PLOTTER NOTES
NUMBER 6

INTERFACING AND HANDSHAKING GUIDE

What's Inside This Document

The procedure required to interface a peripheral device to a computer is a multistep process. This application note thoroughly explains the underlying concepts as well as the instructions used in this process and supplements the material contained in the operating and programming manuals for HP's EIA* RS-232-C/CCITT V.24 plotters. This note emphasizes how to determine and set up communication and handshake protocol.

Who Should Read This and How

This document should prove useful to new owners of HP plotters as well as more experienced users. Depending on your knowledge of handshaking protocol, you may want to read this note from start to finish, or read only selected sections. Use the notes and graphics in the margins to find information quickly or to decide if you need to read that section. Because there are slight differences in the controls and instruction sets for each HP plotter, the appropriate manual for your plotter should be used in conjunction with this application note when interfacing your plotter.

NOTE: All information in this plotter note applies equally to RS-232-C and CCITT V.24 interfaces except for the pin assignments as noted in the appendix. For simplicity, the term RS-232-C has been used to refer to both interfaces.■

AN OVERVIEW OF INTERFACING

Before connecting one of HP's serial RS-232-C graphics devices to a computer, it is helpful to understand what an interface does and why it is necessary. Ideally, all graphics peripherals and computers would conform to some standard that specified all the characteristics of their input/output connections, making all such devices plug-to-plug compatible. Unfortunately, at the present time, no such standard exists. In the case of the serial RS-232-C interface, the EIA standards address only the mechanical and electrical characteristics and the functions of signals; other characteristics are not addressed. As a result, compliance with the RS-232-C standard is not a guarantee of compatibility between two devices. In actuality, there are four areas of compatibility that must be satisfied in order to successfully interface a graphics device and a computer. They are:

- Mechanical compatibility,
- Electrical compatibility,
- Data compatibility, and
- Communication compatibility.

HEWLETT PACKARD

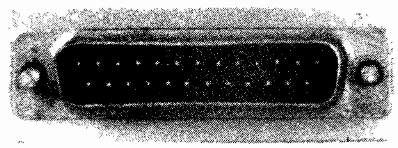
*Electronic Industries Association

Let's take a brief look at the first three areas before dealing in depth with communication compatibility.

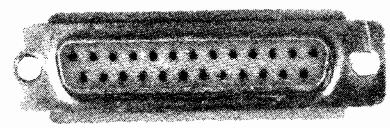
Mechanical Compatibility

RS-232-C mechanical compatibility implies the use of a standard 25-pin connector. The EIA standard defines 25 wires and their respective signal functions; these are the wires that may be used to interface a peripheral device and a computer or modem and, in some cases, a data terminal. On HP plotters a maximum of nine lines have been connected to the plotter's internal circuitry. Refer to your interfacing and programming manual and the appendix of this note for information on the signal designation and direction.

RS-232-C Connectors



Male Connector



Female Connector

Electrical Compatibility

Data is passed between devices over the data lines using two voltage levels to represent the two possible states (1 or 0) of a binary digit or bit. The voltage levels must be compatible for the two devices. RS-232-C standards specify that voltage levels between +3 V and +25 V on data lines be recognized as 0 or "space" and on control lines as ON. Voltage levels between −3 V and −25 V must be recognized as 1 or "mark" for data lines and as OFF for control lines.

Data Compatibility

Once an interface has made the computer and its graphics device mechanically and electrically compatible, they are capable of exchanging messages in the form of electrical signals. But in order for these messages to be understood and executed, certain conventions must be followed regarding the formatting of the data to be exchanged. For internal communication, devices may use any data format. However, each will input and output data in a standard character representation such as EBCDIC or ASCII*.

All HP plotters utilize the ASCII code; as a result, they are compatible with a wide variety of devices. There are two important characteristics to be considered in determining compatibility when transferring data between the computer and plotter — the number of bits per character and the format of those bits. ASCII characters are coded in seven bits, with an eighth bit to be used as a parity, or error-checking, bit. While the parity bit may not be active, it still must be included as the eighth bit of each character. This code allows 128 unique bit patterns for character representation.

The following table shows the binary code for the uppercase characters A through C and the decimal equivalent of their seven low-order bits. The lowercase p in the binary code represents the required eighth bit, the parity bit which may have the value 0 or 1. You will use decimal equivalents of ASCII characters to specify some parameters of the plotter commands which establish communication between the plotter and the computer. A table listing the decimal equivalents of ASCII characters can be found in the appendix.

*EBCDIC = Extended Binary Coded Decimal Interchange Code
ASCII = American Standard Code for Information Interchange

ASCII Character	Binary Code	Decimal Equivalent of Low-Order Seven Bits
A	p1000001	65
B	p1000010	66
C	p1000011	67

Asynchronous Transmission and Stop Bits

Data is transmitted asynchronously across the signal lines by means of these eight-bit characters. Asynchronous transmission means the data transfer may be initiated at any time, and each character sent need not have any time relationship to any other character. In order for the receiving device to distinguish the starting and ending point of each character, each eight-bit group must be preceded by a "start" bit and followed by one or two "stop" bits. In most cases, plotters verify and generate a single stop bit. Two stop bits are generated and verified by HP plotters if their baud rate setting is less than or equal to 110. Switches or jumpers can be used on some plotters so that two stop bits are also generated at settings above 110.

COMMUNICATION COMPATIBILITY

Overview

Both computers and their peripheral devices have a wide range of operating speeds. The graphics peripherals are typically slower in their ability to execute instructions than computers are in their ability to generate them, necessitating the design of a system to ensure the efficient transmission and receipt of data without loss. Providing this communication compatibility is the fourth and final step required to interface your plotter with your computer.

There are four factors to consider in establishing communication compatibility between two devices — baud rate, input buffer characteristics, communication protocol, and handshake protocol.

Baud Rate

The first timing requirement is to match the data transmission speeds of the interfacing devices. This is done by setting the baud or transmission rates of the computer and plotter equal. The baud rate is approximately equal to the number of bits transmitted per second. This can be translated into an approximate number of characters per second, since each character is 10 bits long (eight-bit code plus one start and one stop bit). At 300 baud, a maximum of 30 characters per second may be transmitted. It is crucial that the device receiving data be prepared to interpret the eight-bit characters at the same rate at which they are sent. Failure to do so will result in garbled data. The baud rate selection dial or switches are located on the plotter's rear panel.

Baud Rates for HP Plotters

HP RS-232-C plotters receive and transmit data at the standard baud rates shown in the following table.

Model	Switch-Selectable Baud Rates
7220, 7221, 7240	75, 110, 150, 200, 300, 600, 1200, 2400
7470	*75, 110, 150, 200, 300, 600, 1200, 2400, 4800, 9600
7580/85A	75, 110, 150, 300, 600, 1200, 2400, 4800, 9600
7580/85B	50, 75, 110, 134.5, 150, 200, 300, 600, 1200, 1800, 2400, 3600, 4800, 7200, 9600

*The 7470 can be operated at intermediate baud rates up to 9600 baud if an external clock with a frequency of 16 times the desired baud rate is connected to pin 17 (DD) of the plotter.

Input Buffer

All HP plotters with an RS-232-C interface use an input buffer to store data until the plotter can process it. Because the plotter has an input buffer, the computer can send large blocks of data at high speeds, a more efficient method than sending a few characters at a time. Moreover, the buffer uncouples the plotter's data execution rate from the data transmission rate of the terminal or computer.

What Gets Buffered

All plotter instructions except device control instructions are placed in the buffer on a first-in, first-execute basis; device control instructions, which begin with the escape character followed by the character "." are not placed in the buffer but instead are processed immediately. Plotting instructions from the computer are held in the buffer, if necessary, until the plotter is free to process and execute them. Buffer sizes for HP RS-232-C plotters are shown below.

Buffer Size

Model	Buffer Size
7220C/T	928 bytes, additional 2048 byte RAM optional
7221C/T	1110 bytes, additional 2048 byte RAM optional
7240	1236 bytes
7470	255 bytes
7580/85	1024 bytes

An Introduction to Communication Protocol

Every computer requires that information be sent to it in a particular manner. For instance, the computer may need to be told when data is coming and when the end of data has been reached. In order to determine how information must be sent to your computer, you need to study its "communication protocol." Communication protocol defines the conventions by which handshaking and other output operations will take place. In other words, the protocol establishes a specific format for communication.

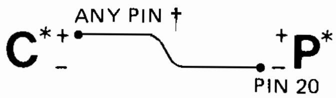
Instructions Used

There are six parameters which you may specify so that data sent from the plotter will be formatted in such a way that it can be understood by the computer. These parameters are specified in two device control instructions: the set output mode instruction, ESC . M, and the set extended output and handshake mode instruction, ESC . N. These instructions and their parameters will be discussed in detail later, in the section titled Matching Your Computer's Communication Protocol.

