hewlett·packard interface bus users guide

9830A

HEWLETT-PACKARD INTERFACE BUS

USERS GUIDE

for systems based on the HP 9830A Calculator

HEWLETT-PACKARD COMPANY, 5301 STEVENS CREEK BLVD., SANTA CLARA, CALIF. 95050

NOVEMBER 1974

□□

CONTENTS

(Continued)

CONTENTS (Continued)

FIGURES

TABLES

Chapter 1
AN OVERVIEW OF THE INTERFACE BUS

The Hewlett-Packard Interface Bus (HP–IB) provides an economical yet extremely powerful means of communication between instruments in a measurement system. The bus provides two way data flow over a single cable using standardized interface techniques. The interface design allows an economical means of forming a system with only two instruments; yet the same design allows highly sophisticated and efficient system operations with a calculator and up to fourteen other devices. In addition to this exceptional flexibility, the Hewlett-Packard Interface Bus is especially easy to use compared to other interfacing schemes.

This user's guide describes how to assemble and program systems using the HP 9830A Programmable Calculator and other HP bus-compatible instruments. It is assumed that the user is already familiar with the HP 9830A Calculator and normal calculator programming techniques. This manual covers the additional programming techniques that pertain to operations of the interface bus.

Likewise, it is assumed that the user is familiar with the basic capabilities and manual operation of the instruments he will use in his system. A knowledge of each instrument adds to a knowledge of the general Interface Bus procedures to provide the skills necessary for system operation.

MAJOR BUS SYSTEM CAPABILITIES

The HP 9830A Calculator plus the Hewlett-Packard Interface Bus offers a wide range of capabilities for system applications. These capabilities include remote control of devices on the bus and data transfer between devices on the interface bus. Up to fourteen devices plus the 9830A Calculator may be connected to the bus to form a measurement system. The calculator can control the operation of these devices, providing substantial improvements in measurement speed and reliability.

The outstanding data handling power of the calculator adds a major benefit. The HP 9830A provides storage for measurement data or virtually instantaneous data reduction. It can compute averages, check for linearity, determine statistical distributions, and perform a wide variety of other data manipulations.

In addition, the calculator can drive any of the standard calculator peripherals such as the plotter or paper tape punch. These calculator peripherals operate through their own calculator I/O interfaces independent of the Hewlett-Packard Interface Bus. The operating manuals for these peripherals describe the use of each of these accessories. Thus the calculator can provide graphical hard-copy output to show the result of system measurements. Such output provides a permanent and easy to interpret record of system measurement results. For example, a final test system for a product might plot a graph that serves as the record of calibration and performance.

INTERFACE BUS OPERATION

The Interface Bus uses 16 signal lines to connect all units of a system in parallel. Eight of the bus lines carry data bits; three of the lines coordinate the flow of data. The remaining five lines are bus management lines.

Each device of an instrumentation system using the Interface Bus follows a strict protocol that enables bus operations to proceed in an orderly manner. By design, the devices always act in assigned roles. These roles—talkers, listeners and controller—provide the basis for all information flow on the bus.

Talkers, Listeners and Controller

Any device that can send data on the bus is a TALKER. Examples of talkers include many digital measurement instruments. Some HP instruments include the 3490A Digital Voltmeter, the 5345A Electronic Counter and the 3570A Network Analyzer. These devices are talkers since they can output measurement results over the bus. The 9830A Calculator is a talker since it can send a variety of information on the bus. Only one talker may be active at a time. The active talker is the sole device with the right to transmit data onto the bus.

Any device that can receive data over the bus is a LISTENER. The HP 59304A Remote Display is a listener because it can receive numeric data from other devices. Talkers may also be listeners. The 3490A DVM, the 3570A Network Analyzer and the 5345A Counter mentioned in the previous paragraph are listeners as well as talkers. These devices are listeners because they can receive remote programming instructions through the bus. The HP 9830A Calculator is a listener since it can receive data such as measurement results from other devices on the bus. Any number of listeners can be active concurrently. Only active listeners can receive data that the active talker transmits onto the bus.

The 9830A Calculator will always act as the CONTROLLER in the systems described in this manual. Only the controller in a system can designate the active talker and active listeners. Therefore the controller manages orderly flow of information over the bus. To assign a talker and listeners to be active, the controller uses a technique called "addressing".

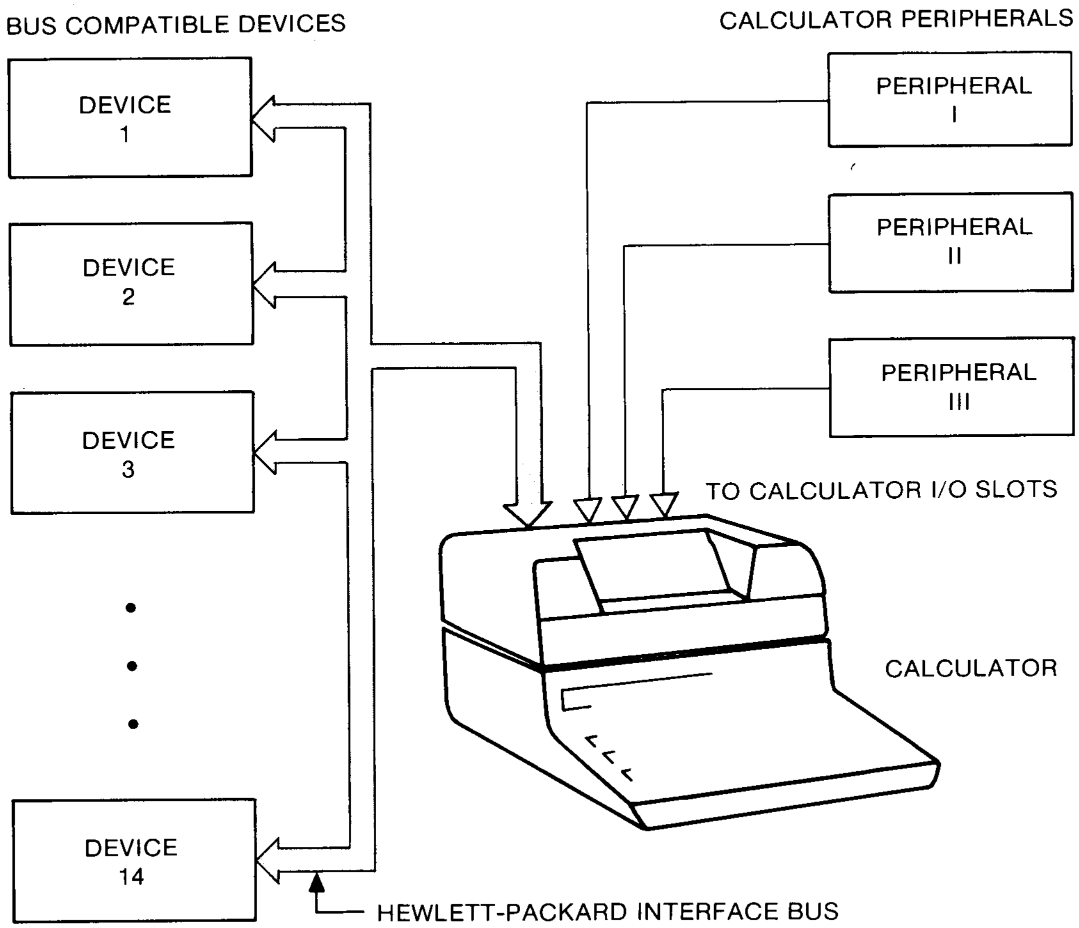


Figure 1-1. Instrument System Using the HP 9830A Calculator and HP-IB

Addressing

Every talker and every listener will have an identifying code called an address. The calculator (controller) uses the addresses to designate active talkers and listeners. The calculator can cause a talker to become active by "addressing" the device to talk. Likewise, the calculator can address one or more listeners on the bus to listen, that is, to become active listeners. After the controller has addressed a talker and some set of listeners, the active talker may transmit data to the active listeners.

For simplicity, the remainder of this text uses the term "talker" to mean active talker and the term "listener" to mean active listener.

Data Transfer

The talker transmits data to listeners in a byte-serial, bit-parallel format. An 8-bit binary code makes up each byte of data. Typically, HP devices use the binary codes of the ASCII character set for each byte. When the talker has transmitted one character (byte) and it has been received by the listeners, the talker may send the next character. Thus data transfer is character serial.

The data transfer rate depends on the speed of the talker and the listeners, not some fixed system clock. This asynchronous data transfer allows great flexibility in system design. A wide variety of devices may effectively operate together in a system even though the devices use different basic transfer speeds. The HP 9830A Calculator can send and receive data at approximately one thousand bytes (characters) per second.

Chapter 2
ASSEMBLING THE SYSTEM

□□

The procedure for interconnecting an instrumentation system using the HP-IB is simple provided an orderly approach is followed. The three basic steps include gathering the proper equipment, setting the appropriate address switches, and connecting the cables.

EQUIPMENT

The following equipment makes up an instrumentation system using the Hewlett-Packard Interface Bus:

- HP 9830A Programmable Calculator
- Bus Interface Kit
- Bus Interface Cables (HP 10631A/B/C)
- Bus compatible instruments and devices (up to 14)

Figure 2-1 shows a typical set of equipment for an interface bus system. Each device must include all options and accessories necessary for interface bus operation. The HP 9830A Programmable Calculator will be used to control the system; the Bus Interface Kit provides the items that enable the calculator to communicate with devices connected by the bus. These items include the Bus Interface I/O Card and the 11272B Extended I/O ROM.

Install the 11272B Extended I/O ROM in any ROM slot as shown in Figure 2-2. This ROM provides the commands which control the interface bus. You may use other ROM's in the other available slots to provide extra calculator capabilities. Installation of the 11274B String Variables ROM is highly recommended. The use of string variables can greatly simplify programming operations on the Interface Bus.

Figure 2-1. System Equipment

Both the 11272B Extended I/O and the 11274B String Variables ROM's can be factory installed in the 9830A as options 272 and 274 respectively. If your calculator already has these ROM's, you do not need the corresponding plug-in ROM blocks.

Next, install the Bus Interface Card in any of the I/O slots in the rear panel of the calculator as shown in Figure 2-3. You may use other I/O cards in other slots so the calculator can control standard peripheral devices. The calculator will control these additional peripherals distinct from the operations of the interface bus.

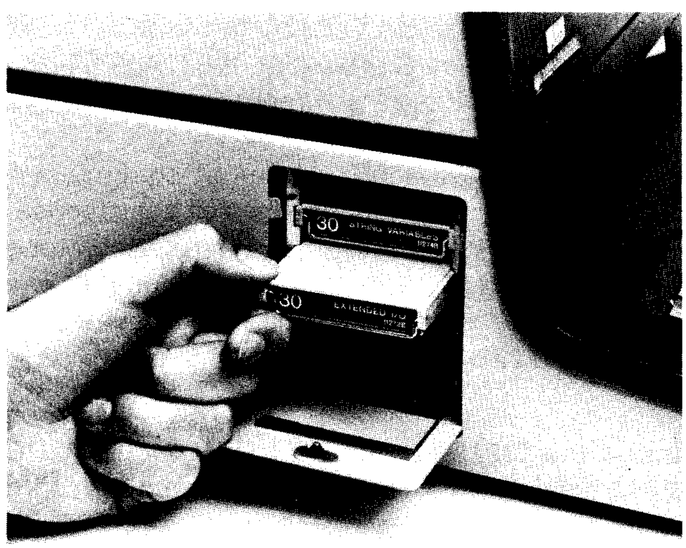


Figure 2-2. Installing the 11272B Extended I/O ROM

SETTING ADDRESSES

The next step in assembling the system is to assign addresses to each device to be used on the interface bus. The device "address" provides the identity to distinguish it from other devices on the bus. The address in an interface bus system is just like a phone number in a telephone system. Phone numbers allow the central switchboard to route calls to the proper phones. Similarly, the calculator will use device addresses to select the talker and listeners on the interface bus.

