light on) **PRINT**@pa,**32:“T”;0** (light off)
MSA	**PRINT**@pa,**26:1** (light on) **PRINT**@pa,**26:0** (light off)

SELECT PEN Command

GPIB	**BP** pen-selector
DAB	**PRINT**@pa,**32:“BP”**;pen-selector
MSA	**PRINT**@pa,**8:**pen-selector

PEN SPEED Command

GPIB	**BY**pen-speed
DAB	**PRINT**@pa,**32:“BY”**;pen-speed-in-mm/second

Appendix C

ASCII CODE CHART

NOTE

Two other code charts are presented in this manual: an ASCII encoding chart in Section 3, and a GPIB code chart in Section 6. 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 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 [illegible]	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

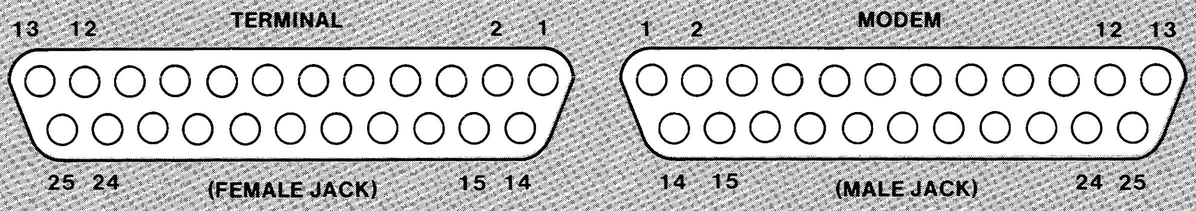
octal	25	
	NK	graphic representation
hex	15 NAK 21	decimal
	mnemonic	

3806-60

Appendix D

SERIAL INTERFACE CONNECTORS

Refer to Figure D-1 for the pin/signal name relationships of the Serial (RS-232-C) Interface connectors.



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

* Pins marked with asterisks are held passively true through 4.7 ohms to +12 Volts.

4164-36

Figure D-1. Serial Interface Connector.

Appendix E

AREA FILL PATTERNS

RECOMMENDATIONS FOR SOLID-AREA FILL PLOTS

PLOTS ON PAPER

Satisfactory solid-area fill plots with good uniformity can be achieved using water-soluble fiber-tip pens.

When doing multicolor plots, whenever possible, light colors should be plotted first and darker colors last to minimize the bleeding of one color into another. It is also helpful to leave a small gap (approximately 0.010 inch, 0.25 mm) between solid areas and adjoining lines, or between areas of different colors.

The plotter should be operated at full speed and acceleration (default values) using one of the two solid-area fill patterns shown in Figure E-1. The pen strokes can be either vertical or horizontal. The pattern in Figure E-1A is recommended because the pattern in Figure E-1B may exhibit some shading at either end of the strokes (where the pen is raised and lowered). Spacing between the pen strokes in the solid-area fill pattern should be 0.010 inch (2.54 mm). Table E-1 shows the proper pen-stroke spacing to use with either the GPIB or Serial Interfaces. Table E-1 assumes the default 10 by 15 inch page. If the page is scaled by the user with the front-panel SET switches, new ADU or GDU spacing values must be calculated to provide the required 0.010-inch pen-stroke spacing.

Table E-1

PLOTTER UNIT STEPS FOR SOLID-AREA FILL ON PAPER

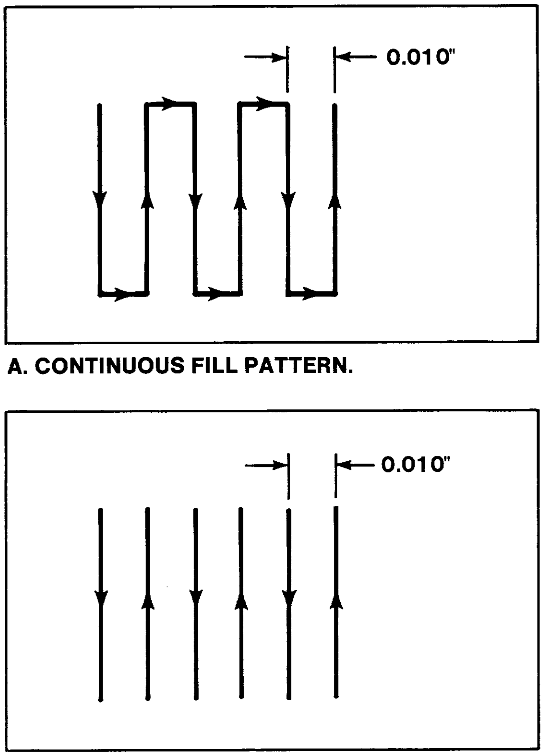
Interface	Plotter Unit Steps Between Pen Strokes
Serial (in ADUs)	2
Serial (modified ADUs[a])	2
GPIB (in GDUs)	0.10

[a] See Serial Interface SELECT PLOTTER UNITS command.

TRANSPARENT FILMS

Permanent-ink fiber-tip pens should be used when plotting solid-area fills on transparent film media. This combination results in excellent line and solid-area fill plots and exhibits little patterning even when projected through an overhead projector.

When doing multicolor plots, whenever possible, plot light colors first and darker colors last to minimize the bleeding of one color into another. It is also helpful to leave a small gap (approximately 0.010 inch or 0.254 mm) between filled areas and adjoining lines, or between areas of different colors.



A. CONTINUOUS FILL PATTERN.

B. BIDIRECTIONAL FILL PATTERN

Figure E-1. Solid-Area Fill Patterns (for Paper).

Plotters without Option 31 should be operated at a slow speed and acceleration (selected by rear-panel switches).

