

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
Copyright, 1974

NOVEMBER 1974

[illegible]

CONTENTS

CHAPTER 1, AN OVERVIEW OF THE INTERFACE BUS	1-1
Major Bus System Capabilities	1-1
Interface Bus Operation	1-2
Talkers, Listeners and Controller	1-2
Addressing	1-4
Data Transfer	1-4
CHAPTER 2, ASSEMBLING THE SYSTEM	2-1
Equipment	2-1
Setting Addresses	2-2
Talk Addresses and Listen Addresses	2-3
Calculator Talk and Listen Addresses	2-5
Setting Other Device Talk and Listen Addresses	2-5
Multiple Addresses	2-5
Addressable Mode	2-6
Address Assignment Table	2-6
Reserved Addresses	2-6
A Review of Addresses	2-7
Connecting Bus Cables	2-7
Bus Cables	2-8
System Configuration	2-8
Cable Length Restrictions	2-8

(Continued)

CONTENTS (Continued)

CHAPTER 3, PROGRAMMING SYSTEM OPERATIONS	3-1
The Bus Command Statement	3-1
Addressing Talkers and Listeners	3-2
Remote and Local Operation	3-6
Programming Instruments	3-8
Program Code Sets	3-8
Using Literals as Program Codes	3-10
Using Variable Values as Program Codes	3-11
Sending and Receiving Data	3-15
Instrument-to-Calculator	3-15
Calculator-to-Instrument	3-19
Instrument-to-Instrument	3-20
Interface Clear	3-22
A Programming Example	3-22
 CHAPTER 4, BUS COMMANDS	4-1
Issuing Bus Commands	4-2
Unaddress Commands	4-3
Unlisten (UNL)	4-3
Untalk (UNT)	4-4
Universal Commands	4-4
Local Lockout (LLO)	4-4
Device Clear (DCL)	4-5
Serial Poll Enable (SPE)	4-5
Serial Poll Disable (SPD)	4-6
Addressed Commands	4-6
Go To Local (GTL)	4-7
Selected Device Clear (SDC)	4-7
Group Execute Trigger (GET)	4-8
 CHAPTER 5, SERVICE REQUESTS AND SERIAL POLLING	5-1
Service Requests	5-1
Status Checks	5-2
Responding to Service Requests	5-3
Serial Polling	5-3
Starting a Serial Poll	5-3
Polling the Devices on the Bus	5-5
Ending a Serial Poll	5-5
Polling Strategies	5-7
A Program Example of Serial Polling	5-8
 APPENDIX A, FOR THE ADVANCED PROGRAMMER	A-1
Bus Command Statement (CMD)	A-1
The Command Mode	A-2
The Data Mode	A-3
 APPENDIX B, ASCII CHARACTER CODES	B-1

FIGURES

1-1. Instrument System Using the HP 9830A Calculator and HP-IB	1-3
2-1. System Equipment	2-2
2-2. Installing the 11272B Extended I/O ROM	2-2
2-3. Installing the Bus Interface Card	2-3
2-4. Address Switches	2-5
2-5. Addressable Mode Switch	2-6
2-6. Bus Cable	2-8
2-7. Connecting Bus Cables	2-8
2-8. Front and Rear Views of an Assembled System	2-9
3-1. VCO Test System Configuration	3-23
5-1. Device Poll Status Byte	5-6

TABLES

2-1. Allowable Address Codes	2-4
2-2. Sample Address Assignment Table	2-7
3-1. Program Code Set Excerpts	3-8
3-2. Address Assignment Table for VCO Test System	3-23
4-1. Summary of Bus Commands	4-1
5-1. Bus Status Codes	5-2

Chapter 1

[illegible]

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 diagram illustrates the Hewlett-Packard Interface Bus architecture. On the left, under the heading "BUS COMPATIBLE DEVICES", are four boxes labeled "DEVICE 1", "DEVICE 2", "DEVICE 3", and "DEVICE 14" (with three dots between 3 and 14). On the right, under the heading "CALCULATOR PERIPHERALS", are three boxes labeled "PERIPHERAL I", "PERIPHERAL II", and "PERIPHERAL III". In the center is a drawing of a "CALCULATOR". At the bottom, a horizontal line represents the "HEWLETT-PACKARD INTERFACE BUS". Arrows show the connection: the bus line enters from the bottom left, goes up, and then branches into four vertical lines. Each vertical line has an arrow pointing to one of the four bus-compatible devices. Another branch from the main vertical line goes to the right, where it splits into three lines, each with an arrow pointing to one of the calculator peripherals. A fourth arrow points from the bus line to the top of the calculator, labeled "TO CALCULATOR I/O SLOTS".