Talk Addresses and Listen Addresses

A device may have either a TALK address, a LISTEN address or both. The calculator may then use the appropriate code to address the device either to talk or listen. For example, a measurement instrument such as the HP 5340A Microwave Counter has both a talk address and a listen address. The instrument receives programming instructions when addressed to listen. It can output measurement data when addressed to talk. On the other hand, the HP 59304A Remote Display has only a listen address since it can only receive data.

A seven bit binary code forms the address of a device on the interface bus. The first two bits of the code determine whether the address is a talk address or listen address. The last five bits specify the identity of the device. Address codes may be represented as follows:

<table>
<tr><td>Bit Position:</td><td>b_7</td><td>b_6</td><td>b_5</td><td>b_4</td><td>b_3</td><td>b_2</td><td>b_1</td></tr>
<tr><td>Talk Address:</td><td>1</td><td>0</td><td rowspan="2">A_5</td><td rowspan="2">A_4</td><td rowspan="2">A_3</td><td rowspan="2">A_2</td><td rowspan="2">A_1</td></tr>
<tr><td>Listen Address:</td><td>0</td><td>1</td></tr>
</table>

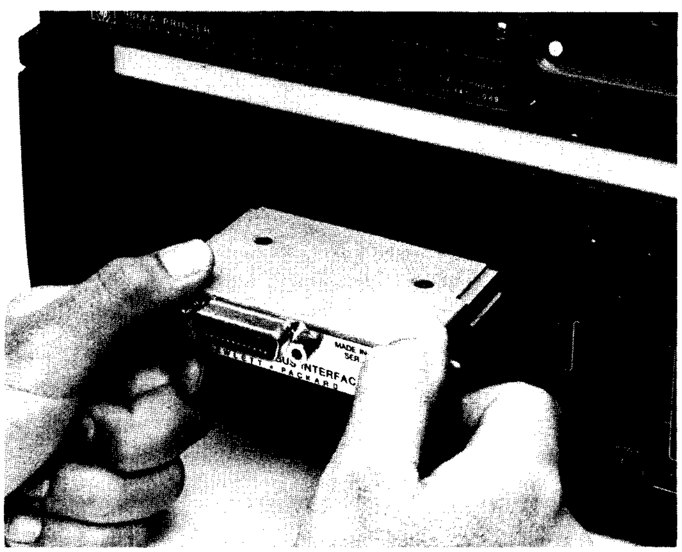


Figure 2-3. Installing the Bus Interface Card

The bit positions A_5 through A_1 represent values for some specific address setting. Note that one set of values in these bit positions identifies both a talk and a listen address.

With the 9830A Calculator, a device address is simply the ASCII character representation of the seven-bit binary code. Therefore an ASCII character uniquely identifies the address of a device. Table 2-1 lists all permissable address codes that may be used on the interface bus.

Table 2-1. Allowable Address Codes

Address Switches					Talk Address Character	Listen Address Character
A_5	A_4	A_3	A_2	A_1		
0	0	0	0	0	@	SP
0	0	0	0	1	A	!
0	0	0	1	0	B	"
0	0	0	1	1	C	#
0	0	1	0	0	D	$
0	0	1	0	1	E	%
0	0	1	1	0	F	&
0	0	1	1	1	G	'
0	1	0	0	0	H	(
0	1	0	0	1	I	)
0	1	0	1	0	J	*
0	1	0	1	1	K	+
0	1	1	0	0	L	,
0	1	1	0	1	M	–
0	1	1	1	0	N	.
0	1	1	1	1	O	/
1	0	0	0	0	P	Ø
1	0	0	0	1	Q	1
1	0	0	1	0	R	2
1	0	0	1	1	S	3
1	0	1	0	0	T	4
1	0	1	0	1	U	5
1	0	1	1	0	V	6
1	0	1	1	1	W	7
1	1	0	0	0	X	8
1	1	0	0	1	Y	9
1	1	0	1	0	Z	:
1	1	0	1	1	[	;
1	1	1	0	0	\	<
1	1	1	0	1	]	=
1	1	1	1	0	^	>

Calculator Talk and Listen Addresses

The Bus Interface Card for the HP 9830A Programmable Calculator comes with factory preset talk and listen addresses. The talk address is "U"; the listen address, "5". You will probably want to leave the addresses as they are; however, you can change the setting should the need arise. All examples in this reference use the standard "U" and "5" as the calculator talk and listen addresses, respectively.

Setting Other Device Talk and Listen Addresses

Most other devices that connect to the interface bus will have a set of address switches or jumpers. These switches allow the user to set addresses to any desired value. Figure 2-4 shows typical address switches for a device. The operating manual for each device will describe the location of the address switches (or internal jumpers) and which positions are the 1's and 0's. To set an address, merely move the switches to the appropriate positions. If the device address is factory preset, you will probably want to leave the address setting as it is.

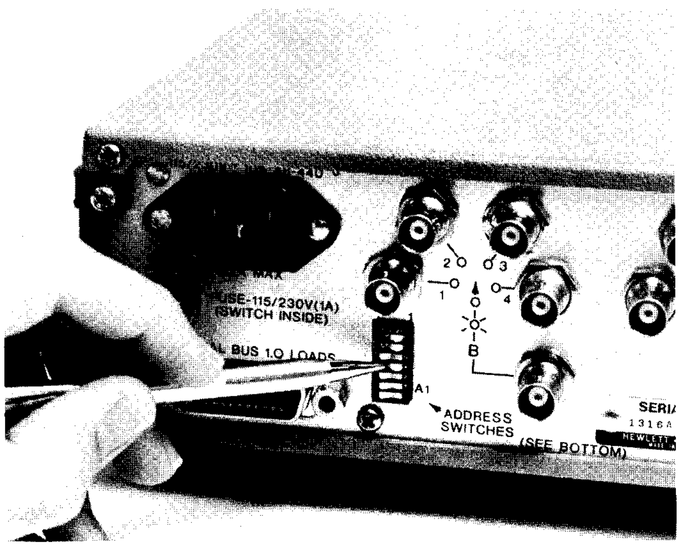


Figure 2-4. Address Switches

Multiple Addresses

Some devices may have two listen addresses or two talk addresses. For example, a device may use one listen address to receive programming information and another to receive measurement data. A measuring instrument such as the 5345A Counter may output data in different formats from each of two different talk addresses.

Such devices may have only four settable address switches. A single setting will determine four addresses, two talk and two listen addresses. The switches will control the A_5 through A_2 positions listed in Table 2-1. For example, address switch settings of $A_5 = 1$, $A_4 = 0$, $A_3 = 0$ and $A_2 = 1$ will produce listen addresses of "2" and "3" with corresponding talk addresses of "R" and "S".

Addressable Mode

Many devices have an additional control switch to set the instrument to the ADDRESSABLE mode. For systems under calculator control, each device must be set to ADDRESSABLE. Figure 2-5 shows a rear panel switch to set the addressable mode. The alternate position for this switch will set the device to TALK ONLY or LISTEN ONLY, depending on whether the device is primarily a talker or listener for data. Such a capability is important for systems without controllers.

The Bus Interface Card for the calculator is always in the addressable mode.

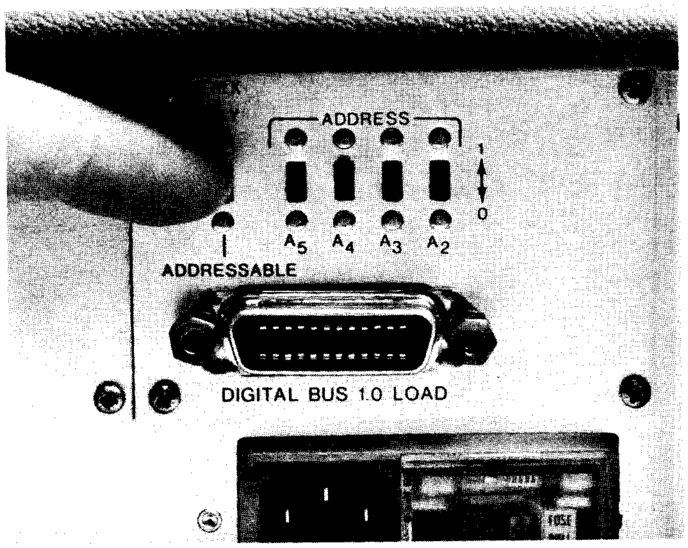


Figure 2-5. Addressable Mode Switch

Address Assignment Table

As you set the addresses for each device, be sure to write down the character codes in a table. Table 2-2 shows an example. Just as a phone system needs a directory, an interface bus system needs an address assignment table. It will save much valuable time during the programming of the system. As you enter addresses in this table, check to be sure that no two devices have been assigned the same address.

Reserved Addresses

Note that one possible setting of the five address switches is not listed in Table 2-1. The code with all switches set to "1" is reserved for special use. Do not set all address switches to the "1" position.

Table 2-2. Sample Address Assignment Table

DEVICE	NAME	TALK ADDRESS	LISTEN ADDRESS
9830A	Calculator	U	5
5345A	Electronic Counter	H	(
3330B	Synthesizer		$
3490A	DVM	V	6
59303A	D-A Converter		+ (Program) * (Data)
59304A	Remote Display 1		1
59304A	Remote Display 2		2
5340A	Microwave Counter	E	%
8660B	Signal Generator		3
3570A	Network Analyzer	A	!

A Review of Addresses

When you have finished setting all addresses, make a quick review for the following items:

- Addresses have been set for every device and entered in the system address assignment table. (Do not use the address code 11111.)
- Each device has been set to addressable mode.
- No two devices have the same address settings.*

CONNECTING BUS CABLES

Connecting the bus cables is the final phase of system assembly. All device addresses should be already set. On many devices, the bus cable will cover the address switches to prevent accidental changes in switch positions later. Thus it is much easier to set addresses before hooking up the cables.

*Two or more devices may have the same listen address only if they are always to receive exactly the same data. For example, two remote displays in different locations may have the same address. Never assign the same address to two talkers.

Bus Cables

The cables for the interface bus use the same piggyback connector on both ends. The connectors may be stacked one on another. This arrangement allows several cables to connect to a single source without the need for special Y's, T's or switch boxes.

The cables are available from Hewlett-Packard in 3 ft., 6 ft. and 12 ft. lengths as the HP 10631A, 10631B and 10631C, respectively.

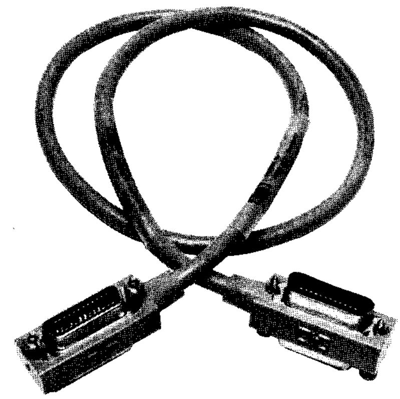


Figure 2-6. Bus Cable

System Configuration

You may interconnect the system components and devices in virtually any order or configuration that you desire. The only requirement is obvious—there must be some path from the calculator through bus cables to every device that will operate on the bus.

As a practical matter, avoid stacking more than three or four cables on any one connector. If the stack gets too long, any force on the stack produces great leverage on the connector and can damage the connector mounting. Check that each connector is firmly screwed in place to prevent it from working loose during use.

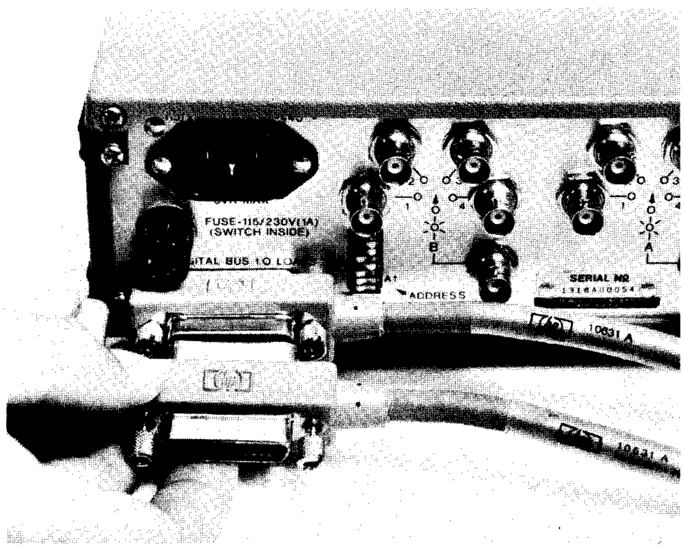


Figure 2-7. Connecting Bus Cables

Cable Length Restrictions

The bus interface electronics must maintain proper line voltage levels and timing relationships to achieve design performance. If the system cable length should grow too long, the interface could no longer drive the lines, and the system would fail to operate properly. Therefore, when interconnecting an interface bus system, observe the following rules:

1. The total cable length for the system must be less than or equal to 20 meters (65 ft.).
2. The total cable length for the system must be less than or equal to 2 meters (6 ft.) times the total number of devices connected to the bus.