A PEN SPEED command accurately controls the speed of plotters equipped with Option 31. Select a pen speed of 100 mm/second (refer to the PEN SPEED command). In plotters with Firmware Level 5 or higher, this also automatically reduces the plotter's acceleration. Option 31 equipped plotters with firmware levels less than 5 must have the acceleration set using the front-panel SLOW switch.

NOTE

Reduced speed and acceleration is recommended for all plotting with permanent ink fiber-tip pens, not just for plotting solid area fill patterns.

Figure E-2 shows the pen-stroke patterns recommended to achieve the best results for solid-area fills on transparent film. The pen strokes can be either vertical or horizontal. The pattern in Figure E-2A is recommended because the pattern in Figure E-2B tends to exhibit shading in the filled area at each end of the pen strokes (where the pen is raised and lowered). Spacing between the pen strokes should be 0.005 inch (1.27 mm). Table E-2 shows the proper plotter unit steps required to use with either the GPIB or Serial Interface. This assumes that the page size is the default, 10 by 15 inches. If the page is scaled by use of the front-panel SET switches, new plotter unit spacing must be calculated to provide the required 0.005-inch spacing.

NOTE

Modified ADUs should be selected (see the Serial Interface SELECT PLOTTER UNITS command), and the plotter's default (power-up) page size should be used. This enables the host to directly address the plotter's maximum mechanical resolution with consistent spacing between the lines in the fill pattern, which cannot be achieved using default standard ADUs.

Table E-2

PLOTTER UNIT STEPS FOR SOLID-AREA FILL ON TRANSPARENT FILMS

Interface	Plotter Unit Steps Between Pen Strokes
Serial (in ADUs)	1
Serial (modified ADUs[a])	1
GPIB	0.05

[a] See the Serial Interface SELECT PLOTTER UNITS command.

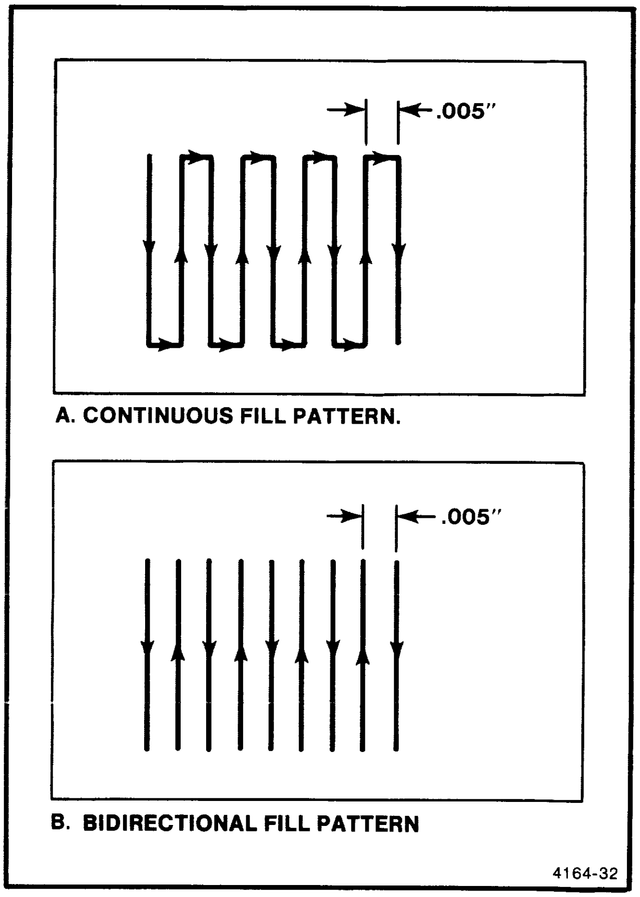


Figure E-2. Solid-Area Fill Patterns (for Transparent Films).

RECOMMENDATIONS FOR CROSS-HATCHING

With the exception of the fill pattern and spacing, all of the information on pens, media, and pen speed described under *Recommendations for Solid-Area Fill Plots* applies to cross-hatched filling. The only additional consideration is that for consistent line width in the fill pattern, all lines should be drawn in same direction (see Figure E-3).

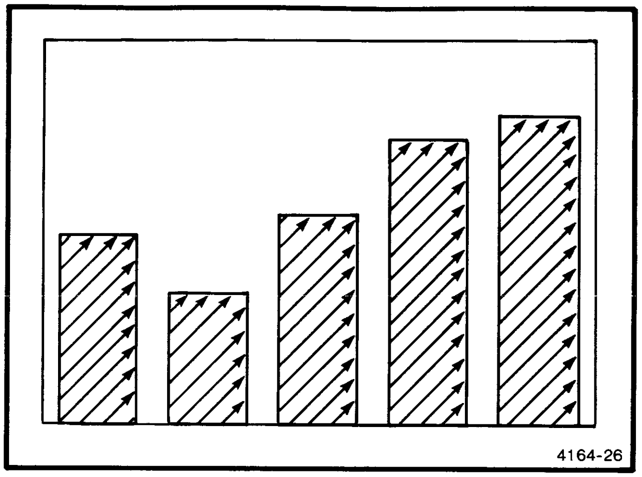


Figure E-3. Cross-Hatching Pattern.

Appendix F

COORDINATE CONVERSION CHART

INTRODUCTION

Tables F-1 and F-2 are designed for use with 12- and 10-bit graphing respectively. Use the appropriate procedure described for your desired coordinate resolution.

10-BIT GRAPHING

This is the appropriate resolution for the TEKTRONIX 4010, 4012, or 4013 Computer Display Terminals. Simply use Table F-2 (Coordinate Conversion Chart). Find the desired X or Y coordinate in the body of the chart; follow that column to the bottom to find the ASCII character which represents the HIY or HIX byte; go to the 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

For example, 480X,100Y would be **SP y SN X** in ASCII code.