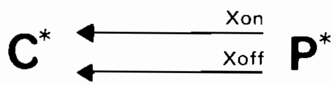
An Introduction to Handshaking Protocol

In order to prevent loss of data due to buffer overflow, the computer and plotter must communicate about the availability of buffer space in the plotter. This process is called "handshaking." HP plotters have the ability to implement several different handshakes. Therefore, they can interface with many types of equipment from different manufacturers, resulting in greater flexibility in the assembling of computer graphics systems. An efficient handshake will optimize computer/peripheral communication while preventing data loss. There are four basic types of handshakes that can be used by HP graphics devices:

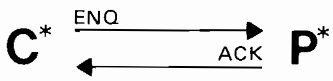
The Four Handshakes



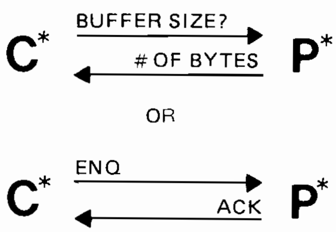
- **Hardwire handshake** — uses a physical wire, pin 20 of the RS-232-C connector, to control handshaking. It can be used if the computer and the plotter are directly connected, without a modem, and the computer system can or does monitor pin 20 (CD, Data Terminal Ready).



- **Xon-Xoff handshake** — is initiated by the peripheral device, e.g., a plotter. It can be used if the computer system supports an Xon-Xoff protocol, which means that control characters are sent from the plotter to the computer to indicate when the plotter's buffer is full and when it again has room for more data.



- **Enquire/acknowledge handshake** — is initiated by the computer operating system in specific cases where it uses the two ASCII characters ENQ and ACK as the enquiry character and acknowledgment string.



- **Software checking handshake** — is managed by the applications programmer. It can be used on almost any computer system, but it must be used if the system cannot implement any of the above handshaking methods. One type of software handshake repeatedly asks how many bytes are available in the buffer, and sends the data when the space available is larger than the block to be sent. This method can degrade system performance in a multi-user environment because it increases the load on the I/O system. Another type of software handshake is a "generic" enquire/acknowledge handshake which asks the plotter if there is room for a block of data; the computer sends that block only after receiving an acknowledgment. Any character(s), not just ENQ and ACK, may be specified as the enquiry character and the acknowledgment string.

MATCHING YOUR COMPUTER'S COMMUNICATION PROTOCOL

Overview

Communication protocol is independent of the handshake chosen. Its primary purpose is to assure data is sent from the plotter in a form that the computer can understand. This data can be general plotter output, or data that is part of a particular handshake.

Remember that communication protocol is determined by your computer. In order to detemine the proper communication conventions, you will need to study your computer documentation. As you do, it may help to know that HP plotters are designed to communicate with a computer as if they were simple teletype terminals. That is, any plotter responses are formatted as though a person typed them on a terminal.

The two instructions used to establish communication protocol are ESC . M and ESC . N. This section contains a general description of the six parameters which may be specified in these instructions. The syntax of each instruction is then discussed, followed by some hints on establishing the parameter values required by your computer system.

*C = computer
P = plotter
†any pin that the computer will monitor

Parameters for Plotter Output

Turnaround Delay

The turnaround delay is the length of time the plotter will wait after receiving a computer request and the output trigger character, if any, before it responds. The purpose of this time delay is to postpone the plotter's transmission of requested data until the computer is ready to receive and process it. Systems may require either a turnaround delay or a trigger character or both.

Output Trigger Character

The output trigger character, when used, is the last character output by the computer when making a request of the plotter. Defining this character tells the plotter, "Don't respond to my request until you receive this trigger character." This character is often a nonprinting character such as "DC1", or a printing character such as "?" or ">". On DEC systems, line feed is often used as the output trigger character.

Echo-Terminate Character

Echoing is commonly found in full-duplex systems. Use of the echo-terminate parameter tells the plotter that the computer will echo all responses, and that this echoed data should be ignored (the plotter's data buffer should be closed) until an echo-terminate character is received. When the plotter receives the echo-terminate character, it reopens the data buffer to receive graphics data from the computer. Computers often append the line feed character (decimal equivalent 10) to the plotter's response as the echo terminator. If the computer echoes the plotter's response without appending a character, then the plotter's output terminator (which is either a carriage return or a different character which you have specified) should be used as the echo terminator. If the computer does not echo the peripheral's response, this parameter must be specified as zero (equivalent to null) or must be omitted.

Output Terminator

The output terminator is a one- or two-character terminator that the computer requires the plotter to send at the end of each response to a data request. The output terminator tells the computer, "This completes my transmission." Often computers expect the carriage return character (decimal equivalent 13) as the plotter's output terminator. The plotter sends a carriage return as the output terminator unless a different character has been specified by an ESC . M command.

Output Initiator Character

The output initiator character can only be specified for the 7470 and 7580/85 plotters. It is a one-character initiator that is sent by the plotter at the beginning of a string. The output initiator tells the computer, "This starts my transmission." Some computers expect the start-of-text character (decimal equivalent 2) as the plotter's output initiator.

Intercharacter Delay

Some computers cannot process data as fast as the plotter can transmit it, especially at high baud rates, due to limited buffering in the I/O port. This can be compensated for by delaying each transmission from the plotter a period of time as specified by the intercharacter delay parameter. This intercharacter delay is added to the turnaround delay (if one has been specified) before the first character is sent by the plotter, and is also inserted before each subsequent character in a string being sent to the computer.

Instructions Used to Set Communication Protocol

Syntax Conventions

Throughout this document, certain syntax conventions are used when discussing device control instructions. Each parameter must be separated by a semicolon ;. A colon : terminates the command. To set the default value for a specific parameter, do not specify the value in that position, but include the semicolon (i.e., ;;). Parameters are either **integers** or **integer decimal equivalents** of ASCII characters, as noted with each instruction.

ESC . M

There are two device control instructions used to establish communication protocol. The set output mode instruction, ESC . M, is used to inform the plotter what turnaround delay, output trigger character, echo-terminate character, output terminator, and, on the 7470 and 7580/85, output initiator to use. This instruction has the following form:

ESC[1] . M turnaround delay[2]; output trigger character[3]; echo-terminate character[3]; output terminator[4]; output initiator (7470 and 7580/85 only)[3]:

The instruction which specifies a 100-millisecond turnaround delay, DC1 as the trigger character, line feed as the echo terminator, and carriage return as the output terminator is:

ESC . M 100 ; 17 ; 10 ; 13 :

The final parameter, 13, could have been omitted by placing the colon directly after the 10 and the results would be the same, because carriage return is the default output terminator sent by the plotter unless an ESC . M command specifies a different character or characters. Note that for the last three parameters, the **decimal equivalent** of the ASCII character is used to specify a parameter.

ESC . N

The set extended output and handshake mode instruction, ESC . N, is used to establish any intercharacter delay. This command has the following form:

ESC . N intercharacter delay[2]; Xoff trigger character(s) or immediate response string[5]:

When a hardwire handshake or a software checking handshake is being established, the intercharacter delay is the only parameter of the ESC . N command. For Xon-Xoff handshakes and sometimes for enquire/acknowledge handshakes, additional parameters will follow the intercharacter delay. Complete information on these parameters is found in the descriptions of the corresponding handshakes.

The following ESC . N command will set up a 10-millisecond intercharacter delay when a hardwire handshake or software checking handshake is in effect:

ESC . N 10 :

Programming Hint

Further examples of these instructions in conjunction with different handshakes appear later in this document and in the individual plotter manuals. Refer to your plotter manual for information on parameter ranges, default values, and omitting parameters.

When ESC . M and/or ESC . N are used with handshakes, it is good programming practice to place them ahead of the instructions that establish the handshake. When the ESC . H instruction (discussed later under the enquire/acknowledge handshake) is used to establish a handshake, any ESC . M or ESC .N instruction must be executed before an enquiry character is sent to a plotter.

[1]Text in reverse type represents a single ASCII character.

[2]An integer, base 10, without a decimal point.

[3]The integer decimal equivalent of an ASCII character.

[4]The integer decimal equivalent(s) of one or two ASCII characters, separated by a semicolon.

[5]The integer decimal equivalent(s) of a string of from 1 to 10 ASCII characters, separated by semicolons.

Determining Parameter Values to Match Your System

Common Values for Communication Protocol Parameters

The following table is intended as a quick reference of typical values for communication protocol parameters. For more details, refer to the definitions and syntax listed previously. Also, a complete table of ASCII codes and their decimal equivalents can be found in the appendix.

Remember that your computer system may require different values from those shown in this table. If you are not sure what values your system needs for output trigger, echo-terminate, and output terminator characters, refer to the FORTRAN test program in the appendix. This program should help you identify these parameters.