Be sure to count the calculator when counting the number of devices in the system. For example, a system with one HP 9830A Calculator and one HP 3490A DVM contains two devices. Up to four meters of bus cable may be used to connect the devices.

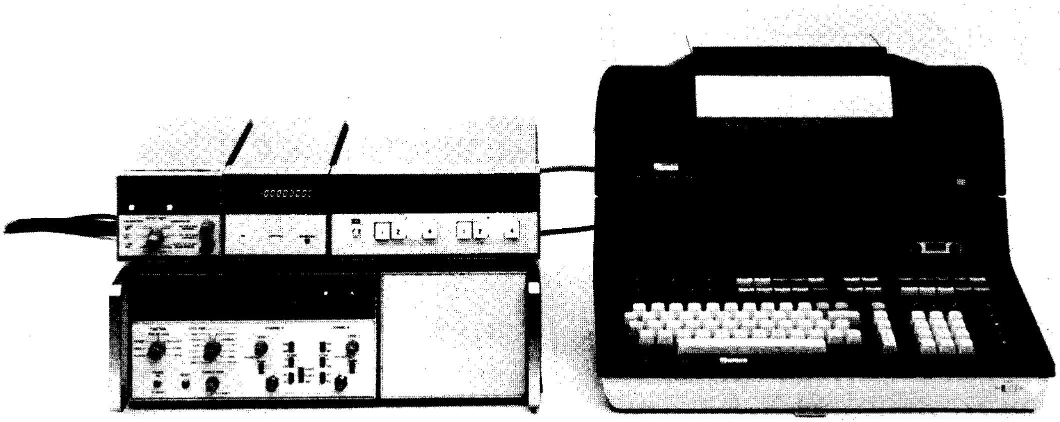


Figure 2-8A. Front View of an Assembled System

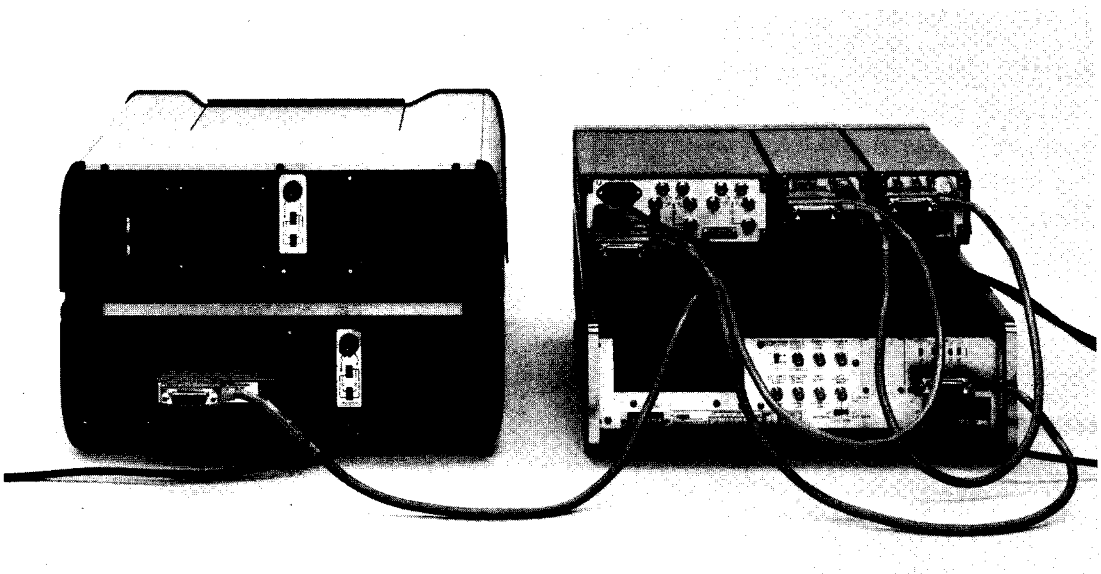


Figure 2-8B. Rear View of an Assembled System

Chapter 3
PROGRAMMING SYSTEM OPERATIONS

Calculator programs can remotely control the operations of bus devices and manage the flow of all information on the bus. This chapter describes the basic calculator capabilities for control of bus operations including remote and local operation, programming instruments, and sending and receiving data.

Of course the calculator may perform its other standard functions. For example, the calculator may store data, compute statistical results, drive a plotter or even ask for keyboard inputs from the operator. Instructions for such standard functions may be found in the calculator operating manual and the manuals for the plug-in ROM's.

THE BUS COMMAND STATEMENT

The calculator controls most operations of bus devices using the BUS COMMAND statement. BUS COMMAND is the primary statement for addressing a talker and listeners to send and receive data and for programming instruments. To use BUS COMMAND you will refer frequently to two basic sources of information. First, you should have the system address assignment table at hand. Second, you will need a program code set for each programmable device on the bus. The program code set for each device can be found in its operating manual.

CMD serves as the mneumonic for BUS COMMAND with the 9830A Calculator. The basic form for the BUS COMMAND statement is

CMD "<Command String>" [, "<Program Code String>"]

where the portion within brackets is optional.* The command string is used to address a talker and listeners; the program code string is used to transmit remote programming instructions to a device. Discussion of program codes appears later in this chapter in the section on programming instruments; this section describes the use of the command string for addressing talkers and listeners.

If the 11274B String Variables ROM has been installed in your 9830A Calculator you may also use CMD as follows:

CMD A$, B$

A$ will be used as the command string and B$ as the program code string. The string variables ROM greatly increases the flexibility of the calculator for programming operations on the Interface Bus.

Addressing Talkers and Listeners

The calculator addresses devices to talk or to listen by using the appropriate address codes within the command string. A command string contains a sequence of address codes and bus command codes.** The address codes may be found in the System Address Assignment Table. Only one of the various bus command codes is required for the basic programming operations described in this chapter. Chapter 4 discusses the remaining bus commands, which provide advanced system capability.

The Basic Addressing Sequence that we will use is:

〈Unlisten Command〉 〈Talk Address〉 〈Listen Addresses〉

This sequence is made up of three major parts which serve the following purposes:

1. The unlisten command is a bus command with the character code "?". It unaddresses all listeners. After the unlisten command is transmitted, no active listeners remain on the bus.

*Actually command strings and program code strings may follow CMD in the general form
CMD " 〈Command String〉 ", " 〈Program Code String〉 ",
" 〈Command String〉 ", " 〈Program Code String〉 " ...
The last program code string is optional.

**When it addresses a device, the calculator "commands" the device to talk or listen—thus the term command string.

2. The talk address designates the device that is to talk. A new talk address automatically unaddresses the previous talker.

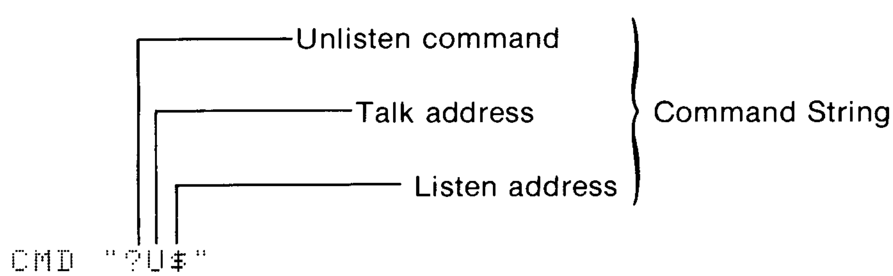
3. The listen addresses designate one or more devices that are to listen. A listen address adds the designated device as a listener along with other addressed listeners.

This basic addressing sequence simply states who is to talk to whom. The unlisten command ("?") plays a vital role in this sequence. It is important that a device receive only the data that is intended for it. Suppose that previous listeners should continue to listen after a new addressing sequence. They would still receive all data transmitted on the bus. The response of these unwanted listeners would be erratic and system operations would be disrupted. A common pitfall in programming is first addressing a device to listen, then later addressing the device to talk without using the unlisten command. Then the device is both talker and listener at once; it will try to "talk to itself", usually to no good.

When a new talk address is transmitted in the addressing sequence, the previous talker is unaddressed. Therefore, only the new talker can send data on the bus and there is no need to routinely use an untalk command in the same manner as the unlisten command.

When using the CMD statement to address devices on the bus, each code in the command string is transmitted and obeyed in the order written. The basic sequence shown here is not mandatory. Nevertheless, you will find it quite convenient, easy to remember, and useful for nearly every application. The following examples illustrate the use of the CMD statement for addressing talkers and listeners. The examples also show some situations where it is possible to abbreviate the basic sequence. These short cuts should only be used if program execution time is critical, since any deviations from the basic addressing sequence can easily lead to programming errors.

Example 1: Basic addressing sequence:



Example 2: Basic addressing sequence with multiple listeners:

- Unlisten command
- Talk address
- Listen addresses

```
CMD "?U123"
```

Example 3: Successive talk addresses:

The statement

- Talk address
- Talk address

```
CMD "UR"
```

is equivalent to

```
CMD "R"
```

since the second talk address supercedes the first.

Example 4: An addressing sequence error:

The statement

- Talk address
- Listen addresses
- Unlisten command

```
CMD "U12?"
```

is equivalent to

```
CMD "?U"
```

since the final "?" in the first statement unaddresses all listeners including the "1" and "2" included in the command string.

Example 5: Addressing a new talker:

With the basic addressing sequence use

Unlisten command

New talk address

Repeat current listen addresses

```
CMD "?R12"
```

or use

New talk address

```
CMD "R"
```

Example 6: Addressing new listeners:

Use the basic addressing sequence:

Unlisten command

Talk address

New listen addresses

```
CMD "?U34"
```

or use

Unlisten command

New listen addresses

```
CMD"?34"
```

Addressing new listeners without addressing a new talker will leave the old talker still addressed.

Example 7: Adding listeners:

To add new listeners to the current group of listeners use the basic addressing sequence:

Unlisten command

Talk address

Repeat current listeners

Add new listeners

```
CMD "?U1234"
```

or use

New listen addresses

```
CMD "34"
```

REMOTE AND LOCAL OPERATION

Devices on the bus may operate under calculator remote control only if the bus is in the remote enabled state. The calculator (controller) sets the state of the bus. When the 9830A Calculator is turned on, it powers up with the interface bus in the remote enabled state. Therefore, devices on the bus normally may operate under calculator remote control. However, it is good programming practice to include the following REMOTE ENABLE statements at the beginning of every calculator program:

Unlisten command

Talk address (calculator)

```
10 CMD "?U"
20 FORMAT B
30 OUTPUT (13,20)768;
```

Code to set remote enabled state

Format statement number

Select code for bus I/O card

These statements insure that the bus is in the remote enable state. Line 30 actually sets the state of the bus. Line 10 is necessary because the calculator must be addressed to talk to execute OUTPUT or WRITE statements to the HP Interface Bus. Remote Enable is necessary because previously run calculator programs may have set the bus to the local state. If the bus is in the local state, then devices on the bus will not respond to remote programming instructions. Instead, the devices will obey their manual controls. The remote enabled state does not automatically shift devices to remote operation, however. Most devices must also be addressed to listen before they switch to remote.

The SET LOCAL statements return all devices to local (manual) operation. The statements

Unlisten command

Talk address (calculator)

```
100 CMD "?U"
110 FORMAT B
120 OUTPUT (13,110)1024;
```

Code to set bus to local state

Format statement number

Select code for bus I/O card

set the bus to the local state.