12-BIT GRAPHING

This is the appropriate resolution for the TEKTRONIX 4014 or 4015 Computer Display Terminals. This requires Table F-1 (Interpolation Insert Chart) and Table F-2 (Coordinate Conversion Chart). Simply find the largest coordinate value in Table F-2 that is equal to or less than the desired coordinate value; follow that column to the bottom of the chart to find the ASCII character which 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 the 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 (X remainder = 1, Y remainder = 3). Notice that these values 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:

HIY,EB,LOY,HIX,LOX

(Figure F-1 shows the increased resolution gained by the extra byte.)

For 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)

		X Remainder			
		0	1	2	3
Y Remainder	3	l	m	n	o
	2	h	i	j	k
	1	d	e	f	g
	0	‘	a	b	c

Table F-2

COORDINATE CONVERSION CHART

Low Order X		X or Y Coordinate								Low Order Y	
ASCII	DEC.									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	}
Λ	94	120	248	376	504	632	824	888	1016	126	~
_	95	124	252	380	508	636	828	892	1020	127	rubout D_T
DEC. ------>		32	33	34	35	36	37	38	39		
ASCII------>		S_P	!	"	#	$	%	&	'		

High Order X & Y

Table F-2 (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-2 (cont)

COORDINATE CONVERSION CHART

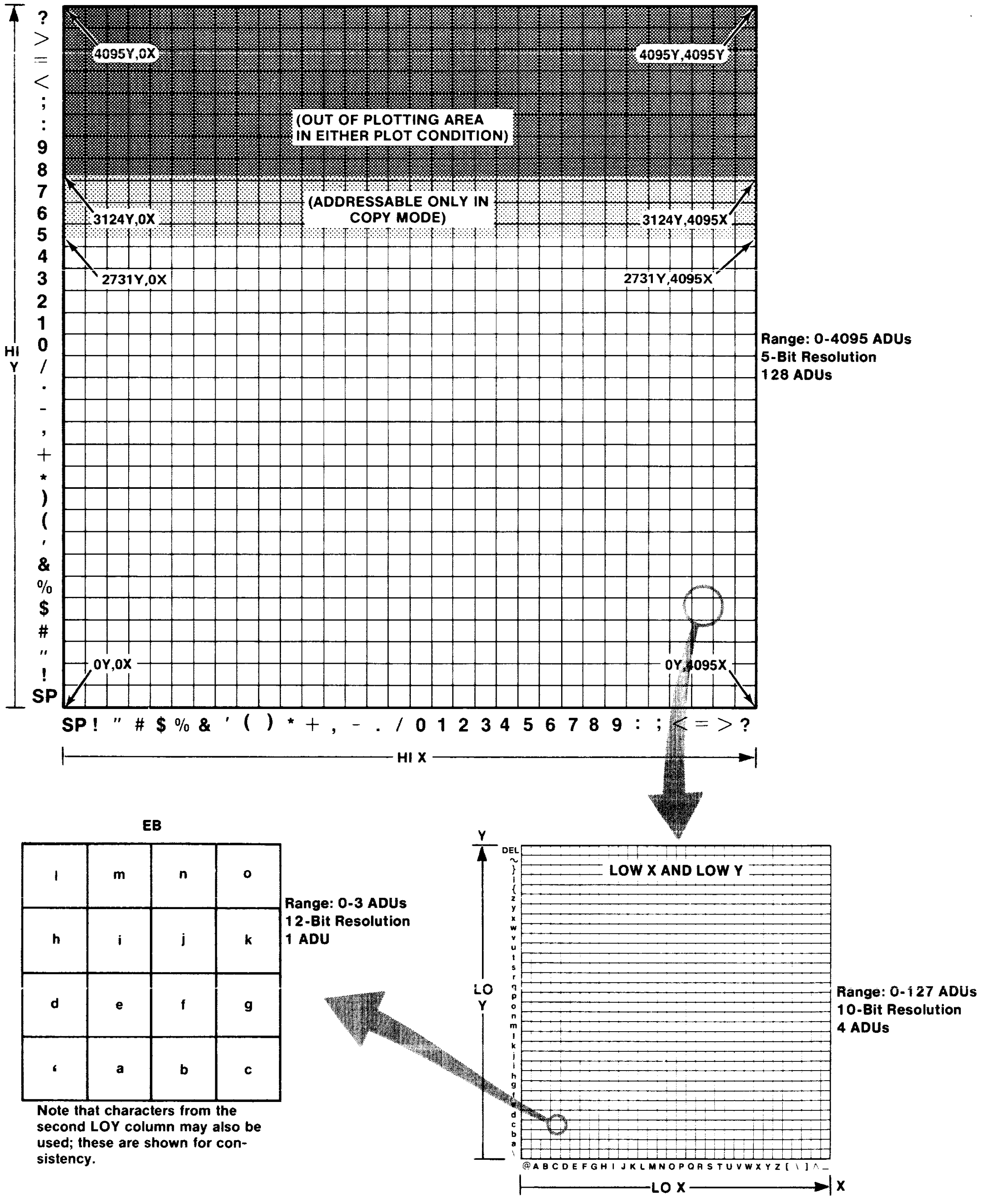
Low Order X										Low Order Y	
ASCII	DEC.	X or Y Coordinate								DEC.	ASCII
@	64	2048	2176	2304	2432	2560	2688	2816	2944	96	`
A	65	2052	2180	2308	2436	2564	2692	2820	2948	97	a
B	66	2056	2184	2312	2440	2568	2696	2824	2952	98	b
C	67	2060	2188	2316	2444	2572	2700	2828	2956	99	c
D	68	2064	2192	2320	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-2 (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