1-3

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

[illegible]

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.

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.

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.

Bit Position:	b ₇	b ₆	b ₅	b ₄	b ₃	b ₂	b ₁
Talk Address:	1	0	A ₅	A ₄	A ₃	A ₂	A ₁
Listen Address:	0	1					

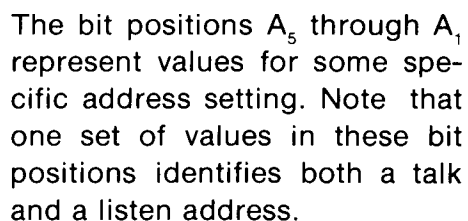


Figure 2-3. Installing the Bus Interface Card

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	0
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	(>

Setting Other Device Talk and Listen Addresses

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.

Multiple 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.

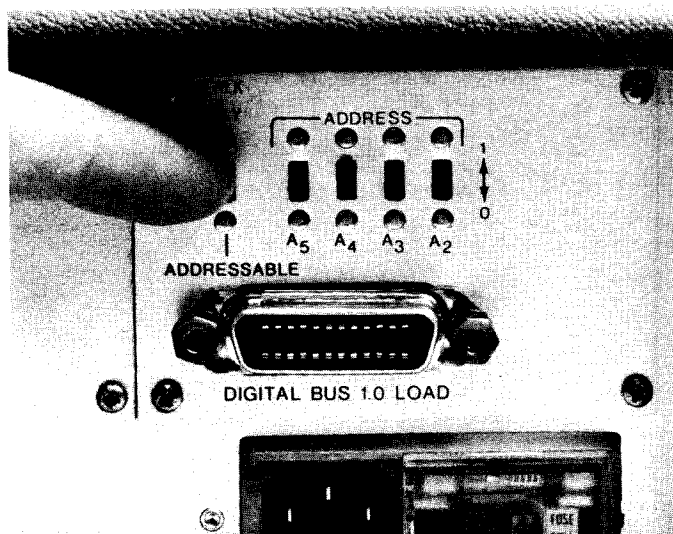


Figure 2-5. Addressable Mode Switch

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	!

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

- ## CONNECTING BUS 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.

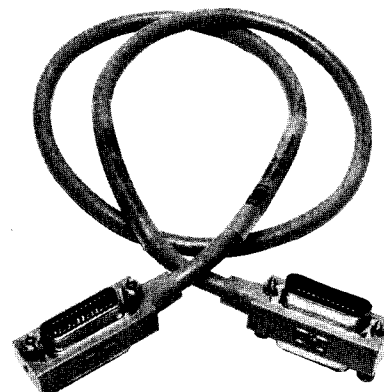


Figure 2-6. Bus Cable

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.

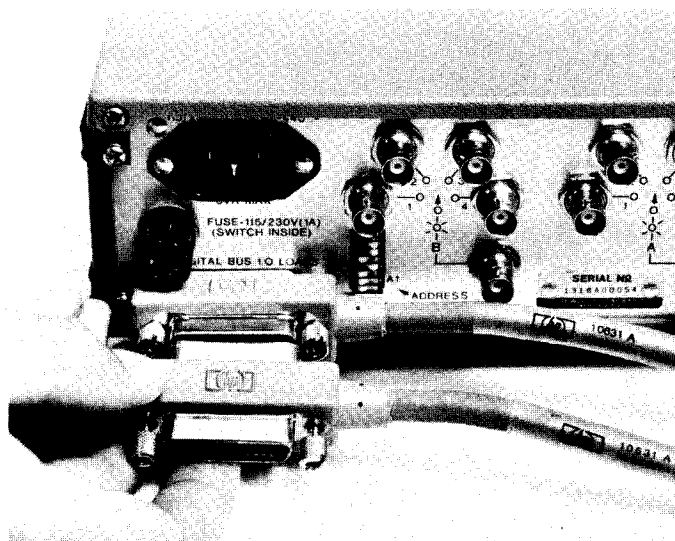


Figure 2-7. Connecting Bus Cables

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.