The Remote Enable and Set Local statements provide the basic tools for controlling the remote/local state of the bus. In addition, the bus commands Local Lockout and Go To Local may also be useful when controlling some devices on the bus. See Chapter 4 (p. 4-4 and p. 4-7) for a description of how these two bus commands may be used to provide additional programming flexibility.

PROGRAMMING INSTRUMENTS

Remote programming of an instrument first requires a thorough knowledge of the instrument capabilities under manual operation. After all, for the most part, remote programming merely substitutes program codes for manual operations of the instrument controls.

Program Code Sets

Each remotely programmable device will have a program code set that controls its operation in bus systems. The operating manual for the device will list its program code set. Portions of the code sets for an electronic counter and an automatic synthesizer are shown in Table 3-1. Note that the codes are simply ASCII characters. Some instruments may use single-character codes; more complex instruments may use character pairs.

The program code set will list all the features of the device that can be remotely controlled. Next check for any special features of remote operation. For example, the device may offer a choice of output modes, or even have some functions that are only available through remote programming.

Table 3-1. Program Code Set Excerpts

FUNCTION:		SAMPLE RATE:	
FREQ A	F0	NOT HOLD	E1
TIME INTERVAL	F1	HOLD	E9
		TAKE A MEASUREMENT	J1
GATE TIME:		OTHER:	
10 SEC	G1	RESET	I1
1 SEC	G0	SWITCH TO LOCAL	E0
100 MSEC	G?	SWITCH TO REMOTE	E8

(a) 5345A Electronic Counter

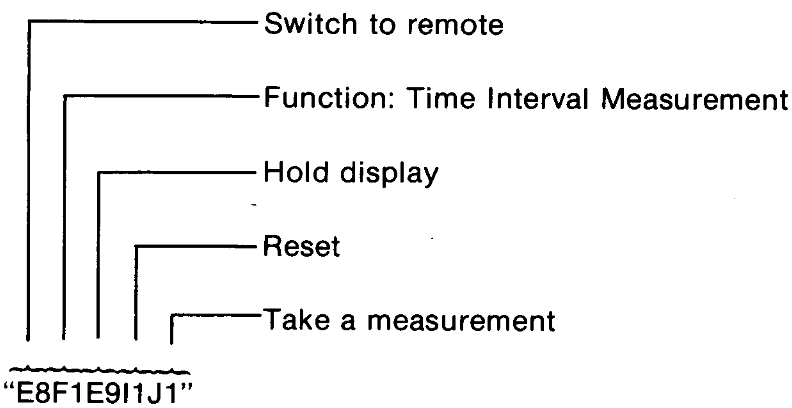
FREQ	L	AMPL	N
FREQ STEP	M	AMPL STEP	O
Hz	=	+dBm	;
kHz	>	−dBm	<
MHz	?		

(b) 3330B Automatic Synthesizer

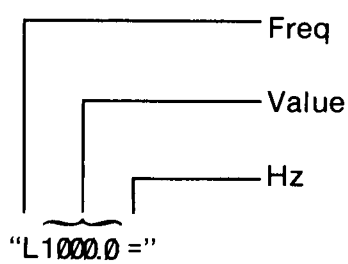
A program code string is merely a series of program codes that will be used to control the operation of a device. As with command strings, the characters of a program code string are transmitted and obeyed in the order written.

Program codes may be separated into two categories—literals and variable values. Literals (a sequence of characters beginning and ending with quotation marks) typically program discrete features of a device; variable values usually control features that can assume a continuous range.

Literals may set the function of a device such as a DVM or electronic counter. The function of a counter can be set to frequency measurement or time interval measurement, clearly not some value in between. An example of a program code string for the HP 5345A Electronic Counter might be (see Table 3-1 for program code set excerpts):



With the HP 3330B Automatic Synthesizer, a program code string to set an output frequency of 1000 Hz could appear as follows:



In this case the desired frequency value is written as a literal within a program code string. A description of how to use variable values to achieve the same result is contained in the section "Using Variable Values as Program Codes."

Using Literals as Program Codes

Use the BUS COMMAND statement when using literals to program a device. The literals form the program code string in the CMD statement. In the command string, use the basic addressing sequence to address the calculator to talk and the device to listen. Since the calculator talk address is "U", the CMD statement takes the form

Unlisten command

Talk address (Calculator)

CMD "?U <Listen Address>", "<Program Code String>"

The following examples illustrate the use of literals for programming devices.

Example 1: 5345A Electronic Counter

Unlisten command

Talk address (Calculator)

Listen address (Counter)

Remote program initialize

Switch to remote

Amplifier check

100 ms gate time

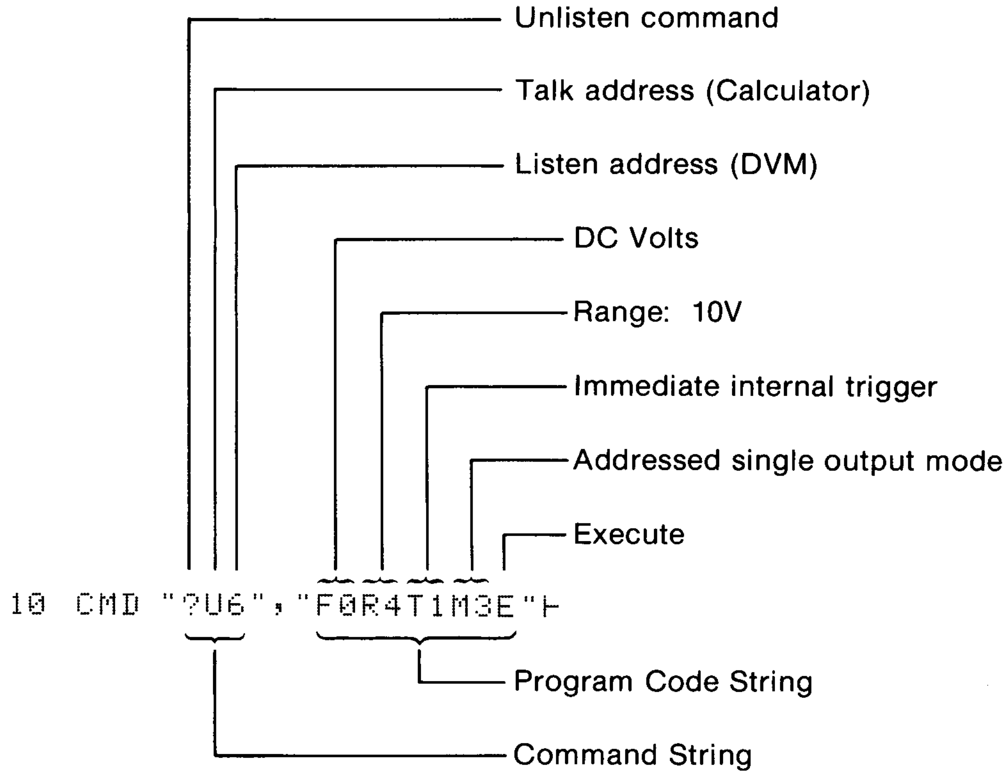
Reset

```
10 CMD "?U(","I2E8E?G?I1"⊢
```

Program Code String

Command String

Example 2: 3490A Digital Voltmeter



Using Variable Values as Program Codes

Sometimes it may be necessary to program a device using a code that takes the value of a variable stored in the calculator. Perhaps the variable contains the desired output frequency or amplitude for a signal source. Using the FORMAT and OUTPUT statements, it is possible to transmit a program code string containing the value of the variable. The CMD statement is still used to address the calculator to talk and the device to listen. The following statements show the basic form for using variable values as programming codes.

```
10  CMD "?U <Listen Address> "
20  FORMAT <format list>
30  OUTPUT (13,20) <variable list>
```

- "?" — Unlisten command
- "U" — Talk address (calculator)
- "13" — Select code for bus I/O card
- "20" — Format statement number

Line 30 above transmits a program code string containing the values of the variables in the variable list. In addition, the FORMAT statement, line 20, may include literals as program codes in the format list.

An example best illustrates using variable values as program codes. Suppose a system includes an HP 3330B Synthesizer with listen address "$". To obtain a frequency output of 1000.0 Hz, we might use literals as program codes in the statement

```
10 CMD "?U$","L1000.0="
```

- "?" — Unlisten command
- "U" — Talk address (Calculator)
- "$" — Listen address (Synthesizer)
- "L" — Freq
- "1000.0" — Value
- "=" — Hz

We can achieve the same result in another way using variable values as program codes. Suppose that the variable A (stored in the calculator register) contains the desired frequency value. Then to transmit the same remote programming information to the synthesizer, we can use the statements in the following example:

Example 1: HP 3330B Automatic Synthesizer

```
10 CMD "?U$"
20 FORMAT "L",F1000.1,"="
30 OUTPUT (13,20)A
```

- "?" — Unlisten command
- "U" — Talk address (Calculator)
- "$" — Listen address (Synthesizer)
- "L" — Code for FREQ
- F1000.1 — Conversion spec.
- "=" — Code for Hz
- A — Variable used to set synthesizer frequency
- 20 — Format statement number
- 13 — Select code for bus I/O card

If the value of A is 1000, then these statements will produce exactly the same results as the previous statement that uses only literals. Note that the format statement can include literals as well as conversion specifications for the variable values.

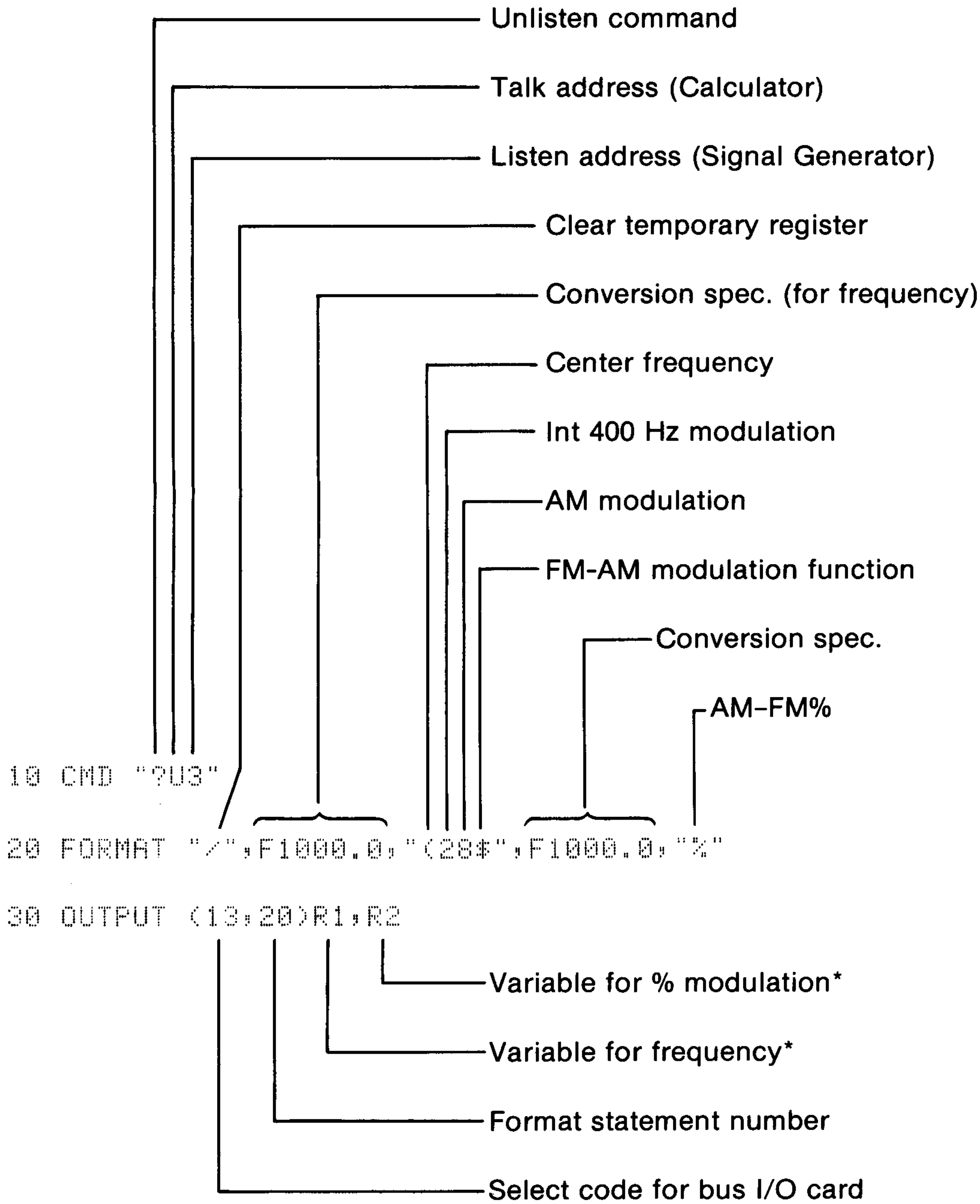
The format specification

F1ØØØ.d

is particularly useful for this example. When used with an OUTPUT statement it deletes all leading spaces from an output data item and places d digits to the right of the decimal point. In the example above, F1ØØØ.1 causes the value of A to be transmitted as the character string "1000.0". See the 11272B Extended I/O ROM Operating Manual for more information on the use of FORMAT statements. Caution: The WRITE statement cannot be used with this format specification to suppress leading spaces.