4095Y,0X
4095Y,4095Y
(OUT OF PLOTTING AREA IN EITHER PLOT CONDITION)
(ADDRESSABLE ONLY IN COPY MODE)
3124Y,0X
3124Y,4095X
2731Y,0X
2731Y,4095X
0Y,0X
0Y,4095X
Range: 0-4095 ADUs
5-Bit Resolution
128 ADUs
HI Y
? > = < ; : 9 8 7 6 5 4 3 2 1 0 / . - , + *) (' & % $ # " ! SP
SP ! " # $ % & ' ( ) * + , - . / 0 1 2 3 4 5 6 7 8 9 : ; < = > ?
HI X
EB
l m n o
h i j k
d e f g
' a b c
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.
Y
LO Y
DEL ~ } | { z y x w v u t s r q p o n m l k j i h g f e d c b a `
LOW X AND LOW Y
Range: 0-127 ADUs
10-Bit Resolution
4 ADUs
@ 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 [\ ] ^ _
LO X
X

(1932)4164-27

Figure F-1. X-Y Coordinates (Showing the Increased Resolution Gained by the Extra Byte).

Appendix G

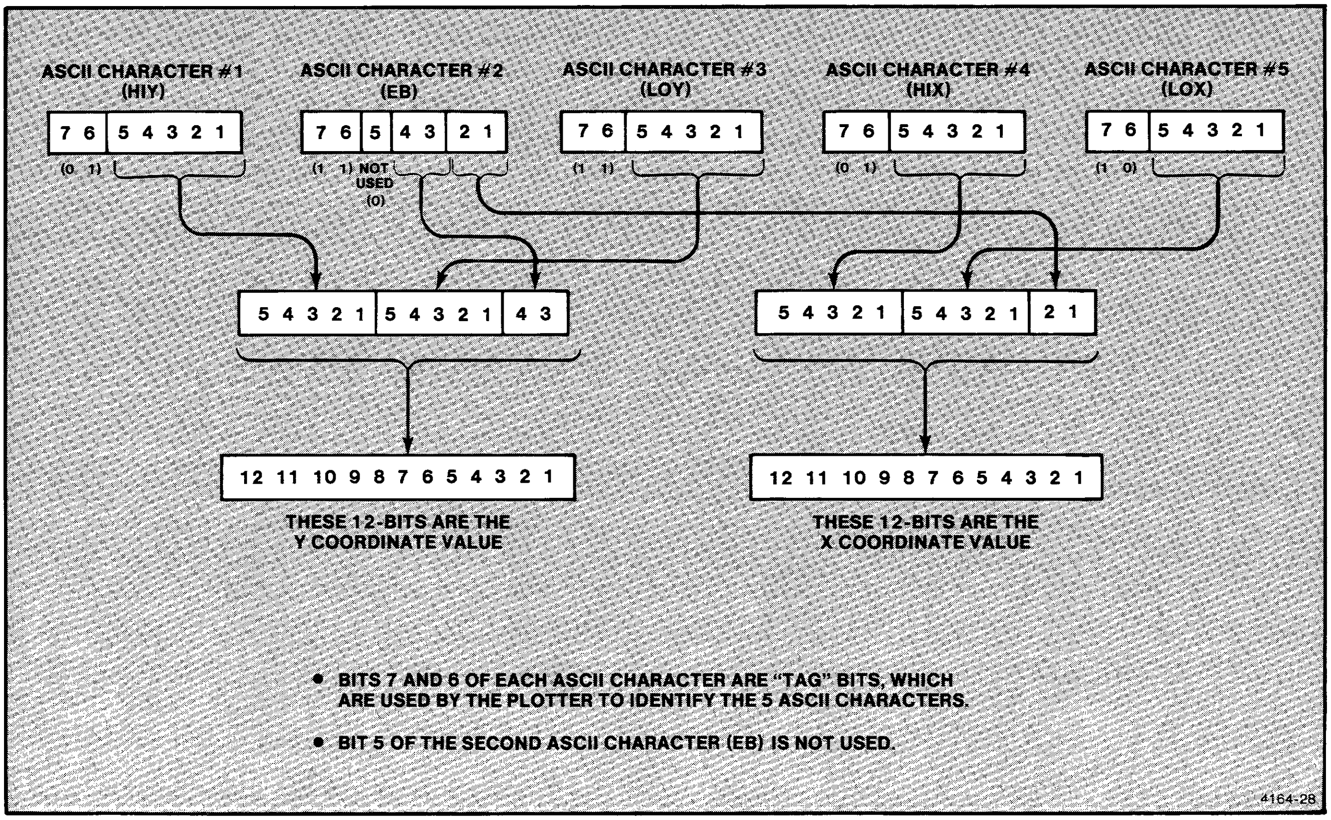
BASIC PROGRAM FOR COORDINATE CONVERSION

The following BASIC program can be included in a host (perhaps as a sub-routine) to convert host-generated graphing coordinates to ASCII characters arranged in the HIY, EB, LOY, HIX, and LOX sequence for the 4662 plotter.

The program generates plotter coordinates as binary numbers; each coordinate (X and Y) is represented by a 12-bit number. Therefore, the X and Y coordinates together make 24 bits, which are coded from five 7-bit ASCII characters as shown in Figure G-1. In the following BASIC program, the input data points are specified in the following form: X,Y, and Z, where X ranges from 0 to 4095, Y ranges 0 to 3124, and Z is either 0 or 1 (DRAW or MOVE).

NOTE

If a TEKTRONIX 4051 Graphic Computing System is used, it is convenient to use Z= 20 or 21 for DRAW and MOVE (change lines 530 and 540 to reflect this).



- BITS 7 AND 6 OF EACH ASCII CHARACTER ARE "TAG" BITS, WHICH ARE USED BY THE PLOTTER TO IDENTIFY THE 5 ASCII CHARACTERS.
- BIT 5 OF THE SECOND ASCII CHARACTER (EB) IS NOT USED.