2-9

Chapter 3

[illegible]

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 mnemonic 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〉

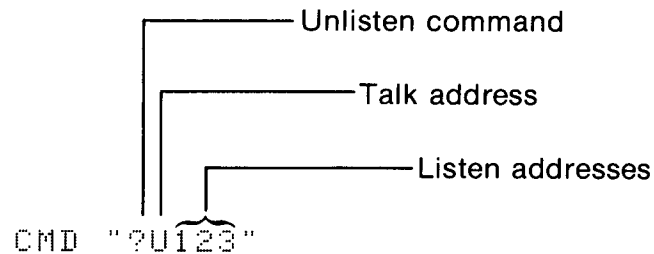
This sequence is made up of three major parts which serve the following purposes:

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.

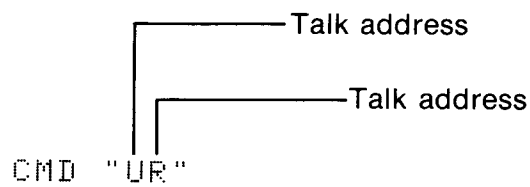
**When it addresses a device, the calculator "commands" the device to talk or listen—thus the term command string.

Example 2: Basic addressing sequence with multiple listeners:



Example 3: Successive talk addresses:

The statement



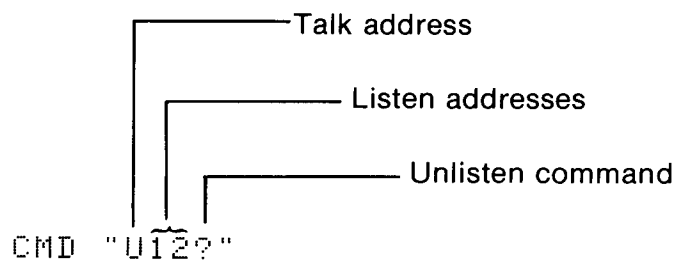
is equivalent to

`CMD "R"`

since the second talk address supercedes the first.

Example 4: An addressing sequence error:

The statement



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:

Unlisten command

New talk address

Repeat current listen addresses

CMD "2R12"

CMD "R" └── New talk address

Use the basic addressing sequence:

Unlisten command

Talk address

New listen addresses

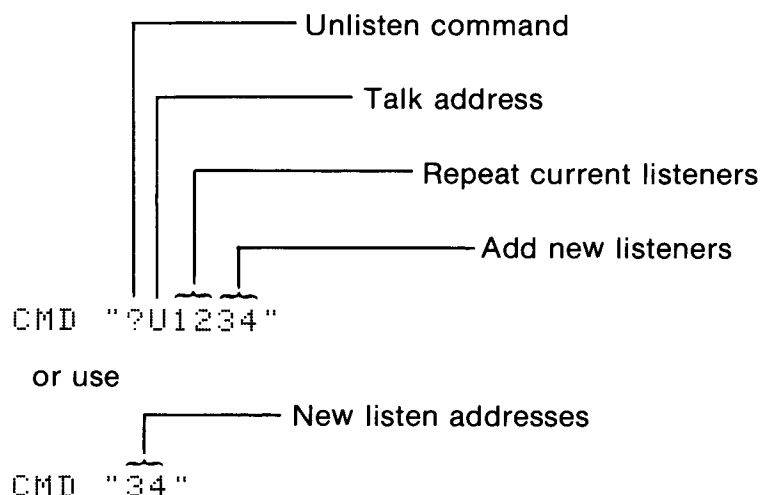
CMD "2U34"

[illegible]