Here is another example of using variable values as program codes.

Example 2: HP 8660B Synthesized Signal Generator



*The HP 8660B requires a special inverted format for these variables. See the Option 005 Operating Manual Supplement for details.

SENDING AND RECEIVING DATA

The interface bus provides communications between devices. In remote programming devices on the bus, the communication is sending and receiving programming codes. We can list three basic situations for sending and receiving data other than remote programming information. These are

1. Instrument-to-calculator
2. Calculator-to-instrument
3. Instrument-to-instrument

Instrument-to-Calculator

For most applications the calculator uses the ENTER statement to obtain data from another device, as follows:

Unlisten command

Listen address (Calculator)

```
10  CMD "? <Talk Address> 5"
20  FORMAT  <format list>
30  ENTER (13,20)  <variable list>
```

Format statement number

Select code for bus I/O card

In this case the CMD statement addresses the sending device to talk and the calculator to listen. The ENTER statement takes the incoming data and stores the data items in the variables specified in the variable list.

The FORMAT statement must be appropriate to the data string the device will output to the calculator. The 9830A Calculator Operating and Programming Manual describes the several types of conversion specifications that may be used in the format list.

Many devices output leading non-numeric characters in their output data strings. The FORMAT statement must account for any leading non-numeric characters so that the ENTER statement can interpret the numeric information. Examples best illustrate the techniques for receiving data from a device.

Example 1: HP 3490A Digital Voltmeter to HP 9830A Calculator

```
10 CMD "?V5"
20 FORMAT 4X,E12.0
30 ENTER (13,20)A
```

- "?" — Unlisten command
- "V" — Talk address (Voltmeter)
- "5" — Listen address (Calculator)
- 4X — Edit spec to skip leading non-numeric output characters
- E12.0 — Conversion spec.
- A — Variable to contain reading
- 20 — Format statement number
- 13 — Select code for bus I/O card

The HP 3490A Digital Voltmeter may output a measurement in the following form when programmed for DC volts, 10V range:

N (SP) D C + 0 8 3 4 6 2 E - 4 (CR) (LF)

The calculator will skip the four leading non-numeric characters as determined by the FORMAT statement. Then the calculator will read the numeric portion of the output string and set A=8.3462. The format of the input string will override the specification in the FORMAT statement. The only requirement for the numeric field specification is that the field width must be equal to or greater than the actual input field width.

With the 11274B String Variables ROM installed in the calculator, the following sequence will achieve the same results as the example above:

```
10 CMD "?V5"
20 ENTER (13,*)A$
30 A=VAL(A$[5])
```

This programming technique also has the advantage of retaining the leading non-numeric characters in A$. Thus the leading characters can be checked by other statements in the program. Another way to look at leading non-numeric characters is illustrated below.

Example 2: HP 5340A Microwave Counter to HP 9830A Calculator

The HP 5340A Microwave Counter outputs a data string as follows:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
{D or L}	{O or (SP)}	(SP)	d	d	d	d	d	d	d	d	E	+	d	(CR)	(LF)

1: D - direct count; L - phase-locked measurement
2: O - overflow; (SP) - measurement within range

The FORMAT B specification can be used to receive the non-numeric characters.

```
10 CMD "?E5"
20 FORMAT 2B,E12.0
30 ENTER (13,20)A,B,C
```

In line 10: ? — Unlisten command; E — Talk address (Counter); 5 — Listen address (Calculator).

In line 30: A — Variable to receive first character; B — Variable to receive second character; C — Variable to contain reading.

Suppose the HP 5340A outputs the data string

L (SP) (SP) Ø 8 1 6 4 Ø 4 7 E + 3 (CR) (LF)

Variables A and B will receive the decimal equivalent for the first two characters of the data string. C will receive the value of the reading. In this case

A = 76 (decimal equivalent for "L")

B = 32 (decimal equivalent for "SP")

C = 8.164Ø47EØ9 (value of counter's measurement)

Example 3: HP 3570A Network Analyzer to HP 9830A Calculator

? — Unlisten command

A — Talk address (Network Analyzer)

5 — Listen address (Calculator)

```
10 CMD "?A5"
20 FORMAT 2F7.2
30 ENTER (13,20)A,B
40 PRINT "AMPLITUDE",A,"PHASE",B
```

The HP 3570A Network Analyzer may output the following data string after an amplitude and phase measurement:

- Ø 3 4 . 4 8 , - Ø 8 8 . 9 3 (CR) (LF)

Statement 40 will cause the calculator to print the following values after receiving A and B

```
AMPLITUDE       -34.48          PHASE           -88.93
```

Calculator-to-Instrument

The calculator uses the OUTPUT statement to transmit data to devices on the bus as follows:

```
10  CMD "?U <Listen Addresses> "
20  FORMAT <format list>
30  OUTPUT (13,20) <variable list>
```

- "?" — Unlisten command
- "U" — Talk address (Calculator)
- 13 — Select code for bus I/O card
- 20 — Format statement number

The CMD statement addresses the calculator to talk and one or more other devices to listen. Then the OUTPUT statement transmits the variable list in accordance with the FORMAT statement. WRITE may be substituted for OUTPUT under some circumstances. However, OUTPUT is highly recommended since it may be used for more varied applications.

Example 1: HP 9830A Calculator to HP 59304A Remote Display

```
10 A=123.45
20 CMD "?U1"
30 OUTPUT (13,60)A
40 CMD "?U2"
50 OUTPUT (13,70)A
60 FORMAT F7.2
70 FORMAT E12.4
```

- "?" (line 20) — Unlisten command
- "U" (line 20) — Talk address (Calculator)
- "1" (line 20) — Listen address (1st display of 2)
- "2" (line 40) — Listen address (2nd display of 2)

These statements will output the value of A in different formats to each of the two displays.

The first display will show

123.45

But the second display will show

1.2345E 02

Instrument-to-Instrument

A device may transmit data directly to other devices without the data passing through the calculator. The calculator addresses one device to talk and the others to listen. Then the transfer takes place, provided the talker has data to output. The operating manual for the talker will describe its data output format. For useful data transfer to take place, this format must match what the listeners are designed to receive. Some listeners can receive a variety of formats. The operating manual for a listener will describe its response to data inputs. Simply use the following statement for instrument-to-instrument data transfer:

10 CMD "? <Talk Address> <Listen Addresses> "

Example 1: HP 3490A Digital Voltmeter to HP 59304A Remote Display

```
          ┌──────────────── Unlisten command
          │ ┌────────────── Talk address (Voltmeter)
          │ │ ┌──────────── Listen address (Display)
          │ │ │
10  CMD  "?V1"
```

The HP 3490A DVM may output a measurement in the following form when programmed for DC volts, 10V range:

N (SP) D C + Ø 8 3 2 5 Ø E - 4 (CR) (LF)

The remote display will ignore the non-numeric portion of the output string and show

Ø8325ØE-4

Nevertheless, it is advisable to include the calculator as a listener even during instrument-to-instrument data transfer. The calculator should read each transferred value to guarantee that the transfer is complete. The following statements may be used to execute a typical transfer of data with the calculator monitoring the transfer as a listener:

- Unlisten command
- Listen address (Calculator)

```
10  CMD "? <Talk Address>  5  <Listen Addresses> "

20  FORMAT <format list>

30  ENTER (13,20) <variable list>
```

Example 2: HP 5340A Microwave Counter to HP 59303A Digital-to-Analog Converter and HP 59304A Remote Display (Calculator listening)

- Unlisten command
- Talk address (Counter)
- Listen address (Calculator)
- Listen address (Display)
- Listen address (D-A Converter)

```
10 CMD "?E52*"

20 FORMAT 2X,E12.0

30 ENTER (13,20)A
```

The HP 5340A may output a measurement in the form:

L (SP) (SP) Ø 8 1 6 4 6 6 Ø E + 3 (CR) (LF)

Both the D-A converter and the remote display will ignore the leading non-numeric information. The D-A converter will obey its currently programmed instructions. The display will show

Ø8164660E 3

The first statement addresses one device to talk and other devices, including the calculator, to listen. The second and third statements insure that the data transfer is complete before the calculator proceeds to execute its next program statement. Without these program lines, the calculator would continue to execute the following lines of its program. If the calculator were to encounter another CMD statement, it would execute that statement, and assign a new talker and new listeners. In such a case there would be no assurance that the original data transfer had been completed. Therefore, during instrument-to-instrument data transfer, include the calculator as a listener and read in each transferred value to be sure the transfer is completed.

INTERFACE CLEAR

At times you may wish to interrupt a calculator program to regain control of a bus system at the calculator keyboard. Merely press the STOP button on the calculator keyboard. The STOP button controls the INTERFACE CLEAR line of the interface bus. This line is one of the management lines and may be activated only by the controller, in this case the 9830A Calculator. Interface Clear stops all I/O operations on the interface bus by unaddressing all talkers and listeners. The calculator will also obey the STOP key in its normal manner. Thus the user will obtain control from the calculator keyboard.

After an interface clear, you may restart a calculator program from its beginning, modify the program, revise the system configuration, or check a suspected trouble spot. Devices under remote control will remain under remote control even after an Interface Clear. The Set Local statements (p. 3-7) can return all devices to local control if necessary.

A PROGRAMMING EXAMPLE

A simple example can illustrate most basic programming techniques for controlling operations on the Interface Bus. Figure 3-1 represents a voltage-controlled oscillator test system. Devices on the interface bus include the HP 9830A Calculator, the HP 59303A Digital-to-Analog Converter and the HP 5345A Electronic Counter. A voltage-controlled oscillator (VCO) is the device under test. The digital-to-analog converter provides the voltage stimulus to drive the VCO; the electronic counter will measure the frequency output of the VCO. With this system we can measure the transfer characteristic of the VCO (frequency out as a function of voltage in).