Figure G-1. Conversion of HIY, EB, LOY, HIX, and LOX to X and Y Coordinates.

Lines 210 and 220 need to be modified to match your input host-program requirements. For example, if a 4051 is used to read data from a tape, line 210 is eliminated and 220 is changed as follows:

```
220 INPUT@33:X,Y,Z
```

Line 580 outputs the ASCII characters to the plotter.

```
100 REM FNM IS THE MODULUS FUNCTION
110 REM WHERE MOD(A1,A2)=A1-SGN(X)[X]A2
120 REM WHERE X=A1/A2
130 FNM(N)=INT(N-SGN(N/A2)*INT(N/A2)*A2)
140 DIM O$(8)
150 REM SET G$ TO A GS (ADE 29)
160 G$=CHR(29)
170 REM SET O$ (OUTPUT STRING) TO A GS (G$)
180 O$=G$
190 ON EOF (0) THEN 620
200 REM READ THE DATA
210 PRINT "INPUT COORDINATES X,Y,Z"
220 INPUT X,Y,Z
230 REM CALCULATE FIRST CHARACTER
240 REM ICHAR(1)=MOD(KY/128,32)+32 (HIY)
250 A2=32
260 I1=FNM(Y/128)+32
270 I$=CHR(I1)
280 O$=O$&I$
290 REM CALCULATE SECOND CHARACTER (EB)
300 REM ICHAR(2)=MOD(KY),4)*4+MOD(KX,4)+96
310 A2=4
320 I2=FNM(Y)*4+FNM(X)+96
330 I$=CHR(I2)
340 O$=O$&I$
350 REM CALCULATE THIRD CHARACTER (LOY)
360 REM ICHAR(3)=MOD(KY/4,32)+96
370 A2=32
380 I3=FNM(Y/4)+96
390 I$=CHR(I3)
400 O$=O$&I$
410 REM CALCULATE FOURTH CHARACTER (HIX)
420 REM ICHAR(4)=MOD(KX/128,32)+32
430 A2=32
440 I4=FNM(X/128)+32
450 I$=CHR(I4)
460 O$=O$&I$
470 REM CALCULATE FIFTH CHARACTER (LOX)
480 REM ICHAR(5)=MOD(KX/4,32)+64
490 A2=32
500 I5=FNM(X/4)+64
510 I$=CHR(I5)
520 O$=O$&I$
530 REM IF Z=1 IT IS A MOVE (0 IS A DRAW)
540 IF Z=0 THEN 580
550 REM CHECK IF A MOVE, IF SO THEN INSERT A GS (G$)
560 O$=REP(G$,1,0)
570 REM OUTPUT THE CHARACTER
580 PRINTO$;
590 O$=""
600 GO TO 210
610 PRINT
620 PRINT "DONE"
630 END
```

If a program is desired that reads 4051 GDUs (with X ranging from 0 to 130 and Y ranging from 0 to 100), add the following two lines to achieve the proper scaling:

```
221 X=X/131*4095
222 Y=Y/100*3124
```

NOTE

Be sure that the plotter is set for Copy mode when using Y values greater than 2731; otherwise those vectors are clipped.