Device Control Instruction	Parameter	Typical Values	
		Decimal Equivalent	**ASCII Character**
ESC . M	Turnaround delay	any number of milliseconds	not applicable
	Output trigger character	17	DC1
	Echo-terminate character	10	LF
	Output terminator	13	CR
	Output initiator character (HP 7470 and 7580/85 only)	2	STX
ESC . N	Intercharacter delay	any number of milliseconds	not applicable

HANDSHAKING: A DETAILED DESCRIPTION

Who Should Read What

The remainder of this note discusses the four handshakes implemented on Hewlett-Packard plotters. The next section contains guidelines for selecting a handshake. Skip that section if you already know which type of handshake you will use. Each of the four sections about a particular handshake includes definitions of the terms associated with that handshake, an explanation of the instructions used, and at least one example. You may want to read only those sections about handshakes which you are considering for your system or application. To help you identify pertinent paragraphs, symbols for the various handshakes have been placed in the left margin of paragraphs concerning a particular handshake. The four symbols are:

for hardwire handshake

for Xon-Xoff handshake

for enquire/acknowledge handshake

for software checking handshake

Guidelines for Selecting a Handshake Method

But how does one know which handshake is best for a specific installation? The ideal handshake minimizes I/O transactions while keeping the plotter's data buffer supplied with data. The communication characteristics and capabilities of the host computer dictate which of the four handshaking methods is possible and will be most efficient. Device control instructions from the plotter's instruction set are used to tell the plotter which handshake is to be used so that the plotter conforms to the computer's requirements. **A thorough understanding of the computer's communication characteristics is necessary in order to make the best decision on the type of handshake.**

If you do not already know which is the best handshake to use, you should first consult your systems manual or the installation manual for your graphics software. Most graphics software designed for use with Hewlett-Packard RS-232-C plotters contains the instructions necessary to set up a handshake. You may need to fill in parameters suitable to your system which will be used in subroutine calls. The information you need to do this is found in the installation guides for the software. If you are not using a commercial software package, your system manual may tell you whether your system supports Xon-Xoff protocol, or hardwire handshake. If no recommendations can be found in your documentation, the following information should help you choose an efficient handshake.

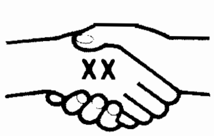
Plotters Hardwired to the Computer

The operating environment of your computer affects the number of choices you have; your plotter operates in either a hardwire or remote environment. In a hardwire environment, there is no intermediate hardware between the plotter and computer; a cable goes directly from plotter to computer. This is the only environment in which the hardwire handshake can be used. It is a very efficient handshake. Sometimes the term automatic handshake is used in computer manuals for this type of handshake.

Plotters in a Remote Environment

In a remote environment, the plotter is connected to the computer via a modem. If your plotter communicates with the computer via a modem, you cannot use a hardwire handshake and must choose between an Xon-Xoff, an enquire/acknowledge, or a software checking handshake. The Xon-Xoff handshake and some enquire/acknowledge handshakes are implemented in the operating system or device driver. The other handshakes are implemented in the graphics program.

Of the two types of software checking handshakes described briefly earlier in this note, the second method, which repeatedly checks the buffer space available, is the least efficient handshake, but it has one advantage. It works on any computer that supports two-way communication, and requires no knowledge of your operating system or computer's characteristics. It may be the one you want to use at least temporarily to get a program running. This handshake, which uses the ESC . B instruction, is described later in this document and in your plotter manual.



If efficiency is important, either because you are in a multi-user environment or want to minimize I/O transactions, and you cannot use a hardwire handshake, you will want to use either Xon-Xoff or the enquire/acknowledge type of handshake. If your system supports Xon-Xoff protocol, you should probably use that handshake. The handshake is efficient when sending either variable- or fixed-length records. Refer to the section on Xon-Xoff handshaking later in this document to learn which parameters and instructions are necessary, and then refer to your plotter manual for a detailed description of the parameters. A list of computers and operating systems on which the Xon-Xoff handshake has been successfully used is found in the appendix.

A Test for Xon-Xoff Capability

You may easily test your system for Xon-Xoff support by printing a long program listing or other text to a terminal plugged into the RS-232-C connector you plan to attach to your plotter. While information is printing on your display, send the ASCII character DC3 (Xoff) to the computer. On many terminals, this is done by pressing and holding the terminal's CONTROL key (sometimes labeled "CNTL" or "CTRL") and then pressing the "S" key. If printing immediately stops, send the character DC1 (Xon), often sent by pressing and holding the CONTROL key and then pressing the "Q" key. The printing should resume. If you can stop and start printing in this manner, you can probably use the Xon-Xoff handshake.

The enquire/acknowledge handshake can be implemented in either the application program or the device driver. In either case, one of two mutually exclusive instructions, ESC . H or ESC . I, must be sent to the plotter to specify the enquiry and acknowledgment characters. Refer to the detailed description of the enquire/acknowledge handshake for information on these two instructions and which one to choose. The enquire/acknowledge handshake is quite efficient and can be implemented on any system, provided the communication needs of that system have been taken into account. Since the block size parameter must be set to accommodate the largest record, use with variable-length records where there is a wide range of record lengths is somewhat inefficient.

Hardwire Handshake

As the name implies, the hardwire handshake takes place in the hardware rather than the firmware or software. The plotter controls the data exchange sequence by setting the electrical voltage on pin 20 of the connector (CD line) to the computer in order to signal the computer when to send another block of data. If there is enough room in the plotter's buffer to store another block of data, the plotter sets the Data Terminal Ready (CD) line to a high state. If there is insufficient space, it sets the line low. By monitoring this line, the computer knows when it can or cannot safely transmit another block of data.

Set Plotter Configuration, ESC . @

The hardwire handshake can be implemented on all Hewlett-Packard RS-232-C plotters except the 7221A, and the 7470 when the 07470-60090 eavesdrop cable is being used. This is done by setting the Data Terminal Ready line control using the set plotter configuration command, ESC . @. Since there are slight differences in the implementation of the ESC . @ command on each plotter, refer to your plotter's documentation when using this command.

Switch-Selectable

In some cases, an ESC . @ command need not be sent to enable hardwire handshake. On the HP 7470, hardwire handshake is enabled at power-on. Also, on plotters with a MODEM/HARDWIRE switch, hardwire handshake is enabled at power-on when the switch is in the hardwire position.

Block Size

Between the time that the plotter signals insufficient space by setting the Data Terminal Ready (CD) line low, and the time that the computer senses this, some data could be sent to the plotter and be lost. You can avoid data loss by specifying a block size that is slightly larger than the maximum amount of data that could be sent before the computer would sense that the CD line is low. This block-size parameter is closely tied to the output characteristics of the computer and is frequently set equal to the line length of the system's printer. The block size is set by the first parameter of an ESC . H or ESC . I command. If you do not establish a block size with one of these commands, the plotter will assume a block size of 80 bytes.

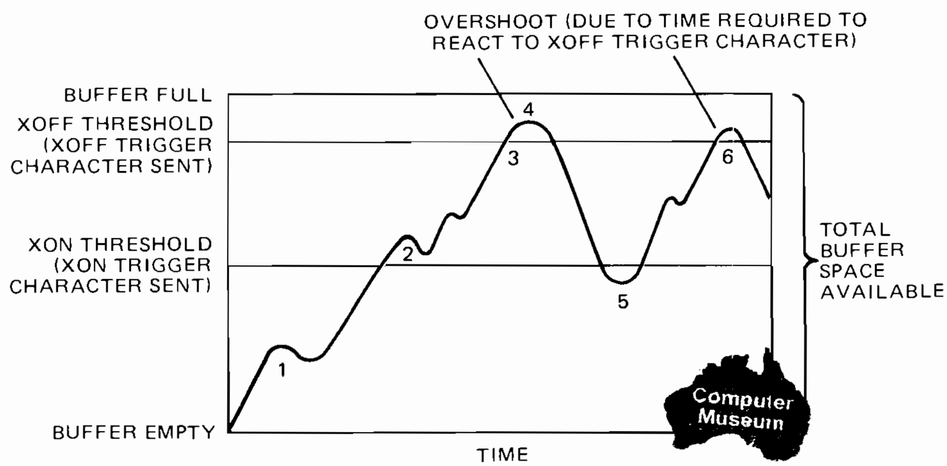
Xon-Xoff Handshake

The Xon-Xoff handshake is becoming more common and is the preferred handshake method for many minicomputer systems and some larger systems. Conceptually, it is easy to understand and relatively transparent to the user. It is probably the most efficient handshake that can be used in a remote environment. The Xon-Xoff handshake is handled by the operating system (firmware) of the computer, rather than by application programs (software).

With the Xon-Xoff handshake, the plotter controls the data exchange sequence by telling the computer when it has room in its buffer for data and when to shut off the flow of data. The plotter uses buffer threshold indicators, an Xon trigger character, and an Xoff trigger character to prevent buffer overflow by sending the Xoff trigger character when the Xoff threshold is reached. These parameters are discussed following the next brief explanation of how an Xon-Xoff handshake works.

A Diagram of Buffer Level

The following diagram is representative of the way the Xon-Xoff handshake works. The stages represented by the six numbers are described following the diagram.



Xon-Xoff Threshold Levels

1. Data enters the buffer faster than it can be acted on by the plotter, and the buffer starts to fill.

2. The plotter begins processing data faster than the computer sends it, and the buffer starts to empty.

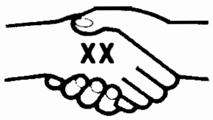
3. The data again enters the buffer at a faster rate than the plotter can process it. The amount of data stored in the buffer reaches the Xoff threshold level, at which point the plotter sends the Xoff trigger character, stopping the flow of data from the computer.