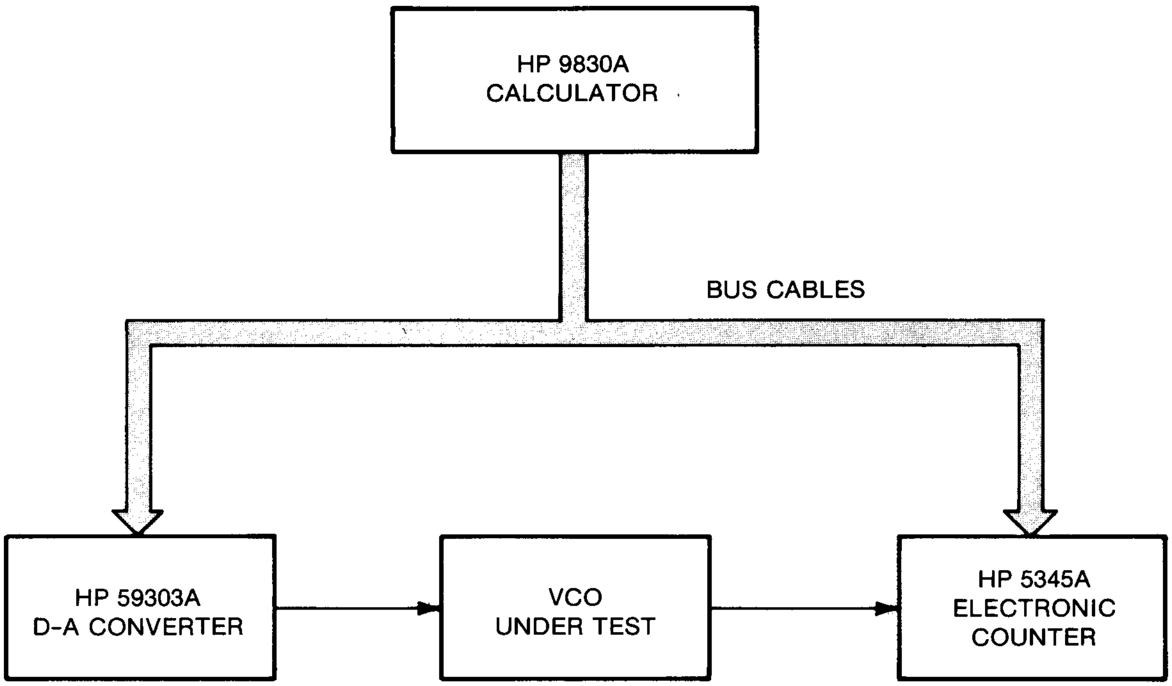


Figure 3-1. VCO Test System Configuration

Table 3-2. Address Assignment Table for VCO Test System

DEVICE	NAME	TALK ADDRESS	LISTEN ADDRESS
9830A	Calculator	U	5
5345A	Electronic Counter	H	(
59303A	D-A Converter		+ (Programming) * (Data)

The following program is a simplified version of an actual test system. With this program the system measures the VCO frequency at 11 voltage levels from -5 to +5 volts in steps of 1 volt. The measurement results are stored in a calculator array R(1) through R(11). However, the system designer could add to the basic program to compute parameters such as non-linearity or to plot the transfer characteristic. The Hewlett-Packard Application Note Series 174-1 through 174-8 illustrates a variety of programs for controlling measurement test systems.

When you write your own programs, keep in mind the following two rules:

1. The calculator must be addressed to talk to execute statements that output data onto the bus. Such statements include OUTPUT, WRITE, and BUS COMMAND when the CMD statement is used to send a program code string.

2. The calculator must be addressed to listen to execute statements such as ENTER that receive data from the bus.

If the calculator is not properly addressed for a bus I/O statement, it will "hang-up" and not complete executing the statement. The programming techniques described in this chapter include proper addressing sequences for each operation. However, if the calculator stops when executing a bus I/O statement, be sure to double-check the address sequence.

Sample Program

```
5 DIM R[11]
10 CMD "?U"
20 FORMAT B
30 OUTPUT (13,20)768;
40 CMD "?U+","E0"
50 CMD "?U(","I2E8E9I1"
60 FOR I=-5 TO 5
70 CMD "?U*"
80 FORMAT F1000.2
90 OUTPUT (13,80)I
100 CMD "?U(","J1"
110 CMD "?H5"
120 FORMAT E16.0
130 ENTER (13,120)R[I+6]
140 NEXT I
150 END
```

Program Explanation

10–30: Remote Enable statements set bus to remote enabled state and allow remote control of devices on the bus.

40: Addresses the calculator to talk [U] and the D-A Converter to listen for programming information [+]. Program code string sets DAC to ±10V output mode [E] and to convert last three digits of data string [0].

50: Addresses calculator to talk [U]; counter to listen [(]. Programs counter to initialize standard remote program [12], switch to remote [E8], hold display [E9], reset [I1].

60: I is the index variable for the measurement loop and is given the initial value -5. I will range from -5 to +5.

70: Addresses calculator to talk [U] and DAC to listen for data [*].

80: Format suppresses leading blanks in output string.

90: Sends a 3-digit data string to DAC for conversion to voltage output.

100: Addresses calculator to talk [U]; counter to listen [(]. Programs counter to take a measurement [J1].

110: Addresses counter to talk [H]; calculator to listen [5].

130: Reads counter measurement into the calculator.

140: Repeats measurement loop.

Chapter 4
BUS COMMANDS

□□

Bus commands allow the system controller to obtain a response from several bus devices at once, rather than with a series of CMD statements. Table 4-1 summarizes the bus commands and their applications. Devices that operate on the interface bus are not required to recognize and obey all bus commands. Each device will usually obey just those commands that have some useful meaning to the device. Check the operating manual for a device to find out which bus commands it obeys.

Table 4-1. Summary of Bus Commands

	COMMAND		DECIMAL CODE	DESCRIPTION	GENERAL FORM AND EXAMPLES
UNADDRESS COMMANDS	UNL	UNLISTEN	63	Clears bus of all listeners.	10 CMD "?"
	UNT	UNTALK	95	Unaddresses the current talker so that no talker remains on the bus.	10 CMD "?U" / 10 CMD "?U" 20 FORMAT 3B / 20 FORMAT 3B 30 OUTPUT (13,20)256,C,512; / 30 OUTPUT (13,20)256,17,512; command code (C) / Local Lockout (17)
UNIVERSAL COMMANDS	LLO	LOCAL LOCKOUT	17	Disables front panel local-reset button on responding devices.	
	DCL	DEVICE CLEAR	20	Returns all responding devices to predetermined states.	
	SPE	SERIAL POLL ENABLE	24	Sets serial poll mode.	
	SPD	SERIAL POLL DISABLE	25	Terminates serial poll mode.	
ADDRESSED COMMANDS	GTL	GO TO LOCAL	1	Returns to local control all responding devices that are addressed to listen.	10 CMD "?U <Listen Addresses> " / 10 CMD "?U89" 20 FORMAT 3B / 20 FORMAT 3B 30 OUTPUT (13,20)256,C,512; / 30 OUTPUT (13,20)256,8,512; command code (C) / Group Execute Trigger (8)
	SDC	SELECTED DEVICE CLEAR	4	Returns to predetermined states all responding devices that are addressed to listen.	
	GET	GROUP EXECUTE TRIGGER	8	Initiates a preprogrammed action by all responding devices that are addressed to listen.	

Bus commands fall into three categories.

1. *Unaddress commands* are obeyed by all addressable devices. These commands unaddress devices that are currently addressed.

2. *Universal commands* affect all responding devices on the bus, whether addressed or not.

3. *Addressed commands* affect only responding devices which are addressed to listen. Addressed commands allow the controller to initiate action by a selected group of devices on the bus.

ISSUING BUS COMMANDS

One bus command, the unlisten command, has already been introduced in Chapter 3. The unlisten command can be issued using a "?" within a command string of a CMD statement as follows:

CMD "?"

Other bus commands may be issued using the following basic sequence of statements:

```
           ┌──────────────────────Unlisten command
           │ ┌────────────────────Talk address (Calculator)
10  CMD "?U"

20  FORMAT 3B

30  OUTPUT (13,20)256,C,512;
             │  │     │
             │  │     └────────── Bus command code
             │  └──────────────── Format statement number
             └─────────────────── Select code for bus I/O card
```

The OUTPUT statement transmits the bus command code. The CMD statement is required because the calculator must be addressed to talk to be able to execute an OUTPUT statement. A decimal equivalent code may be substituted for the variable C.

UNADDRESS COMMANDS

The two unaddress commands allow the controller to unaddress bus devices which are talkers or listeners. The unaddress commands and their corresponding decimal codes are

UNLISTEN	UNL	63
UNTALK	UNT	95

Unlisten (UNL)

The UNLISTEN command unaddresses all listeners on the bus. UNL is the most widely used of all the bus commands. In Chapter 3, Programming System Operations, the unlisten command is used routinely at the beginning of the basic addressing sequence.

By unaddressing all listeners, UNL prevents unwanted listeners from remaining on the bus. UNL is the one bus command that may be included as a character in a command string when using the 9830A Calculator. The following example illustrates this command.

Unlisten command
Talk address (Calculator)
Listen address
Listen address

```
10 CMD "?U24"
```

After the calculator executes this statement, only the devices "2" and "4" will be addressed to listen. If UNL were omitted from the command string, these two devices plus previously addressed listeners would now be addressed to listen.

Untalk (UNT)

The UNTALK command is available in the event that it becomes necessary to unaddress a talker without assigning a new talker. With the 9830A Calculator the following statements issue the untalk command:

```
10 CMD "?U"

20 FORMAT 3B                  ┌──────────── Untalk
                              │
30 OUTPUT (13,20)256,95,512;
```

The untalk command is used only in special cases. Recall that addressing one device on the bus as a talker automatically unaddresses the previous talker. Thus it is not necessary to use the untalk command routinely like the unlisten command.

The untalk command might be used to suspend output from a device on the bus. If the device is a talker and sending data, then UNT will halt the data output.

UNIVERSAL COMMANDS

All responding devices in a bus system will obey a universal command when it is issued whether they are addressed to listen or not. Thus the controller does not need to address devices on the bus before sending a universal command. However, keep in mind that by design, some devices in a system may not recognize and respond to every universal command.

The universal commands and the corresponding decimal codes are:

LOCAL LOCKOUT	LLO	17
DEVICE CLEAR	DCL	20
SERIAL POLL ENABLE	SPE	24
SERIAL POLL DISABLE	SPD	25

Local Lockout (LLO)

Many devices that operate on the interface bus have a front panel LOCAL/RESET button. This button allows a system operator to regain local control of a device that is under remote control. LOCAL LOCKOUT disables the local/reset button so that the unit will remain under remote control even if the button is pushed. Thus LLO protects devices on the bus from accidental returns to local control during system operation.

To execute the local lockout command use the statements

```
10 CMD "?U"
20 FORMAT 3B
30 OUTPUT (13,20)256,17,512;
```

(17 = Local Lockout)

Local lockout is not affected by the Interface Clear. To regain local control of bus devices after sending a local lockout command, use the Set Local statements described in Chapter 3 (p. 3-7) or, for devices that respond to it, the Go To Local bus command.

Device Clear (DCL)

The DEVICE CLEAR command returns each responding device to a pre-defined state. For any device which obeys the device clear command, its operating manual will describe the effect of the command. The response of each device will be appropriate to that device. For example, a signal generator may stop its signal output; a remotely operated switch may open. To execute the device clear command use

```
10 CMD "?U"
20 FORMAT 3B
30 OUTPUT (13,20)256,20,512;
```

(20 = Device Clear)

See the Selected Device Clear (p. 4-7) for a means of clearing only addressed devices.

Serial Poll Enable (SPE)

The calculator uses SERIAL POLL ENABLE to initiate the serial poll procedures. The next chapter describes serial polling in detail. To issue SPE use the statements

```
10 CMD "?U"
20 FORMAT 3B
30 OUTPUT (13,20)256,24,512;
```

(24 = Serial Poll Enable)

The Serial Poll Disable command is normally used to end a serial poll. However, Interface Clear (pushing the calculator STOP button) also terminates the serial poll mode.

Serial Poll Disable (SPD)

The SERIAL POLL DISABLE command ends a serial poll. All devices which respond to SPE will also respond to SPD. Use the statements

```
10 CMD "?U"
20 FORMAT 3B
30 OUTPUT (13,20)256,25,512;
```

(25 — Serial Poll Disable)

ADDRESSED COMMANDS

The addressed commands affect only devices that have been addressed to listen. Thus the addressed commands provide a way to send a command to a selected group of devices on the bus. The calculator must address the devices to listen before transmitting an addressed command.* To transmit an addressed command use the following statements:

(? — Unlisten command; U — Talk address (calculator))

```
10 CMD "?U <Listen Addresses> "
20 FORMAT 3B
30 OUTPUT (13,20) 256,C,512;
```

(C — Addressed command code)

These statements are similar to the basic sequence for issuing bus commands with the addition of the appropriate listen addresses in the CMD statement.

*The specifications for the Hewlett-Packard Interface Bus include one addressed command that can only be obeyed by a device that is addressed to talk. However, this command is not used with the 9830A Calculator.