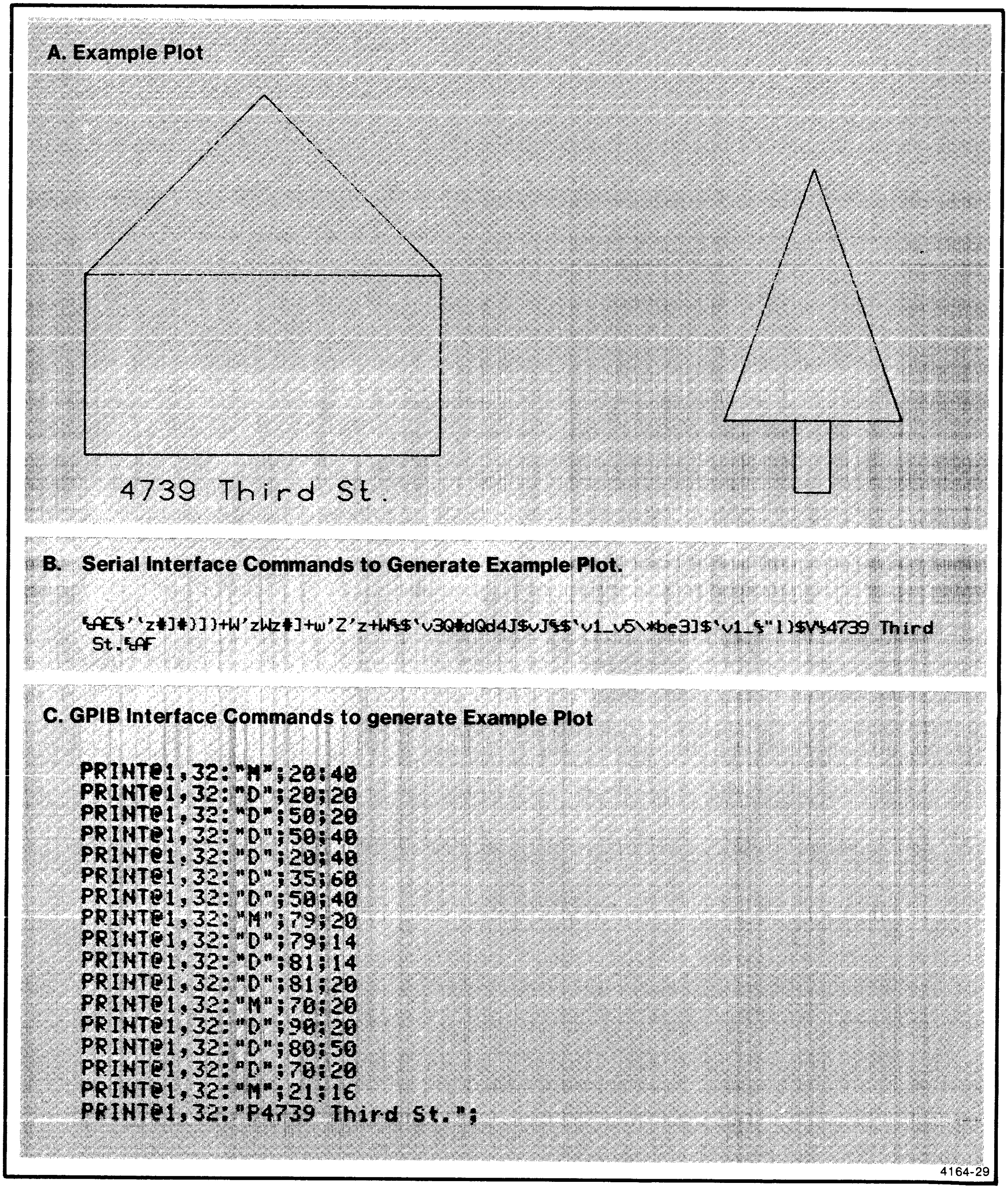
Appendix H

EXAMPLE PLOTS

Figure H-1A shows an example of a very simple plot that can be made using either the Serial or the GPIB Interface. The plot makes no attempt to demonstrate all commands, but it does demonstrate how several commands can be strung together.

Figure H-1B shows the series of commands necessary to produce Figure H-1A using the Serial Interface. Likewise, Figure H-1C shows the series of commands necessary to produce Figure H-1A using the GPIB Interface. The following pages explain each Serial and GPIB command used to generate Figure H-1A.

Figure H-2 shows the coordinates of the example plot shown in Figure H-1A.



A. Example Plot

B. Serial Interface Commands to Generate Example Plot.

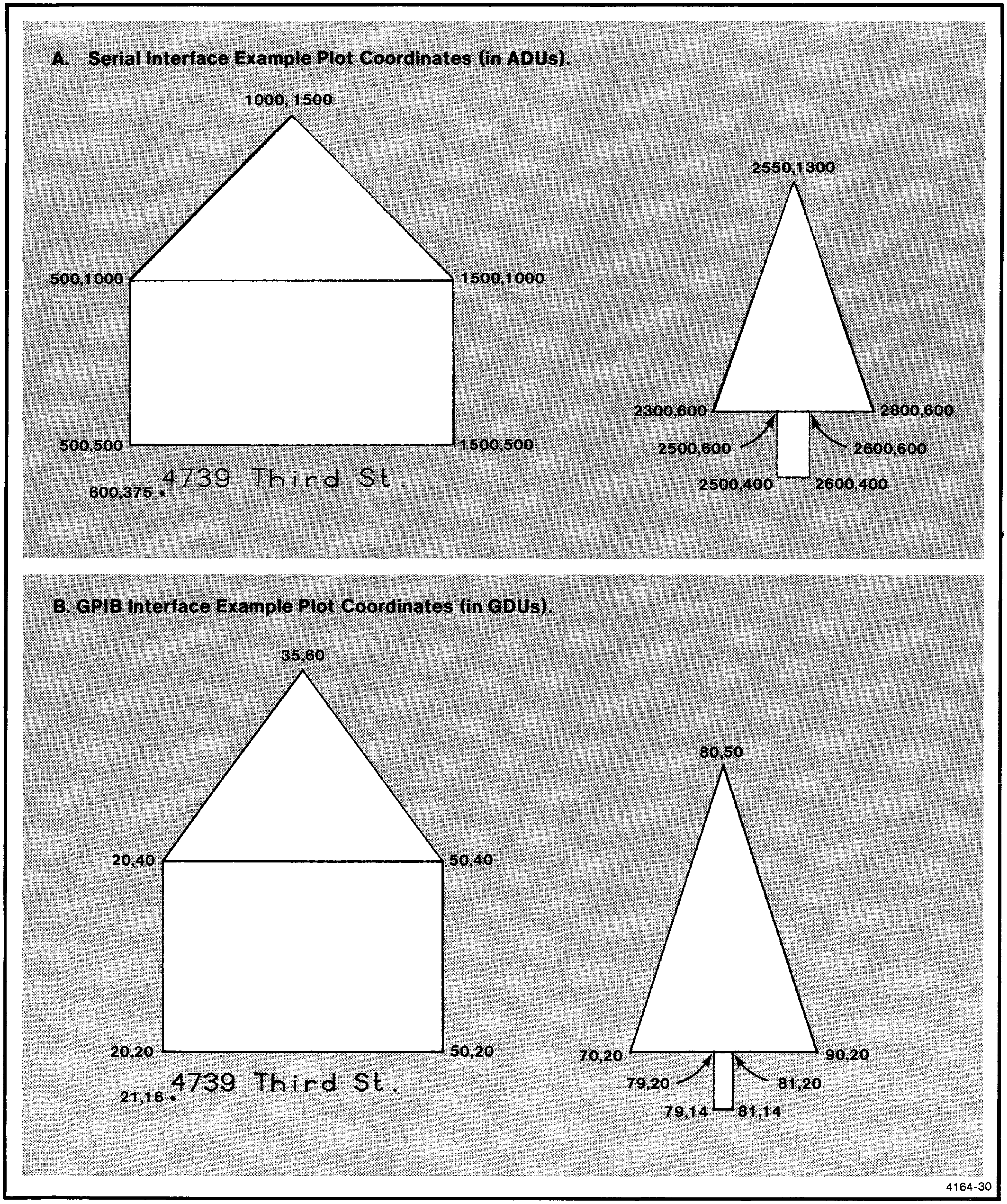
```
␛AE␛''z#]#)]]+W'zWz#]+w'Z'z+W␛$`v3Q#dQd4J$vJ␛$`v1_v5\*be3]$`v1_␛"1)$V␛4739 Third
 St.␛AF
```