3-5

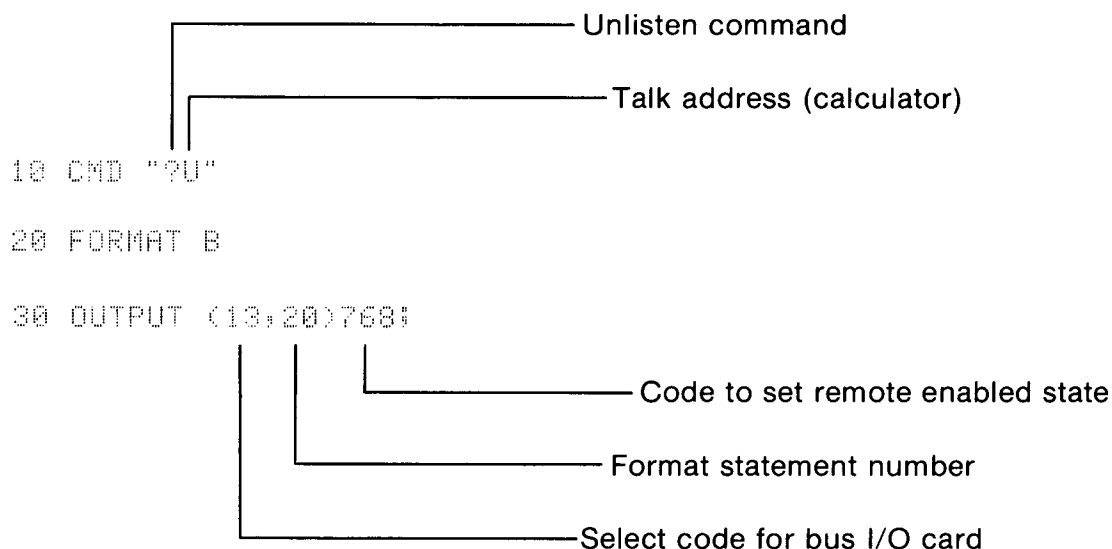
Example 7: Adding listeners:

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



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:



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

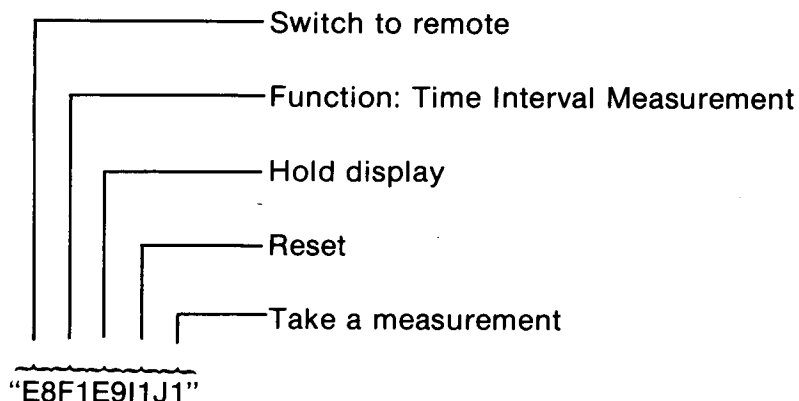
FREQ	L	AMPL	N
FREQ STEP	M	AMPL STEP	O
Hz	=	+dBm	;
kHz	>	-dBm	<
MHz	?		

(b) 3330B Automatic Synthesizer

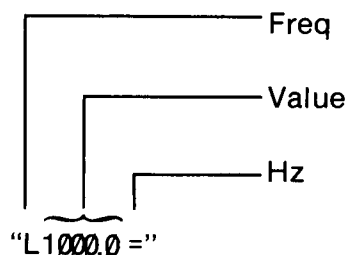
A program code string is merely a series of program codes that will be used to control the operation of a device. As with command strings, the characters of a program code string are transmitted and obeyed in the order written.

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):



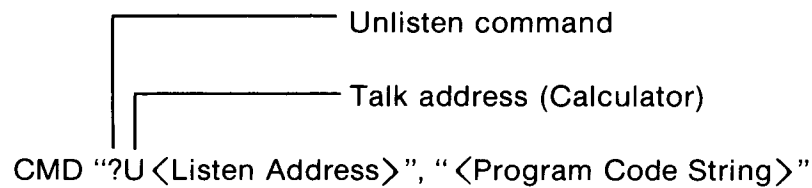
With the HP 3330B Automatic Synthesizer, a program code string to set an output frequency of 1000 Hz could appear as follows:



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.”