4. Due to a finite delay between the time the plotter sends the Xoff trigger character and the time it takes the computer to react, a slight overshoot may occur and extra characters may be sent.

5. After the Xoff trigger character has been sent and the amount of data stored in the buffer drops to the Xon threshold level, the plotter sends the Xon trigger character to signal the computer to resume sending data. The Xon threshold level is automatically set at one-half the buffer size. If the Xoff threshold is greater than

one-half the buffer size, the Xon threshold is reset to send the Xon character when one more byte than required by the Xoff threshold is available in the plotter's buffer.

6. Data is again stored in the buffer until all the data are transferred or until the Xoff threshold level is exceeded again.



Parameters Used in an Xon-Xoff Handshake

The following parameters are used to establish an Xon-Xoff handshake.

Xoff Threshold Level

The Xoff threshold level is the number of empty bytes remaining in the buffer when the plotter sends the Xoff trigger character to the computer, telling it to stop sending data. On most systems and with most plotters, 10 is a suitable value. The Xoff threshold level is specified by the first parameter in an ESC . I command.

Xon Trigger Character

The Xon trigger character is the character the plotter will use to signal the computer that there is sufficient space in the buffer to resume sending data. By convention, the DC1 character (decimal equivalent 17) is the Xon trigger character. It is specified by the third parameter of the ESC . I command, the second parameter having been omitted by entering only a semicolon.

Xoff Trigger Character

The Xoff trigger character is sent by the plotter to signal the computer to temporarily stop sending data while the plotter processes what it has already received. By convention, the DC3 character (decimal equivalent 19) is the Xoff trigger character. It is specified by the second parameter of the ESC . N command.

Common Values for Xon-Xoff Handshake Parameters

The following table is intended as a quick reference of typical values for Xon-Xoff handshake parameters. For more details, refer to the above definitions and the syntax listed below. Also, a complete table of ASCII codes and their decimal equivalents can be found in the appendix. *Remember that your computer system may require different values from those shown in this table.*

Device Control Instruction	Parameter	Typical Values	
		Decimal Equivalent	**ASCII Character**
ESC . I	Xoff threshold level	10	not applicable
ESC . I	Xon trigger character	17	DC1
ESC . N	Xoff trigger character	19	DC3

Instructions Used to Establish the Xon-Xoff Handshake

Two commands are necessary to establish an Xon-Xoff handshake, the set extended output and handshake mode instruction, ESC . N, and the set handshake mode 2 instruction, ESC . I.

The format of the set extended output and handshake mode instruction, ESC . N is:

ESC . N

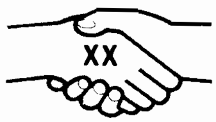
ESC[1] . N intercharacter delay[2] ; Xoff trigger character(s)[3] :

[1]Text in reverse type represents a single ASCII character.
[2]An integer, base 10, without a decimal point.
[3]The integer decimal equivalent(s) of a string of from 1 to 10 ASCII characters, separated by semicolons.

The format of the set handshake mode 2 instruction is:

ESC . I

ESC . I Xoff threshold level[1] ; omitted parameter ; Xon trigger character(s)[2] :



An Example of an Xon-Xoff Handshake

The following commands will establish an Xon-Xoff handshake.

ESC . N;19:

ESC . I 10;;17:

These commands will cause the Xoff character to be sent when there are less than 10 empty bytes in the plotter's buffer. Note that two semicolons follow this parameter. To implement the Xon-Xoff handshake, the second parameter must be omitted by including only the semicolon, or specifying it as 0. Except for a variation in the Xoff threshold level, the only likely variation in the two commands shown above might be the addition of a parameter specifying an intercharacter delay before the semicolon in the ESC . N command. The size of this delay would depend upon the system requirements and perhaps the baud rate. A system which supports an Xon-Xoff handshake may or may not require use of an ESC . M command to set plotter output protocol to match the computer system.

Enquire/Acknowledge Handshake

With the enquire/acknowledge handshake, the computer's operating system or the application program initiates the data exchange process by querying the plotter about the availability of buffer space. This handshake method derives its name from the two characters, ENQ and ACK, used on some Hewlett-Packard systems to verify that a peripheral is ready to receive output.

An Enquire/Acknowledge Handshake in the Computer's Operating System

When the enquire/acknowledge handshake is controlled by the operating system using the characters ENQ and ACK (decimal equivalents of 5 and 6 respectively), it might be called a "true" enquire/acknowledge handshake. It involves no additional steps in the application program to check for buffer space; it takes advantage of the fact that the operating system will check before sending a block of data to assure there is room for that block in the peripheral.

An Enquire/Acknowledge Handshake in an Application Program

An enquire/acknowledge handshake can also be implemented in an application program, using ENQ and ACK or almost any other characters for the enquiry character and acknowledgment string. For the enquiry character, never use any character that will be sent to the plotter as part of a plot command or a label; this is because, once defined, the enquiry character will be stripped from any incoming data stream before the data is sent to the plotter's parser or buffer. Also, you may want to use a digit as the acknowledgment string because it is the easiest variable to read into a program since it can be successfully interpreted as a string or numeric variable.

An enquire/acknowledge handshake implemented in an application program is more time consuming than one implemented in the operating system because, as an extra step in an application program, it requires an extra write and read each time a block of data is to be sent. However, the handshake is far more efficient than the software checking handshake, which uses ESC . B to ask continually how much empty space remains in the buffer. An enquire/acknowledge handshake in an application program can be implemented on any system, provided the communication protocol of the

[1] An integer, base 10, without a decimal point.

[2] The integer decimal equivalent(s) of a string of from 1 to 10 ASCII characters, separated by semicolons.

system has been taken into account and the interface the plotter is connected to will receive as well as send data.

Plotter's Response to ENQ

With the HP 1000 and 3000 computers, a "dummy" ENQ/ACK handshake occurs when the plotter uses *another* handshake method. That is, the computer automatically sends an ENQ; the plotter then automatically sends an ACK. If the plotter did not send the "dummy" ACK response, the computer would not send data to the plotter. This "dummy" ACK does not indicate that buffer space is available; it is simply an expected response to the ENQ. This dummy handshake does not prevent you from using a different handshake method. Just be aware that this dummy handshake takes place. You can also use a true ENQ/ACK handshake, in which case the ACK **does** indicate that buffer space is available.

Parameters Used in Enquire/Acknowledge Handshakes

NOTE: In some plotter manuals, the "enquiry character" is called the "handshake enable character." Similarly, the "acknowledgment string" is called the "handshake response string." These terms are interchangeable. ■

The following parameters are used to establish an enquire/acknowledge handshake.

Enquiry Character

When initiating an enquire/acknowledge handshake, the computer sends an enquiry character to ask the plotter if it has room for a block of data. The character ENQ (decimal equivalent 5) is frequently used as the enquiry character, but other characters can be used. Printing characters or control characters which are used in plot programs, e.g., carriage return, line feed, shift-in and shift-out, should not be used as an enquiry character. It is specified by the second parameter in the ESC . H or ESC . I commands.

Acknowledgment String

The acknowledgment string specifies the character or characters that the plotter will send to the computer in response to an enquiry character when the plotter's input buffer has room for another block of data. The character ACK (decimal equivalent 6) is frequently used as the acknowledgment string, but other characters can be used. Avoid using characters which have special meaning to the computer. The acknowledgment string is specified by the third parameter in the ESC . H or ESC . I commands.

Immediate Response String

Certain systems require an immediate response from the plotter acknowledging the enquiry from the computer. Systems of this type include a computer that transmits data to the plotter after a certain time interval but before receiving a go-ahead signal from the plotter. If the plotter's buffer is full and the computer sends more data, the buffer will overflow. The immediate response string prevents this inadvertent transmission of data before the plotter is ready. The immediate response string is transmitted by the plotter immediately after the receipt of an enquiry character and tells the computer, "Wait, I am checking my buffer space." NAK (decimal equivalent 21) is sometimes used for this character. It is specified by the second parameter in an ESC . N command.

Data Block Size

This specifies the maximum size of any data block that the computer will transmit to the plotter. It is specified by the first parameter of an ESC . H or ESC . I command.

Other Parameters

You may also wish to review the definitions given in the section, Matching Your Plotter's Communication Protocol. Which one of the two instructions, ESC . H or ESC . I, is used to establish an enquire/acknowledge handshake depends on whether the computer system requires any output trigger character, output terminator, echo terminator, or output initiator in the handshake process. The immediate response string, turnaround delay, and intercharacter delay, if any, will always affect handshake responses no matter which instruction is used to establish the handshake.

Common Values for ENQ/ACK Handshake Parameters

The following table is intended as a quick reference of typical values for ENQ/ACK handshake parameters. For more details, refer to the above definitions and the syntax listed below. Also, a complete table of ASCII codes and their decimal equivalents can be found in the appendix. *Remember that your computer system may require different values from those shown in this table.*

Device Control Instruction	Parameter	Typical Values	
		Decimal Equivalent	ASCII Character
ESC . H or ESC . I*	Data block size	80	not applicable
	Enquiry character	5	ENQ
	Acknowledgment string	6	ACK
ESC . N	Immediate response string	21	NAK

*Refer to the table in the next section for hints on choosing the correct instruction.