C. GPIB Interface Commands to generate Example Plot

```
PRINT@1,32:"M";20;40
PRINT@1,32:"D";20;20
PRINT@1,32:"D";50;20
PRINT@1,32:"D";50;40
PRINT@1,32:"D";20;40
PRINT@1,32:"D";35;60
PRINT@1,32:"D";50;40
PRINT@1,32:"M";79;20
PRINT@1,32:"D";79;14
PRINT@1,32:"D";81;14
PRINT@1,32:"D";81;20
PRINT@1,32:"M";70;20
PRINT@1,32:"D";90;20
PRINT@1,32:"D";80;50
PRINT@1,32:"D";70;20
PRINT@1,32:"M";21;16
PRINT@1,32:"P4739 Third St.";
```

4164-29

Figure H-1. Example Plot Made Using the Serial and GPIB Interfaces.



A. Serial Interface Example Plot Coordinates (in ADUs).

B. GPIB Interface Example Plot Coordinates (in GDUs).

Figure H-2. Coordinates of the Example Plot.

SERIAL INTERFACE

Command	Explanation
^{E}cAE	Turns on the plotter's interface logic and permits the plotter to interpret the following commands.
^{G}s’‘z#]	12-bit graphics move to coordinates 500,1000 (upper-left corner of rectangle).
#}]	Optimized graphics draw to coordinates 500,500 (lower-left corner of rectangle). Only the new HIY, LOY, and LOX bytes need to be sent — see Table 3-3.
}+W	Optimized graphics draw to coordinates 1500,500 (lower-right corner of rectangle). Only the new LOY, HIX, and LOX bytes need to be sent.
’zW	Optimized graphics draw to coordinates 1500,1000 (upper-right corner of rectangle). Only the new HIY, LOY, and LOX bytes need to be sent.
z#]	Optimized graphics draw to coordinates 500,1000 (upper-left corner of rectangle). Only the new LOY, HIX, and LOX bytes need to be sent.
+w’Z	Optimized graphics draw to coordinates 1000,1500 (upper corner of gable). Only the new HIY, LOY, HIX, and LOX bytes need to be sent).
’z+W	Optimized graphics draw to coordinates 1500,1000 (upper-right corner of rectangle). Only the new HIY, LOY, HIX, and LOX bytes need to be sent.
^{G}s$‘v3Q	12-bit graphics move to the coordinates 2500,600 (upper-left corner of tree trunk).
#dQ	Optimized graphics draw to coordinates 2500,400 (lower-left corner of tree trunk). Only the new HIY, LOY, and LOX bytes need to be sent.
d4J	Optimized graphics draw to coordinates 2600,400 (lower-right corner of the tree trunk). Only the new LOY, HIX, and LOX bytes need to be sent.
$vJ	Optimized graphics draw to coordinates 2600,600 (upper-right corner of the tree trunk). Only the new HIY, LOY, and LOX bytes need to be sent.
^{G}s$‘v1 _	12-bit graphics move to coordinates 2300,600 (lower-left corner of the triangle representing the tree).
v5	Optimized graphics draw to coordinates 2800,600 (lower-right corner of the triangle representing the tree). Only the new LOY, HIX, and LOX bytes need to be sent.
***be3]**	12-bit graphics draw to coordinates 2550,1300 (top of tree).
$‘v1 _	12-bit graphics draw to coordinates 2300,600 (lower-left corner of the triangle representing the tree).
^{G}s“|} $V	12-bit graphics move to coordinates 600,375 (below the house).
^{U}s4739 Third St.	Places the plotter in Alpha mode and prints the text “4739 Third St.”
^{E}cAF	Turns off the plotter's interface logic (the plotter will not respond to any more commands until another PLOTTER ON command is sent).

GPIB INTERFACE

Command	Explanation
PRINT@1,32:"M";20;40	Causes the plotter to Move to the coordinates 20,40 (upper-left corner of the rectangle).
PRINT@1,32:"D";20;20	Causes the plotter to Draw to the coordinates 20,20 (lower-left corner of the rectangle).
PRINT@1,32:"D";50;20	Causes the plotter to Draw to the coordinates 50,20 (lower-right corner of the rectangle).
PRINT@1,32:"D";50;40	Causes the plotter to Draw to the coordinates 50,40 (upper-right corner of the rectangle).
PRINT@1,32:"D";20;40	Causes the plotter to Draw to the coordinates 20,40 (upper-left corner of the rectangle).
PRINT@1,32:"D";35;60	Causes the plotter to Draw to the coordinates 35,60 (top of gable of house).
PRINT@1,32:"D";50;40	Causes the plotter to Draw to the coordinates 50,40 (upper-right corner of the rectangle).
PRINT@1,32:"M";79;20	Causes the plotter to Move to the coordinates 79,20 (upper-left corner of the tree trunk).
PRINT@1,32:"D";79;14	Causes the plotter to Draw to the coordinates 79,14 (lower-left corner of the tree trunk).
PRINT@1,32:"D";81;14	Causes the plotter to Draw to the coordinates 81,14 (lower-right corner of the tree trunk).
PRINT@1,32:"D";81;20	Causes the plotter to Draw to the coordinates 81,20 (upper-right corner of the tree trunk).
PRINT@1,32:"M";70;20	Causes the plotter to Move to the coordinates 70,20 (lower-left corner of the triangle representing the tree).
PRINT@1,32:"D";90;20	Causes the plotter to Draw to the coordinates 90,20 (lower-right corner of the triangle representing the tree).
PRINT@1,32:"D";80;50	Causes the plotter to Draw to the coordinates 80,50 (top of the tree).
PRINT@1,32:"D";70;20	Causes the plotter to Draw to the coordinates 70,20 (lower-left corner of the triangle representing the tree).
PRINT@1,32:"M";21;16	Causes the plotter to Move to the coordinates 21,16 (below the house).
PRINT@1,32:"P4739 Third St.";	Causes the plotter to Print the address below the house).