The addressed commands and the decimal codes are:

GO TO LOCAL	GTL	1
SELECTED DEVICE CLEAR	SDC	4
GROUP EXECUTE TRIGGER	GET	8

Go To Local (GTL)

The GO TO LOCAL command provides a convenient way to return control of selected devices to the system operator. GTL allows the operator to perform tasks that cannot be done solely under remote control. Suppose that two devices on the bus have listen addresses "8" and "9" and both obey GTL. Then the statements

Unlisten command

Talk address (Calculator)

Listen addresses

```
10 CMD "?U89"

20 FORMAT 3B

30 OUTPUT (13,20)256,1,512;
```

Go to local

will cause both devices to go to local control.

If a device is in the Local Lockout state when it receives GTL, the device will remain in the LLO state. The device will operate according to its manual controls, but the local/reset pushbutton remains disabled. Thus when later calculator program statements return the device to remote control, LLO will still be in effect.

Selected Device Clear (SDC)

The SELECTED DEVICE CLEAR is similar to the DEVICE CLEAR universal command. The difference is that devices will obey SDC only if they are addressed to listen. Suppose that two devices on the bus have the listen addresses "8" and "9" and that both obey SDC. Then the statements to clear both devices are

Unlisten command
Talk address (Calculator)
Listen addresses

```
10 CMD "?U89"
20 FORMAT 3B
30 OUTPUT (13,20)256,4,512;
```

Selected device clear

Group Execute Trigger (GET)

GROUP EXECUTE TRIGGER can start a simultaneous action by several devices on the bus. For example, GET may initiate a simultaneous measurement by a voltmeter and a counter to check a VCO characteristic. The action that GET initiates depends on the design of each responding device. Check the operating manual for a device to see whether it obeys GET, and if so, what its response to the command will be. Let "8" and "9" be the listen addresses of two devices that obey GET. Then the statements

Unlisten command
Talk address (Calculator)
Listen addresses

```
10 CMD "?U89"
20 FORMAT 3B
30 OUTPUT (13,20)256,8,512;
```

Group execute trigger

will trigger both devices.

Chapter 5
SERVICE REQUESTS AND SERIAL POLLING

Both service requests and serial polling provide an additional means of communications between the calculator (controller) and other devices on the bus. A device may use a service request to ask for the attention of the calculator. The calculator may use serial polling to find out the status or condition of a device on the bus. Typically, the calculator uses serial polling to locate the source of a service request, and the cause. However, serial polling is not limited to situations involving service requests.

Every bus compatible instrument that is designed to use the service request should also respond to a serial poll. However, a device can be designed to respond to serial polling even though it does not use service request. The operating manual for each device will describe whether the device uses service request and how the device responds to a serial poll.

All of the operations described in this chapter depend heavily on the specific characteristics of the devices in the system. The system programmer must use programming techniques that are appropriate for his own system.

SERVICE REQUESTS

Some devices that operate on the interface bus have the ability to request service from the system controller. A device may request service when it has completed a measurement, when it has detected a critical condition, or for any other reason. The operating manual for a device will tell whether it uses service request, and if so, for what purpose.

Service requests use one of the management lines of the interface bus. The 9830A Calculator can check the status of this line to see whether a service request is present. All devices on the interface bus use the same line to request service. Therefore, when the calculator detects a service request, one or more devices may be the source.

Status Checks

To check for the presence of a service request use the Status function (STAT) as follows:

⟨variable⟩ = STAT13

The number 13 is the select code of the Interface Bus I/O Card. This statement returns one of the values listed in Table 5-1. The status codes 0 and 1 both indicate that one or more devices on the bus have requested service. The most common use of STAT is within an IF statement to test bus status as follows:

```
20 IF (STAT13=0) OR (STAT13=1) THEN 50
```

This statement can be used to transfer to a section of the calculator program that carries out a serial poll.

Table 5-1. Bus Status Codes

STAT13=0	Service has been requested, I/O card is not ready to input a character to the calculator.
STAT13=1	Service has been requested, I/O card is enabled and has a character ready to input to calculator.
STAT13=2	Service not requested, I/O card not ready to input a character to the calculator.
STAT13=3	Service not requested, I/O card is enabled and has a character ready to input to the calculator.

Responding to Service Requests

The calculator can be programmed to identify the source of a request and to service the requesting device(s); or the calculator can completely ignore all service requests. However, in most cases a service request indicates that the calculator should take some action to sustain proper system operations. In addition, the calculator must "clear" the service request line so that new requests from the same device or from other devices can be detected. Serial polling can identify the source of a service request, reveal the cause, and clear the service request line.

SERIAL POLLING

A serial poll enables the calculator to learn the status or condition of devices on the bus. The serial poll is so named because the calculator polls devices one at a time in sequence rather than all at once. When polled, a device transmits a single byte of information to indicate the status of the device. For example, a status byte may indicate the device is overloaded (power supply), the device output has stabilized at a new level (signal generator), or the device has requested service (any of several types of instruments).

If a device is requesting service, it will stop the request when it is polled and reports its status to the calculator. Once the calculator has polled each device that has been requesting service, the request line will be clear (assuming no new requests are received). It is important for the calculator to clear the service request line so that it can detect the presence of new requests.

Starting a Serial Poll

The Serial Poll Enable bus command starts a serial poll. During a serial poll the calculator must be addressed to listen so that it can receive the status byte from each device that it polls. Therefore, when starting a serial poll, use the following sequence of statements:

Unlisten command

Talk address (Calculator)

```
10 CMD "?U"
20 FORMAT 5B
30 OUTPUT (13,20)256,95,53,24,512;
```

Serial poll enable

Calculator listen address, "5"

Untalk command

The CMD statement addresses the calculator to talk so that the calculator can execute the OUTPUT statement. The OUTPUT statement issues the untalk command, addresses the calculator to listen, then sets the serial poll mode.

A typical 9830A Calculator program might initiate a serial poll as follows:

```
10 CMD "?U"
20 IF STAT13 <= 1 THEN 210
30 REM CONTINUE PROGRAM
      .
      .
      .
210 CMD "?U"
220 FORMAT 5B
230 OUTPUT (13,220)256,95,53,24,512;
```

In this example, line 20 checks for the presence of a service request. If a service request is found, the program jumps to line 210 to set the serial poll mode.

Polling the Devices on the Bus

The serial poll enable command affects only the devices that have been designed to respond to a serial poll. When polling devices, include only devices that can respond. Do not include a device in a poll if the device does not respond to SPE.

To poll a device, simply address it to talk, then use the RBYTE (Read Byte) function to read the status byte the device will output. A typical poll would have the form

```
80   CMD " ⟨Talk Address⟩ "

90   ⟨variable⟩  = RBYTE13
```

The number 13 following RBYTE is the select code of the Interface Bus I/O Card. Line 90 will place the status byte in the calculator register designated by the variable. The status byte contains eight bits; Figure 5-1 lists the meaning of the bits.

The RBYTE function produces the decimal equivalent of the status byte. Thus if we use

```
A = STAT13
```

to poll a device, then the variable A will contain the decimal equivalent of the status byte for the device. Suppose the device has requested service and responds to the poll with a one in status bit 7 and zero in all other bits. Then we will find that

A=64

in this case. Other decimal values from 0 to 255 can be returned to the variable A depending on the device being polled and the state of the device.

Ending a Serial Poll

At the end of a serial poll the calculator resumes normal bus operations by issuing the serial poll disable bus command. SPD does not change the addressed talker and listeners. To end a serial poll use the statements

```
10 CMD "?U"
20 FORMAT 3B
30 OUTPUT (13,20)256,25,512;
```

- "?" — Unlisten command
- "U" — Talk address (Calculator)
- 256 — Serial poll disable

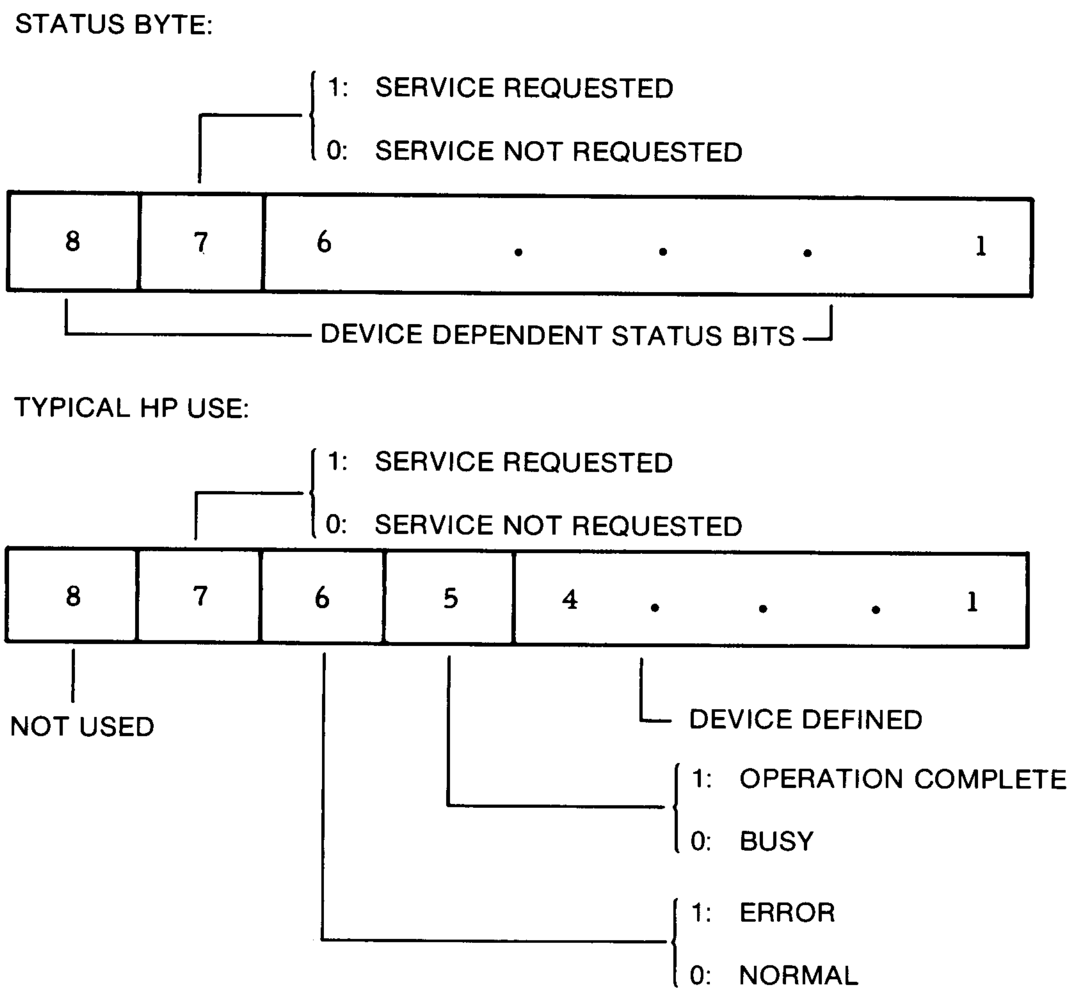


Figure 5-1. Device Poll Status Byte

The best programming methods for conducting a serial poll will vary from system to system, and from programmer to programmer. Major considerations in setting a polling procedure include

- How many devices must be polled?
- What action may be required after each device is polled?
- How quickly must the calculator take action after polling a device?
- What are the requirements for overall system speed of operation?

With some programming imagination it is possible to create a wide variety of polling strategies. Polling a single device is relatively simple; polling several devices can become quite involved. Some devices may need fast action from the calculator when they request service. Others may be able to wait. The relative need for service can set priorities in the polling sequence. Some systems may tolerate a simple but slow polling procedure; other systems may require a more involved but faster approach.

The system programmer bears the responsibility of creating and implementing his polling strategy. However, most strategies are variations of two basic approaches. Each approach has its own advantages for systems where several devices must be polled. The two techniques reduce to the same method where only a single device will be polled.

Strategy 1: Poll all devices, then act after the serial poll is complete.

This strategy is the simplest to implement. As the calculator polls each device, it stores the device status byte in a calculator variable or array (e.g., A (1) through A (n) for n devices). After it has polled each device, the calculator terminates the serial poll. Then the calculator examines each of the stored status bytes to determine what action it should take for each device it has polled.