Choosing the Proper Handshake Mode

To implement an effective enquire/acknowledge handshake, you must know whether to use the ESC . H or ESC . I instruction. **Only one instruction is correct for any system.** That instruction determines whether the communication protocol set up for use with responses to plotter output commands is used with the handshake enable character and the handshake response string.

Mode 1: ESC . H

The ESC . H command establishes what is called handshake mode 1. This mode is used when the computer requires that all parameters set in the ESC . M instruction be used in the response to the enquiry character. Generally, this mode is used when the handshake is controlled in application software.

Mode 2: ESC . I

The ESC . I command establishes what is called handshake mode 2. This mode is used when the computer expects only the turnaround delay and not the other parameters set by ESC . M to be included when the enquiry character and the acknowledgment strings are sent. Generally, this mode is used when the handshake is controlled by the computer's operating system.

Parameter Usage in Plotter/Computer Communication

The following chart shows which parameters affect output responses of handshake mode 1, handshake mode 2, and all plotter output commands.

ESC . M/ESC . N Parameters	With Handshake Characters In Mode 1 (ESC . H)	With Handshake Characters In Mode 2 (ESC . I)	With Plotter Output Commands
Turnaround Delay	Yes	Yes	Yes
Output Trigger Character	Yes	No	Yes
Echo Terminator	Yes	No	Yes
Output Terminator	Yes	No	Yes
Output Initiator*	No	No	Yes
Intercharacter Delay	Yes	Yes	Yes

*7470 and 7580/85 plotters only.

Instructions Used to Establish the Enquire/Acknowledge Handshake

The format of both the ESC . H and ESC . I instructions is identical and is given next:

ESC . H (or I) block size[1]; enquiry character[2]; acknowledgment string[3]:

In the next section you will find examples of both handshake mode 1, which uses ESC . H, and handshake mode 2, which uses ESC . I.

Examples of Enquire/Acknowledge Handshakes

Example 1: Mode 1

If the computer can communicate with a plotter without specifying any of the ESC . M/ESC . N parameters, you can establish the simplest form of an enquire/acknowledge handshake by sending the following command to the plotter:

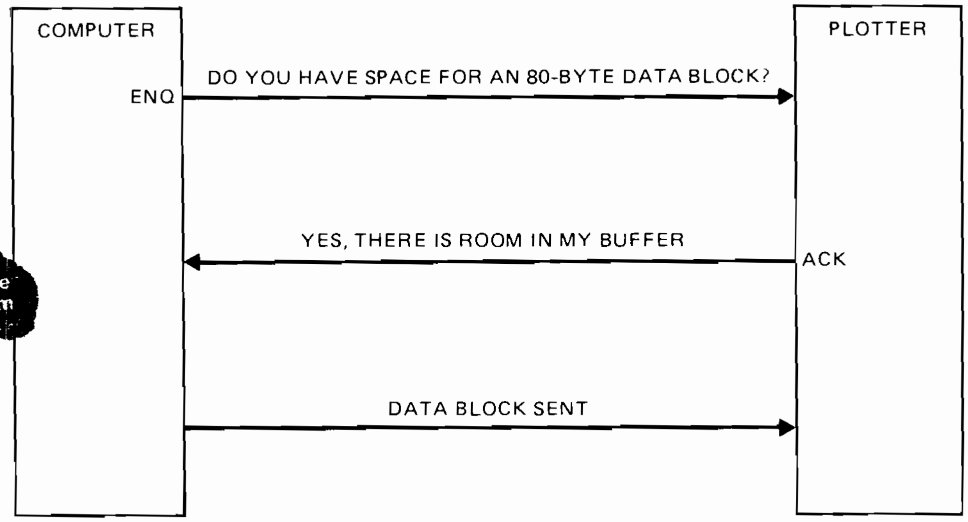
ESC . I;5;6:

When this handshake is implemented, block size defaults to 80 bytes because the first parameter is omitted by entering only the semicolon. The enquiry character is ENQ, which is specified by its decimal equivalent 5, and the acknowledgment string is ACK, which is specified by its decimal equivalent 6. The following diagram illustrates data exchange as it will occur when this handshake is implemented.

[1]An integer, base 10, without a decimal point.

[2]The integer decimal equivalent of an ASCII character.

[3]The integer decimal equivalent(s) of a string of from 1 to 10 ASCII characters, separated by semicolons.



Enquire/Acknowledge Handshake — Example 1

Example 2: Mode 1

In many systems, additional communication protocol will have been specified using the ESC . M and/or ESC . N commands. When this is true, it is imperative that the proper set handshake mode instruction, ESC . H or ESC . I, be used to establish an enquire/acknowledge handshake. Let's assume a system required the following:

no turnaround delay,

output trigger character, DC1 (decimal equivalent 17),

echo-terminate character, line feed (decimal equivalent 10),

output terminator, carriage return (decimal equivalent 13),

data block size, 80 bytes,

enquiry character, ENQ (decimal equivalent 5), and

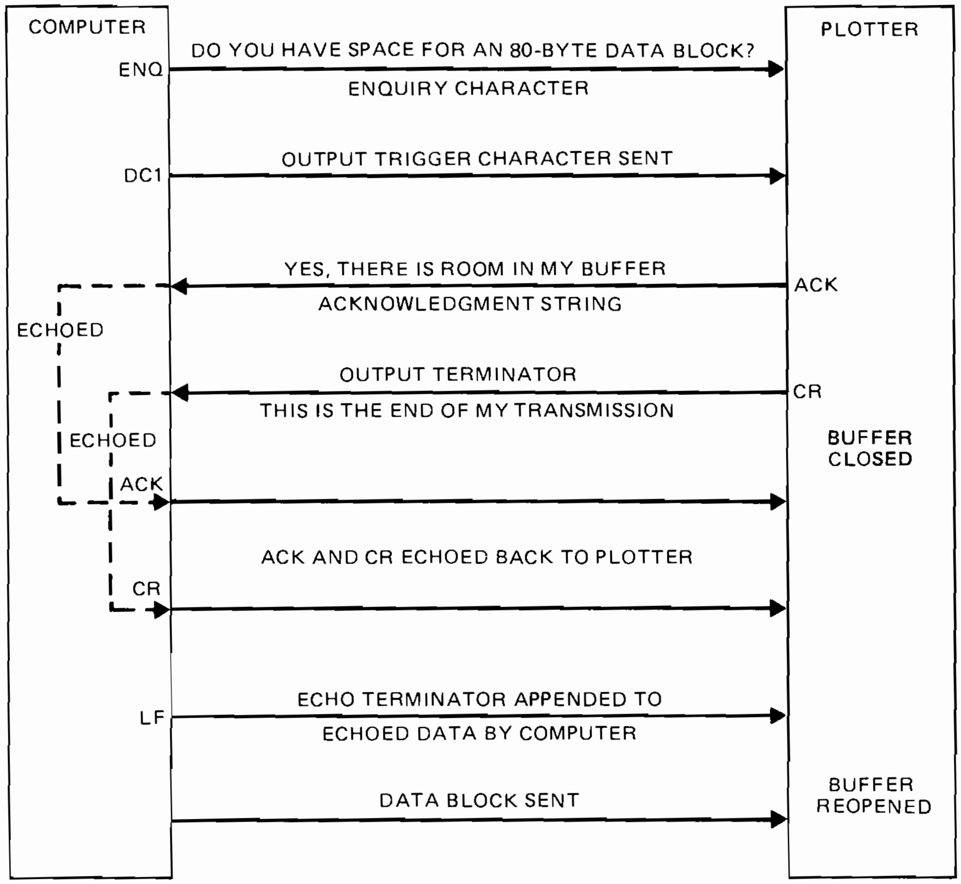
acknowledgment string, ACK (decimal equivalent 6).

If the ESC . I instruction were used, the handshake diagram would be identical to the previous diagram because the output trigger character, echo-terminate character, and output terminator would not be used with the enquiry character and acknowledgment string. (Refer to the previous table on parameter usage.)

If the ESC . H instruction were used, the data exchange which would result when the proper ESC . H and ESC . M commands, listed below, were sent to the plotter is shown next.