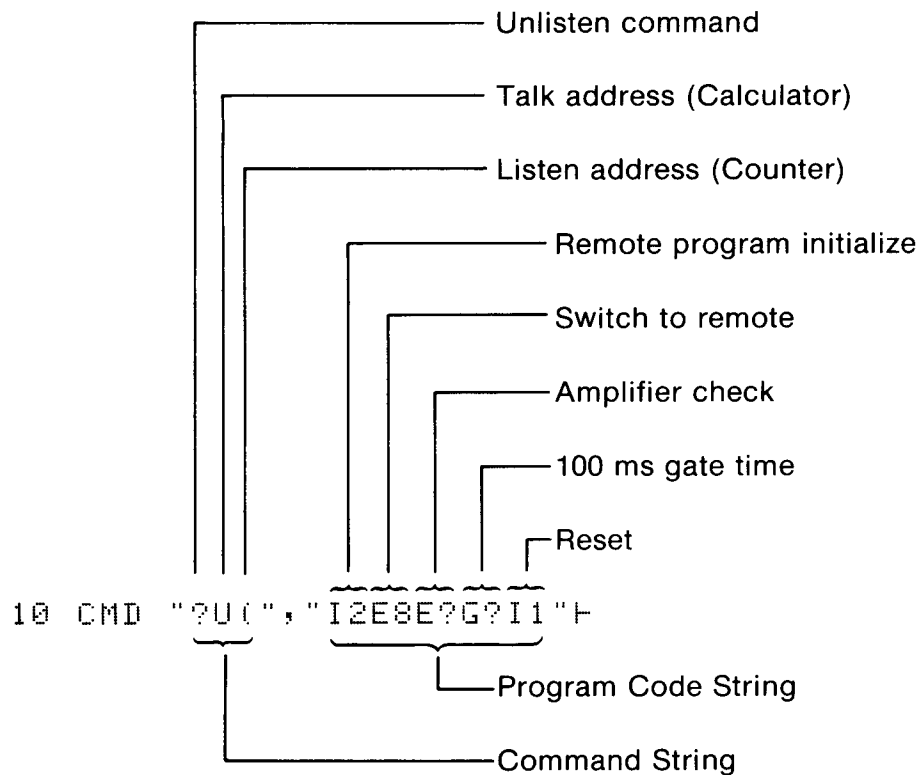
Using Literals as Program Codes

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

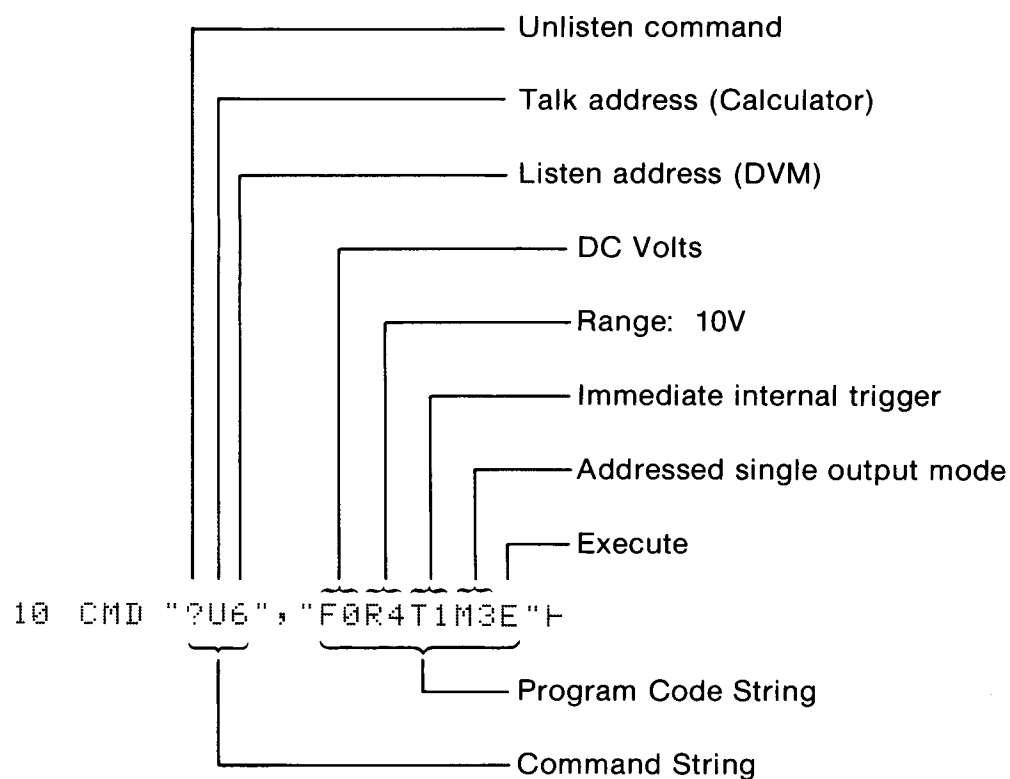


The following examples illustrate the use of literals for programming devices.

Example 1: 5345A Electronic Counter