Strategy 2: As each device is polled, take immediate action on the status byte.

This polling technique provides the fastest possible response to a device which has requested service. Speed may be important, for example, to a device which requests service to indicate an overload condition, especially if quick action by the calculator is necessary to prevent catastrophic failure of the device. With this polling strategy, the polling order depends on the relative priority for action by the calculator.

The trade-off for fast response to service requests is increased programming complexity. Each time the calculator interrupts the serial poll to service a device, the calculator must issue SPD to end the serial poll mode, take the appropriate action to service the device, then issue SPE again to resume the serial poll.

A PROGRAM EXAMPLE OF SERIAL POLLING

Assume that a bus system has two devices that can request service. The devices have talk addresses "X" and "Y". When polled both of these devices return a status byte with decimal value 64 if they have requested service and 0 otherwise. Then a 9830A Calculator program may conduct a serial poll as follows:

Sample Program Excerpt

```
60 IF STAT13 <= 1 THEN 230
70 REM CONTINUE IF SERVICE NOT REQUESTED
                    •
                    •
                    •
230 CMD "?U"
240 FORMAT 5B
250 OUTPUT (13,240)256,95,53,24,512;
260 CMD "X"
270 A[1]=RBYTE13
280 CMD "Y"
290 A[2]=RBYTE13
300 CMD "?U"
310 OUTPUT (13,240)256,25,512;
320 IF A[1]#64 THEN 340
330 GOSUB 530
340 IF A[2]#64 THEN 360
350 GOSUB 550
360 GOTO 70
                    •
                    •
                    •
530 DISP "SERVICE DEVICE 1"
540 RETURN
550 DISP "SERVICE DEVICE 2"
560 RETURN
570 END
```

Program Explanation

60	Check for service request.
230–250	Set serial poll mode.
260–270	Poll 1st device; store status byte in A(1).
280–290	Poll 2nd device; store status byte in A(2).
300–310	End serial poll.
320	Check if 1st device requested service.
340	Check if 2nd device requested service.

You will have to modify the techniques illustrated here to fit the devices in your system. Keep in mind that serial polling is not limited to situations involving service requests, but rather it depends entirely on the devices in the system. The status byte that a device transmits may be easy to interpret, or it may require some programming effort to look at the status bits. Finally, you must program the calculator to act on the status information in a way that is correct for the device.

Appendix A
FOR THE ADVANCED PROGRAMMER

Information transfer on the Hewlett-Packard Interface Bus follows a strict protocol. A more thorough understanding of the information transfer process can be helpful in many situations, particularly for diagnosing system operations.

The calculator can transmit messages with the bus in either of two modes. Acting as the system controller, the calculator can transmit messages with the bus in the command mode. The calculator can also set the bus to the data mode to send messages. To describe the role of the command mode and data mode it is first necessary to look again at the BUS COMMAND statement.

BUS COMMAND Statement (CMD)

The typical form for the BUS COMMAND statement is

CMD "<Command String>"[,"<Data String>"]

where the portion within brackets is optional. Note that for the second parameter the term "data string" has been substituted for the term "program code string". A program code is merely one type of data that may be transmitted on the bus, and is transmitted just like any other data character. Therefore, the second parameter is not restricted to program codes.

The calculator transmits the characters of the command string with the bus in the command mode; it transmits the characters of the data string with the bus in the data mode. The calculator automatically sets the proper bus mode as it encounters each command string and data string following CMD. For example, the calculator sets the bus mode as follows in this simple statement:

Set command mode

Transmit command string

Set data mode

Transmit data string

```
10 CMD "?U6";"E"
```

To send the data string the calculator must be addressed to talk. If the calculator is not addressed to talk it cannot complete the execution of the CMD statement. The calculator only sends the characters that appear in each string. It does not send a carriage return or line feed to mark the end of a string. When the execution of the CMD statement is complete, the bus will be in the data mode.

It is possible to explicitly set the bus to either the command mode or data mode. Then the OUTPUT statement can be used to send command or data messages.

The Command Mode

To place the interface bus in the command mode use the statements

```
10 CMD "?U"

20 FORMAT B

30 OUTPUT (13,20)256;
```

Code to set command mode

Once this statement is executed, the bus remains in the command mode until the calculator executes a CMD statement or until it executes the statement described below to explicitly place the bus in the data mode.

Only the controller (calculator) can send messages when the bus is in the command mode. All other devices on the bus will interpret each message as command information (addresses and bus commands). The FORMAT B and OUTPUT statements are particularly useful for transmitting specific bytes of information, such as bus commands or other codes listed in Appendix B.

The Data Mode

To return the bus to the data mode execute the statement

Code to set data mode

```
40 OUTPUT (13,20)512;
```

Format statement number

or any CMD statement, such as

```
10 CMD ""
```

When the bus returns to the data mode the addressed talker can send data to the addressed listeners. Data messages include programming information, measurement information, or any other alphanumeric message sent by the talker to the listener when the bus is in the data mode.

Appendix B

ASCII CHARACTER CODES

ASCII CHARACTER	OCTAL CODE	DECIMAL CODE	ASCII CHARACTER	OCTAL CODE	DECIMAL CODE
NUL	00	0	SP	40	32
SOH	01	1	!	41	33
STX	02	2	‘	42	34
ETX	03	3	#	43	35
ETO	04	4	$	44	36
ENQ	05	5	%	45	37
ACK	06	6	&	46	38
BEL	07	7	’	47	39
BS	10	8	(	50	40
HT	11	9	)	51	41
LF	12	10	*	52	42
VT	13	11	+	53	43
FF	14	12	,	54	44
CR	15	13	—	55	45
SO	16	14	.	56	46
SI	17	15	/	57	47
DLE	20	16	Ø	60	48
DC1	21	17	1	61	49
DC2	22	18	2	62	50
DC3	23	19	3	63	51
DC4	24	20	4	64	52
NAK	25	21	5	65	53
SYN	26	22	6	66	54
ETB	27	23	7	67	55
CAN	30	24	8	70	56
EM	31	25	9	71	57
SUB	32	26	:	72	58
ESC	33	27	;	73	59
FS	34	28	<	74	60
GS	35	29	=	75	61
RS	36	30	>	76	62
US	37	31	?	77	63

ASCII CHARACTER	OCTAL CODE	DECIMAL CODE	ASCII CHARACTER	OCTAL CODE	DECIMAL CODE
@	100	64	(Apost.)	140	96
A	101	65	a	141	97
B	102	66	b	142	98
C	103	67	c	143	99
D	104	68	d	144	100
E	105	69	e	145	101
F	106	70	f	146	102
G	107	71	g	147	103
H	110	72	h	150	104
I	111	73	i	151	105
J	112	74	j	152	106
K	113	75	k	153	107
L	114	76	l	154	108
M	115	77	m	155	109
N	116	78	n	156	110
O	117	79	o	157	111
P	120	80	p	160	112
Q	121	81	q	161	113
R	122	82	r	162	114
S	123	83	s	163	115
T	124	84	t	164	116
U	125	85	u	165	117
V	126	86	v	166	118
W	127	87	w	167	119
X	130	88	x	170	120
Y	131	89	y	171	121
Z	132	90	z	172	122
[	133	91	{	173	123
\	134	92	¦	174	124
]	135	93	}	175	125
⌒	136	94	~	176	126
—	137	95	DEL	177	127

QUICK REFERENCE GUIDE

For systems based on the Hewlett-Packard Interface Bus and HP 9830A Calculator

BASIC PROGRAMMING

	OPERATION	DESCRIPTION	GENERAL FORM AND EXAMPLES
REMOTE LOCAL	Remote Enable	Enables remote operation of devices on the bus.	10 CMD "?U" 20 FORMAT B 30 OUTPUT (13,20)768;
	Set Local	Returns all devices to local control.	100 CMD "?U" 110 FORMAT B 120 OUTPUT (13,110)1024;
PROGRAM-MING INSTRU.	Using literals as program codes	Calculator sends program codes to control listening device.	10 CMD "?U ⟨Listen Address⟩"," ⟨Program Code String⟩ " 10 CMD "?U$","L1000.0="
	Using variable values as program codes	Calculator transmits program code string containing variable values to control listening device.	10 CMD "?U ⟨Listen Address⟩ " 20 FORMAT ⟨format list⟩ 30 OUTPUT (13,20) ⟨variable list⟩ 10 CMD "?U$" 20 FORMAT "L",F1000.1,"=" 30 OUTPUT (13,20)A
SENDING AND RECEIVING DATA	Instrument-to-Calculator	The calculator receives data from some other device on the bus.	10 CMD "? ⟨Talk address⟩ 5" 20 FORMAT ⟨format list⟩ 30 ENTER (13,20) ⟨variable list⟩ 10 CMD "?V5" 20 FORMAT 4X,E12.0 30 ENTER (13,20)A
	Calculator-to-Instrument	The calculator sends data to other devices on the bus.	10 CMD "?U ⟨Listen Addresses⟩ 20 FORMAT ⟨format list⟩ 30 OUTPUT (13,20) ⟨variable list⟩ 10 CMD "?U2*" 20 FORMAT F7.2 30 OUTPUT (13,20)A
	Instrument-to-Instrument	The calculator commands one device to send data to other devices.	10 CMD "? ⟨Talk Address⟩ ⟨Listen Addresses⟩ " 10 CMD "?V1"
	Interface Clear	Halts all I/O operations on the bus; unaddresses talker and listeners.	Press STOP key.

BUS COMMANDS

	COMMAND		DECI-MAL CODE	DESCRIPTION	GENERAL FORM AND EXAMPLES
UNADDRESS COMMANDS	UNL	UNLISTEN	63	Clears bus of all listeners.	10 CMD "?"
	UNT	UNTALK	95	Unaddresses the current talker so that no talker remains on the bus.	
UNIVERSAL COMMANDS	LLO	LOCAL LOCKOUT	17	Disables front panel local-reset button on responding devices.	10 CMD "?U" 20 FORMAT 3B 30 OUTPUT (13,20)256,C,512; (C = command code) 10 CMD "?U" 20 FORMAT 3B 30 OUTPUT (13,20)256,17,512; (17 = Local Lockout)
	DCL	DEVICE CLEAR	20	Returns all responding devices to predetermined states.	
	SPE	SERIAL POLL ENABLE	24	Sets serial poll mode.	
	SPD	SERIAL POLL DISABLE	25	Terminates serial poll mode.	
ADDRESSED COMMANDS	GTL	GO TO LOCAL	1	Returns to local control all responding devices that are addressed to listen.	10 CMD "?U ⟨Listen Addresses⟩ " 20 FORMAT 3B 30 OUTPUT (13,20)256,C,512; (C = command code) 10 CMD "?U89" 20 FORMAT 3B 30 OUTPUT (13,20)256,8,512; (8 = Group Execute Trigger)
	SDC	SELECTED DEVICE CLEAR	4	Returns to predetermined states all responding devices that are addressed to listen.	
	GET	GROUP EXECUTE TRIGGER	8	Initiates a preprogrammed action by all responding devices that are addressed to listen.	

SERVICE REQUESTS AND SERIAL POLLING

	OPERATION	DESCRIPTION	GENERAL FORM AND EXAMPLES
SERVICE REQUESTS	Checking Bus Status	Determines whether service has been requested by any devices.	10 ⟨variable⟩ = STAT13 10 A=STAT13
SERIAL POLLING	Starting a Poll	Set the serial poll mode; also use untalk command and address calculator to listen.	10 CMD "?U" 20 FORMAT 5B 30 OUTPUT (13,20)256,95,53,24,512;
	Polling a Device	The polled device sends a status byte indicating the device condition. The variable will contain the decimal equivalent of the byte.	40 CMD " ⟨Talk Address⟩ " 50 ⟨variable⟩ = RBYTE13 40 CMD "X" 50 A[1]=RBYTE13
	Ending a Poll	Serial Poll Disable terminates the serial poll mode.	60 CMD "?U" 70 FORMAT 3B 80 OUTPUT (13,20)256,25,512;

hewlett·packard interface bus users guide

9830A

HP STOCK NO. 59300-90002
HP MICROFICHE NO. 59300-90004

HEWLETT hp PACKARD

PRINTED IN U.S.A.