ESC . M;17;10;13:

ESC . H80;5;6:



Enquire/Acknowledge Handshake — Example 2

Example 3: Mode 1 or 2

The next diagram shows the data exchange if only an ESC . N command were used to set the following conditions:

intercharacter delay, 50 milliseconds, and

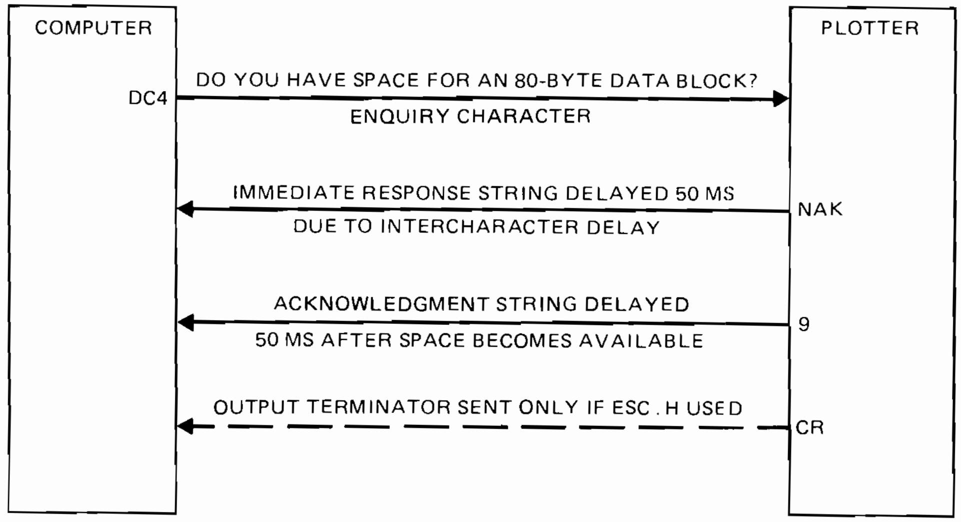
immediate response string, NAK (decimal equivalent 21).

Similar diagrams would result for handshake mode 1, established by ESC . H, and handshake mode 2, established by ESC . I. The only difference is that ESC . H uses the output terminator, which is set by the plotter to be a carriage return (decimal equivalent 13) at power on. The ESC . M instruction does not need to be sent unless you want to set a different output terminator.

In the commands, we have changed the enquiry character and acknowledgment string to "DC4" and "9" instead of ENQ and ACK to illustrate that characters other than ENQ and ACK are often used for enquire/acknowledge handshakes in graphics programs. Some computers may not be able to read a nonprinting control character sent by the plotter; because a digit can be read into a character or numeric variable, a digit is often a good choice for an acknowledgment string.

ESC . N 50;21:

ESC . H (or I);20;57:



Enquire/Acknowledge Handshake — Example 3

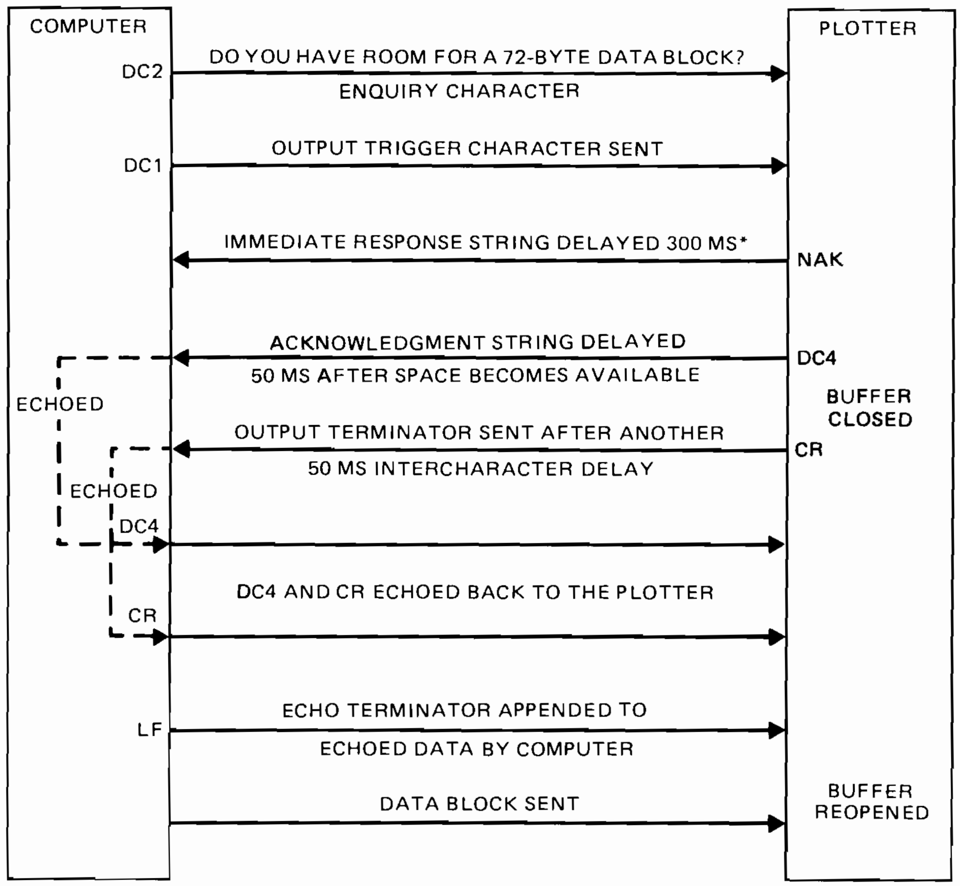
Example 4: Mode 1

The final example shows an enquire/acknowledge handshake when both the ESC . M and ESC . N commands are used with the ESC . H command. This would be necessary if the computer system required either an intercharacter delay or immediate response string and one of these parameters: turnaround delay, output trigger character, echo terminator, or an output terminator other than carriage return. Again we have used characters other than ENQ and ACK for the enquiry character and acknowledgment string. The instructions and a diagram of the data exchange are shown next.

ESC . M 250 ; 17 ; 10 ; 13 :

ESC . N 50 ; 21 :

ESC . H 72 ; 18 ; 20 :



*300 MS DELAY IS TOTAL OF 250 MS TURNAROUND DELAY AND 50 MS INTERCHARACTER DELAY

Enquire/Acknowledge Handshake — Example 4

Software Checking Handshake

Two Methods

When your computer system does not implement one of the previously described handshakes in its hardware or operating system, you must add to your application program steps to assure that data is transferred from computer to plotter without loss. In the previous section, we have talked about one type of software handshake using the enquire/acknowledge principle (mode 1: ESC . H). This is the preferred and more efficient of the two classes of software handshake.

The second type of software handshake uses the plotter command, ESC . B, which causes the output of the number of empty bytes in the buffer. So, in order to establish this handshake, communication protocol must be specified to match the requirements of the computer system. You may want to review the section, Matching Your System's Communication Protocol. The following variables may be specified for the software handshake, using the appropriate command:

turnaround delay (ESC . M command)

output trigger character (ESC . M command),

echo-terminate character (ESC . M command),

output initiator (ESC . M command),

output terminator (ESC . M command), and

intercharacter delay (ESC . N command).

You should remember that long graphics programs utilizing the ESC . B command to implement software handshaking tend to degrade system performance. Where system performance is important, one of the other handshaking methods should be used whenever possible.

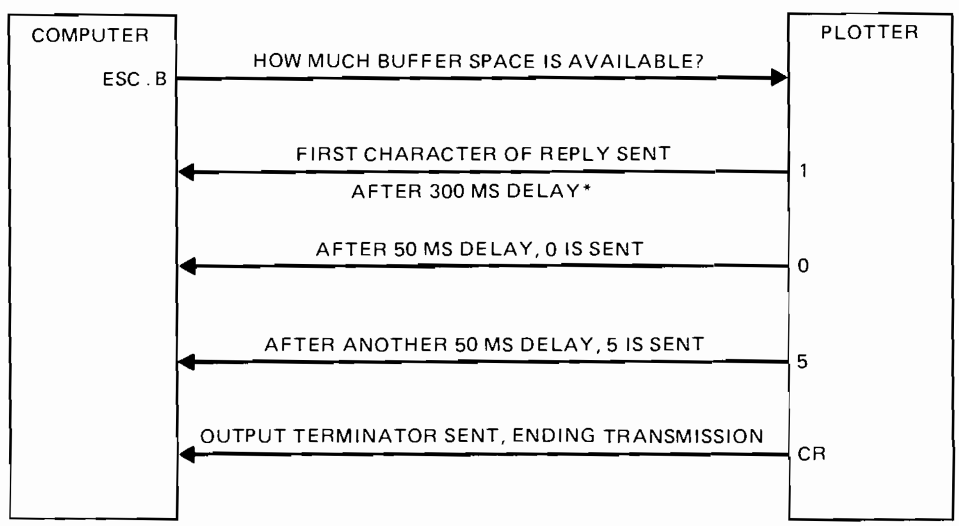
An Example Using ESC . B

The following diagram shows the data exchange between the computer and the plotter when an ESC . B instruction is sent after an intercharacter delay and turnaround delay have been specified using the ESC . M and ESC . N instructions shown. All other parameters of these two instructions assume their default values because they are omitted.