Appendix I

GLOSSARY

address character
(Serial) The second character of most commands. Can be set to ASCII characters A, B, C, or D.

ADE
See *ASCII decimal equivalent.*

ADUs
Addressable Device Units. The default plotter unit when using the Serial Interface.

Alpha mode
The plotter mode after receiving an ASCII $^{U}{}_{S}$ character. All subsequently received printing ASCII characters are printed by the plotter.

argument
A parameter associated with a command (e.g., for the angle of rotation or 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 as an argument separator.

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 $^{S}{}_{P}$, and 65 for A).

attention character
(Serial) The first character of most commands, which is $^{E}{}_{C}$ for the 4662 plotter.

Attention Action commands
(Serial) A command consisting of a series of two characters beginning with the attention character ($^{E}{}_{C}$).

auto move-to-home
The process of automatically performing a move to the home position. This process occurs whenever a PRINT command follows power-up 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) A 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 that make up an ASCII character.

Bypass function
The Bypass function directs the plotter to ignore echoed characters from a host.

character argument
A command argument consisting of a single ASCII character.

checksum
(Serial) Part of a Block mode block. A checksum is determined by summing the contents of the block and is used to detect transmission errors.

clipping
The process by which the portion of a plot that falls outside of the page is not plotted.

command code
One or two ASCII characters that identify a command.

command syntax
The conventions for a particular application (such as Serial, GPIB, or 4050 Series BASIC) that define how to form a command.

command terminator
A character that makes the end of the command. See reprocessed terminator.

Continuous mode
This is the plotter's default mode on power-up if DC1/DC3 Flagging has not been selected by the four rear-panel switches. The plotter can receive commands and transmit responses, but has no protection for input buffer overflow.

coordinate
The x-axis and y-axis values used to describe a point.

DAB command
(GPIB) A command consisting of data bytes from a controller that contains the command code and arguments.

DC1/DC3 Flagging
ASCII characters sent from the plotter to instruct the host when to start and stop sending data to the plotter.

device status
See *status information*.

digitizing
The process of locating and transmitting the pen's current coordinate information to the host.

Draw
The plotter draws a line to a specified position (a pen carriage movement with the pen down).

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 steady ERROR light and by sounding the plotter's bell.

escape sequence
(Serial) Same as an Attention Action command with the specific attention character of ^{E}c.

font
A set of one or more character definitions for print actions.

full duplex
This communication mode allows communication in two directions at the same time.

GDUs
Graphic Device Units. The default plotter units when using the GPIB Interface.

GIN
Graphic Input. Also called *digitizing.*

GIN terminator
See *output terminator*

GPIB
Abbreviation for General Purpose Interface Bus. Used in reference to interface or interface-dependent characteristics.

Graph mode
The plotter's mode after receiving an ASCII ^{G}s character. Causes the plotter to interpret subsequent ASCII characters as Move and Draw coordinates.

home position
A point located one character height below the upper-left corner of the page.

input buffer
Refers to the plotter memory that permits incoming commands to be temporarily stored until the plotter can process them.

integer argument
Consists of only the digits 0 through 9, which follow the command character with no spaces.

listener
(GPIB) A device that receives commands being transmitted over the GPIB interface.

loop-through
(Serial) A scheme where the plotter is inserted into the normal communication path between two pieces of equipment, such as between a computer or modem and a computer terminal.

media
The paper, mylar, etc., on which the 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; this use is unique to the 4050 Series Graphic System.

MTA (My Talk Address)
(GPIB) The device address that enables the plotter to be talker.

numeric argument
A numeric argument expressed in one of three standard forms: integer, floating point, or scientific notation (E format).

Optimized Coordinate Encoding
The process in which it is only necessary to transmit the portion of the new Move or Draw coordinate that changes from the previous Move or Draw.

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 or characters sent following an output message transmitted by the plotter. Also referred to as GIN terminator.

padding
(Serial) The process of intentionally limiting the average rate of information transfer by inserting extra characters that will be ignored.

page
A rectangular area on the platen that can be used for plotting.

pen carriage
The part of the plotter that includes the pen holder.

platen
The flat surface on which the media is placed.

plot
Refers to the composite line and character information drawn on the media.

plotter coordinate system
A Cartesian coordinate system for the page. The origin is the lower-left page corner. (See *plotter units* for additional details.)

plotter units
The numerical values assigned to the page for the plotter coordinate system. The default selection is ADUs for the Serial Interface and GDUs for the GPIB Interface. The range of values is from 0 to 4095 ADUs (x-axis) and from 0 to 2730 ADUs (or 3124 in Copy mode for the y-axis). For GDUs, the range of values is from 0 to 150 GDUs (x-axis) and from 0 to 100 GDUs (y-axis).

primary address
(GPIB) A number assigned to each device on the GPIB bus. The plotter's primary address is selected by a setting on the four rear-panel switches.

prompt character
(Serial) An ASCII character 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 ^{E}c character for a command terminates the previous command).

response block
A grouping of information transmitted from the plotter to the host.

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 Serial 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 that the plotter can send on request to describe its current state.

Style I commands
Plotter commands that use ASCII decimal coding.

Style II comands
Plotter commands that use binary data coding. Used with TEKTRONIX 4010 Series Computer Display 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 devices.

terminator
See *command terminator.*

transmit baud rate
(Serial) The rate at which information is transmitted by the plotter to the host.

turnaround delay
A function that delays plotter transmissions for a set time after host/controller information has been received by the plotter.

INDEX