ESC . M 250 :

ESC . N 50 :

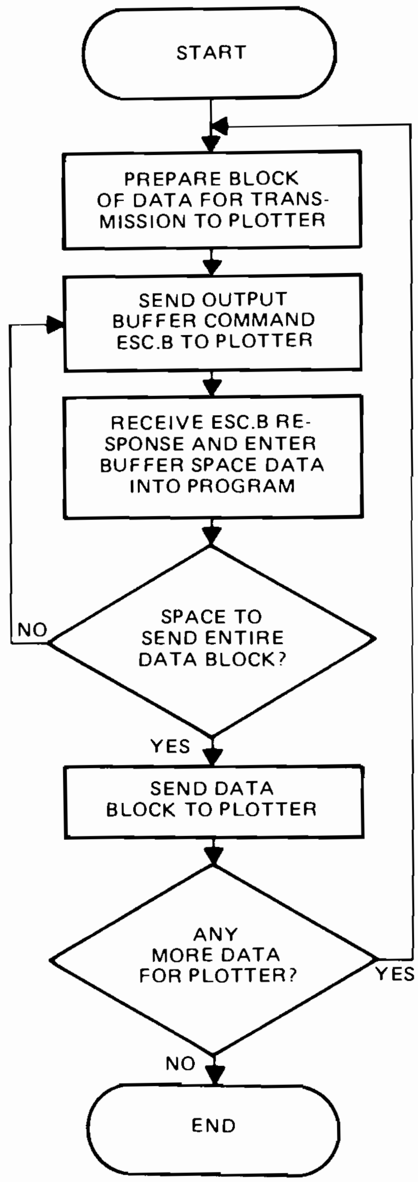
ESC . B



*300 MS DELAY IS TOTAL OF 250 MS TURNAROUND DELAY AND 50 MS INTERCHARACTER DELAY

Software Checking Handshake

The following flowchart shows how a typical program might continually monitor the plotter's buffer. The output buffer space command, ESC . B, is used to ask how many empty bytes remain in the buffer. When the plotter's response is bigger than the next block of data, the program transmits the data block to the plotter.



Software Checking Flowchart

Either or both of the following techniques may be used to reduce the requests concerning buffer space available.

1. Count the number of bytes sent to the plotter and delay the initial request for available buffer space until enough data has been sent to fill the buffer once.

2. After a negative replay, wait a short time before again asking how much buffer space is available.

SUMMARY

All aspects of interfacing a plotter with a computer using the RS-232-C/V.24 interface have been discussed in this application note. Your task is to determine and establish the best handshake for your particular system. The key to determining which type of handshake is best for a specific application is a thorough understanding of the plotter environment and communication requirements of the host computer.

The following questions concerning communication protocol need to be answered.

1. Does the computer send any character as a result of an input operation that would require definition of an output trigger character?
2. What turnaround delay or intercharacter delay, if any, is required?
3. Does the computer require some output terminator other than carriage return?
4. Does the computer supply an echo-terminate character (usually the case in full-duplex or echoplex systems)?
5. If an enquire/acknowledge handshake is to be used, does the computer require an immediate response string be sent in answer to an enquiry character, so that data is not automatically sent after a certain time interval but before the acknowledgment string has been sent?

If the answer to any of the above questions is yes, specify the appropriate value as the corresponding parameter of the ESC . M and/or ESC . N command.

One of the four handshakes — hardwire, Xon-Xoff, enquire/acknowledge, or software checking — can be established using other device control commands. The choice is limited by the plotter environment — hardwired or remote — and the capabilities of the computer. A given computer may or may not support Xon-Xoff protocol or a true ENQ/ACK handshake. Once a handshake has been established, all subsequent exchanges of plot data and plotter responses will conform to that communication format.

When setting up a handshake, you should refer to your plotter manual as well as to this document. This document has focused on how to choose the best handshake for a system or application, and included methods for determining which parameters are required on a given system. Each computer manual has a detailed description of every device control instruction, including parameters and allowable parameter ranges. Using your manual along with this document should enable you to set up a handshake and successfully communicate with your plotter.

Quick Reference of Handshake Types

The following table is a summary of the basic instructions required for each handshake. Please refer to the individual discussions in this plotter note for variations on the combinations of instructions for each handshake, and for examples of typical parameters.

Hardwire	Handled by Operating System		Handled by Software	
	Xon-Xoff	ENQ/ACK (Mode 2)	ENQ/ACK (Mode 1)	Buffer Checking
ESC . @	ESC . N	ESC . I	ESC . M	ESC . M
	ESC . I		ESC . H	ESC . B

Summary of Device Control Instructions Used to Establish Handshakes on Hewlett-Packard RS-232-C Plotters

ESC . @ maximum buffer size (ignored by 7470)[1]; set configuration options (which include hardwire handshake)[2]:

ESC . B

Outputs the number of bytes currently available for data in the buffer. Response is a decimal number of four digits or less with no decimal point.

ESC . H block size[1]; enquiry character[3] or omitted; acknowledgment string[4]:

ESC . I block size or Xoff threshold level[1]; enquiry character[3] or omitted; acknowledgment string or Xon trigger character(s)[4]:

ESC . M turnaround delay[1]; output trigger character[3]; echo-terminate character[3]; output terminator(s)[5]; output initiator[3] (7470 and 7580/85 only):

ESC . N intercharacter delay[1]; immediate response string or Xoff trigger characters[4]:

[1] An integer, base 10, no decimal point.

[2] Decimal equivalent of a four-bit word, in ASCII.

[3] The integer decimal equivalent of an ASCII character.

[4] The integer decimal equivalent(s) of from 1 to 10 ASCII characters, separated by semicolons.

[5] The integer decimal equivalent of 1 or 2 ASCII characters, separated by a semicolon.

APPENDIX

To specify values for the parameters of device control instructions, use the integer decimal equivalents of the ASCII characters.

ASCII Code Table

ASCII Character	Decimal Equivalent	Binary Equivalent	Octal Equivalent	Hexadecimal Equivalent
NULL	0	00000000	000	00
SOH	1	00000001	001	01
STX	2	00000010	002	02
ETX	3	00000011	003	03
EOT	4	00000100	004	04
ENQ	5	00000101	005	05
ACK	6	00000110	006	06
BEL	7	00000111	007	07
BS	8	00001000	010	08
HT	9	00001001	011	09
LF	10	00001010	012	0A
VT	11	00001011	013	0B
FF	12	00001100	014	0C
CR	13	00001101	015	0D
SO	14	00001110	016	0E
SI	15	00001111	017	0F
DLE	16	00010000	020	10
DC1	17	00010001	021	11
DC2	18	00010010	022	12
DC3	19	00010011	023	13
DC4	20	00010100	024	14
NAK	21	00010101	025	15
SYN	22	00010110	026	16
ETB	23	00010111	027	17
CAN	24	00011000	030	18
EM	25	00011001	031	19
SUB	26	00011010	032	1A
ESC	27	00011011	033	1B
FS	28	00011100	034	1C
GS	29	00011101	035	1D
RS	30	00011110	036	1E
US	31	00011111	037	1F
SP	32	00100000	040	20
!	33	00100001	041	21
"	34	00100010	042	22
#	35	00100011	043	23
$	36	00100100	044	24
%	37	00100101	045	25
&	38	00100110	046	26
'	39	00100111	047	27
(	40	00101000	050	28
)	41	00101001	051	29
*	42	00101010	052	2A
+	43	00101011	053	2B
,	44	00101100	054	2C
–	45	00101101	055	2D
.	46	00101110	056	2E
/	47	00101111	057	2F
0	48	00110000	060	30
1	49	00110001	061	31
2	50	00110010	062	32
3	51	00110011	063	33
4	52	00110100	064	34
5	53	00110101	065	35
6	54	00110110	066	36
7	55	00110111	067	37

ASCII Code Table (Continued)

ASCII Character	Decimal Equivalent	Binary Equivalent	Octal Equivalent	Hexadecimal Equivalent
8	56	00111000	070	38
9	57	00111001	071	39
:	58	00111010	072	3A
;	59	00111011	073	3B
<	60	00111100	074	3C
=	61	00111101	075	3D
>	62	00111110	076	3E
?	63	00111111	077	3F
@	64	01000000	100	40
A	65	01000001	101	41
B	66	01000010	102	42
C	67	01000011	103	43
D	68	01000100	104	44
E	69	01000101	105	45
F	70	01000110	106	46
G	71	01000111	107	47
H	72	01001000	110	48
I	73	01001001	111	49
J	74	01001010	112	4A
K	75	01001011	113	4B
L	76	01001100	114	4C
M	7	01001101	115	4D
N	78	01001110	116	4E
O	79	01001111	117	4F
P	80	01010000	120	50
Q	81	01010001	121	51
R	82	01010010	122	52
S	83	01010011	123	53
T	84	01010100	124	54
U	85	01010101	125	55
V	86	01010110	126	56
W	87	01010111	127	57
X	88	01011000	130	58
Y	89	01011001	131	59
Z	90	01011010	132	5A
[	91	01011011	133	5B
\	92	01011100	134	5C
]	93	01011101	135	5D
^	94	01011110	136	5E
_	95	01011111	137	5F
`	96	01100000	140	60
a	97	01100001	141	61
b	98	01100010	142	62
c	99	01100011	143	63
d	100	01100100	144	64
e	101	01100101	145	65
f	102	01100110	146	66
g	103	01100111	147	67
h	104	01101000	150	68
i	105	01101001	151	69
j	106	01101010	152	6A
k	107	01101011	153	6B
l	108	01101100	154	6C
m	109	01101101	155	6D
n	110	01101110	156	6E
o	111	01101111	157	6F
p	112	01110000	160	70
q	113	01110001	161	71
r	114	01110010	162	72
s	115	01110011	163	73
t	116	01110100	164	74
u	117	01110101	165	75
v	118	01110110	166	76
w	119	01110111	167	77

ASCII Code Table (Continued)

ASCII Character	Decimal Equivalent	Binary Equivalent	Octal Equivalent	Hexadecimal Equivalent
x	120	01111000	170	78
y	121	01111001	171	79
z	122	01111010	172	7A
{	123	01111011	173	7B
|	124	01111100	174	7C
}	125	01111101	175	7D
~	126	01111110	176	7E
DEL	127	01111111	177	7F

*EIA RS-232-C and CCITT/V.24 Interface Lines for Hewlett-Packard Plotters with One Connector**

Wire/Signal Name	Circuit		Function		Signal Direction Relative to Plotter
	RS-232-C	CCITT/V.24	HP 7470	HP 7580A HP 7585A	
Protective** Ground	AA/1	101	Used	Used	
Transmitted Data	BA/2	103	Data line from plotter to computer	Data line from plotter to computer	C ◄ P
Received Data	BB/3	104	Data line to plotter from computer	Data line to plotter from computer	C ► P
Request to Send	CA/4	105	Always high Internally strapped		C ◄ P
Signal Ground	AB/7	102	Used	Used	
Secondary Transmit Data	SBA/14	118	Data line from plotter to terminal	Not used	P ► T
Secondary Received Data	SBB/16	119	Data line to plotter from terminal	Not used	P ◄ T
External Clock Input	DD/17	115	Used	Not Used	C ◄ P
Data Terminal Ready	CD/20	108.2	Used	Used	C ◄ P

*HP Models 7580/85A and 7470 have a single connector which must go to the host computer ("modem"). An additional cable is available with the 7470 to enable connection of the plotter between a modem and terminal.

**Cannot be used for Signal Ground.

EIA RS-232-C and CCITT/V.24 Interface Lines for Hewlett-Packard Plotters with Two Connectors Labeled Modem and Terminal*

Wire/Signal Name	Circuit		Signal Direction Between Computer or Modem and Plotter
	RS-232-C	CCITT/V.24	
Transmitted Data	BA/2	103	C ◄ P
Received Data	BB/3	104	C ► P
Request to Send	CA/4	105	C ◄ P
Clear to Send	CB/5	106	C ► P
Data Set Ready	CC/6	107	C ► P
Signal Ground	AB/7	102	
Data Terminal Ready	CD/20	108.2	C ◄ P C ◄ T
Received Line	CF/8	109	

*HP Models 7220, 7221, 7580B, and 7585B each have two connectors, one labeled MODEM and the other TERMINAL.

Plotter		Signal Direction Between Terminal and Plotter
(Modem Connector)	(Terminal Connector)	
Data line from plotter to computer	Data line to plotter from terminal	P ◄ T
Date line to plotter from computer	Data line from plotter to terminal	P ► T
Clear to Send Plotter-activated; says "Prepare to receive data from the plotter"	Request to Send Terminal-activated; says "Prepare to receive data from the terminal"	P ◄ T
Modem-activated; says "Ready to receive data from the plotter"	Plotter-activated; tells terminal that plotter is ready to receive	P ► T
Modem-activated; says modem is operational	Plotter-activated; tells terminal that plotter is operational	P ► T
Used	Used	
In NORMAL mode: Plotter-activated; tells modem that plotter is operational In DTR BYPASS mode: wired through so signal supplied to terminal	No connection	
No connection	Always high (tied)	P ► T

Finding the ESC . M Parameter Values for Your System

If you are not sure what parameters your system needs for output trigger, echo terminate, and output terminator characters, running the following FORTRAN program on your computer may help you identify these parameters.

Setting Up Your System

You should direct the output of this program to the output port to which the plotter will be connected. Replace the plotter with a standard ASCII CRT terminal. Set the terminal (and the plotter) to the mode which matches your system, i.e., full-or half-duplex, echo on or off, and odd, even, or no parity. If the system does not echo transmissions, the proper value for the echo terminator, the third parameter of the ESC . M instruction, is zero. Set the baud rate on both the plotter and the terminal to the fastest rate allowed by the plotter and computer. Disable any automatic line feed capability in the terminal. Later in this test you will want to place the terminal in "display functions" mode, if possible. In this mode, a corresponding symbol is displayed for each ASCII control character received. Thus, you can see normally nonprinting characters such as ESC and LF displayed. Most HP terminals, Tektronix terminals, and some other CRT terminals have this capability. For now, be sure the terminal is not in display functions mode.

Setting Up the Test

Assign the FORTRAN unit numbers 10 and 11, specified in the WRITE and READ statements of the following program, to the output port to which the terminal is connected. If your system does not use the standard FORTRAN format specifier "1H+" to suppress the line feed before output, consult your FORTRAN programming documentation, and substitute the FORMAT specifier which will suppress line feeds.

```
      PROGRAM IOTEST
      WRITE(10,9000)
9000  FORMAT(1H+,13HABCDEFGHIJKLM)
      WRITE(10,9001)
9001  FORMAT(1H+,13HNOPQRSTUVWXYZ)
C
C THESE 2 LOOPS ARE TO CAUSE A DELAY BETWEEN THE OUTPUT
C OF THE 26 CHARACTERS AND THE TRIGGER CHARACTER FOR THE
C READ OF THE INTEGER AND A DELAY BETWEEN ENTERING THE
C INTEGER AND THE END OF THE PROGRAM.
C
      X=1.0
      DO 100 I=1,32000
         X=X+2.0-3.0*2.0-8.0
100   CONTINUE
      READ(11,9002) IANS
9002  FORMAT(I1)
      DO 200 I=1,32000
         X=X+2.0-3.0*2.0-8.0
200   CONTINUE
      STOP
      END
```

What's in the DO Loops

The two DO loops are used to create a noticeable time delay before and after the READ operation so you will be able to observe the action of the terminal's cursor. Faster systems may require more delay; simply increase the number of times the loop is executed. When the program runs, the WRITE statements should overprint on the same line, and the program should stop, waiting for your response. If the WRITE statements print on separate lines, check to see if you have used the proper format specifier for suppressing line feeds in the FORMAT statement, and be sure your terminal was not in display functions mode and that auto line feed was disabled.

Testing for the Output Trigger Character

Now set your terminal to display functions mode and run the program again. To check which character you should use for a trigger character, note any character displayed after the Z while the program pauses for your input. Often it is a non-printing control character, but a question mark, colon, or some other prompt is sometimes used. You should specify the last displayed character as the output trigger character by sending its decimal equivalent as the second parameter of the ESC . M command.

If your terminal does not have display functions mode, carefully observe what happens to the terminal's cursor while the text is written and where the cursor is when it is waiting for your response. If the cursor stops at the end of the line of printed characters, probably no trigger character is required. If the cursor is on the line immediately below the characters, a line feed is being sent. Note whether the line feed appears to be printed together with the line of printed characters, or if there appears to be some delay after the last character is printed before the line feed occurs. If a delay is evident, the line feed is being sent as part of the READ operation. Unless your system documentation can provide other information, it is probably safe to assume that a line feed character is being sent as, and should be specified as, the output trigger character.

Testing for the Echo-Terminate Character

The next step will determine what you should use as the echo-terminate character. If you are in half-duplex mode or your system can operate with no echo, enter zero as the parameter, signifying no echo terminator. If you are in full-duplex mode with echo enabled, observe what happens to the cursor after you type a one-digit number followed by a carriage return.

For terminals with display functions mode enabled, if a line feed character or some other character is displayed following the carriage return, that character is being appended to the echoed response and should be specified as the echo terminator. Otherwise, specify the carriage return character as both the echo terminator and output terminator.

For terminals without display functions mode, if the cursor simply returns to the left margin on the same line, your system does not append a line feed to the echoed response, and you will need to specify the carriage return as the echo terminator. If the cursor returns to the left margin of the next line down, a line feed is being appended to the end of the echo, and should be used as the echo-terminate character.

Echo Terminator Values for Certain Systems

The DEC PDP-11 systems with RT-11 and RSTS/E and VAX VMS operating systems append a line feed to the echo, while the DEC RSX-11M operating system does not. Most other systems also append a line feed. If you change the plotter's output terminator (using ESC . M) on a system which does not append a line feed, specify the same character as both the output terminator and the echo terminator.

Computer Systems with Which Xon-Xoff Handshake Has Been Used with Hewlett-Packard RS-232-C Plotters

This list is intended to suggest that implementation may be possible on your system. Names or locations of installations using this handshake with a particular computer are not available from Hewlett-Packard.

Digital Equipment Corp. — all models

Hewlett-Packard 3000

Honeywell 6600

Prime Computer — all models

HEWLETT
PACKARD

(11) 5953-4124
APPLICATION NOTE 229-6 APRIL 1983 PRINTED IN U.S.A.