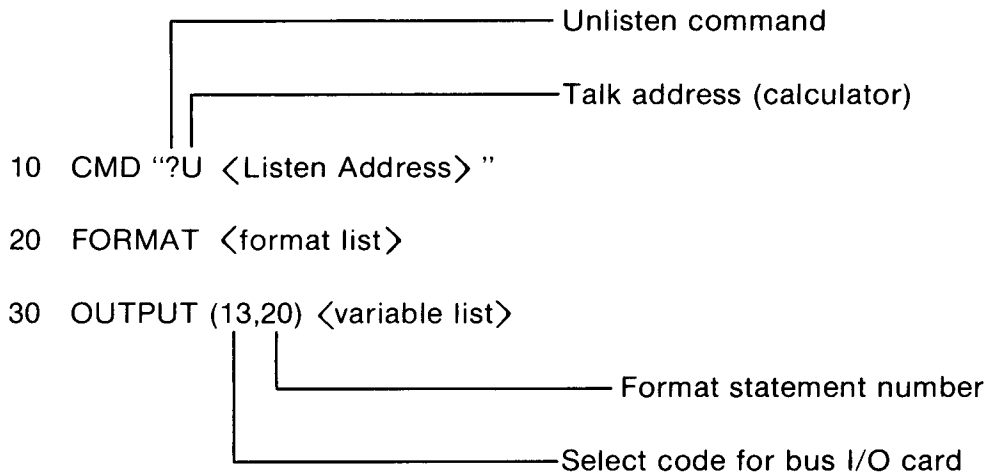


Example 2: 3490A Digital Voltmeter



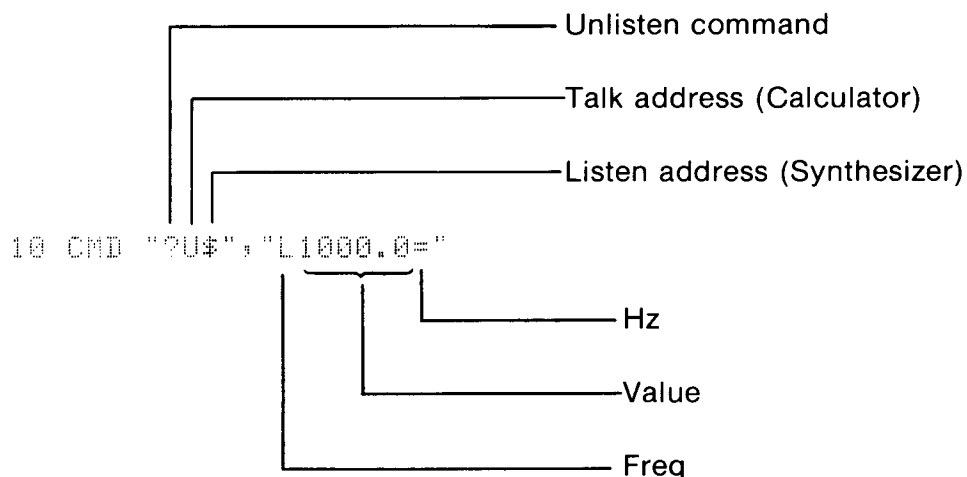
Using Variable Values as Program Codes

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.



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.

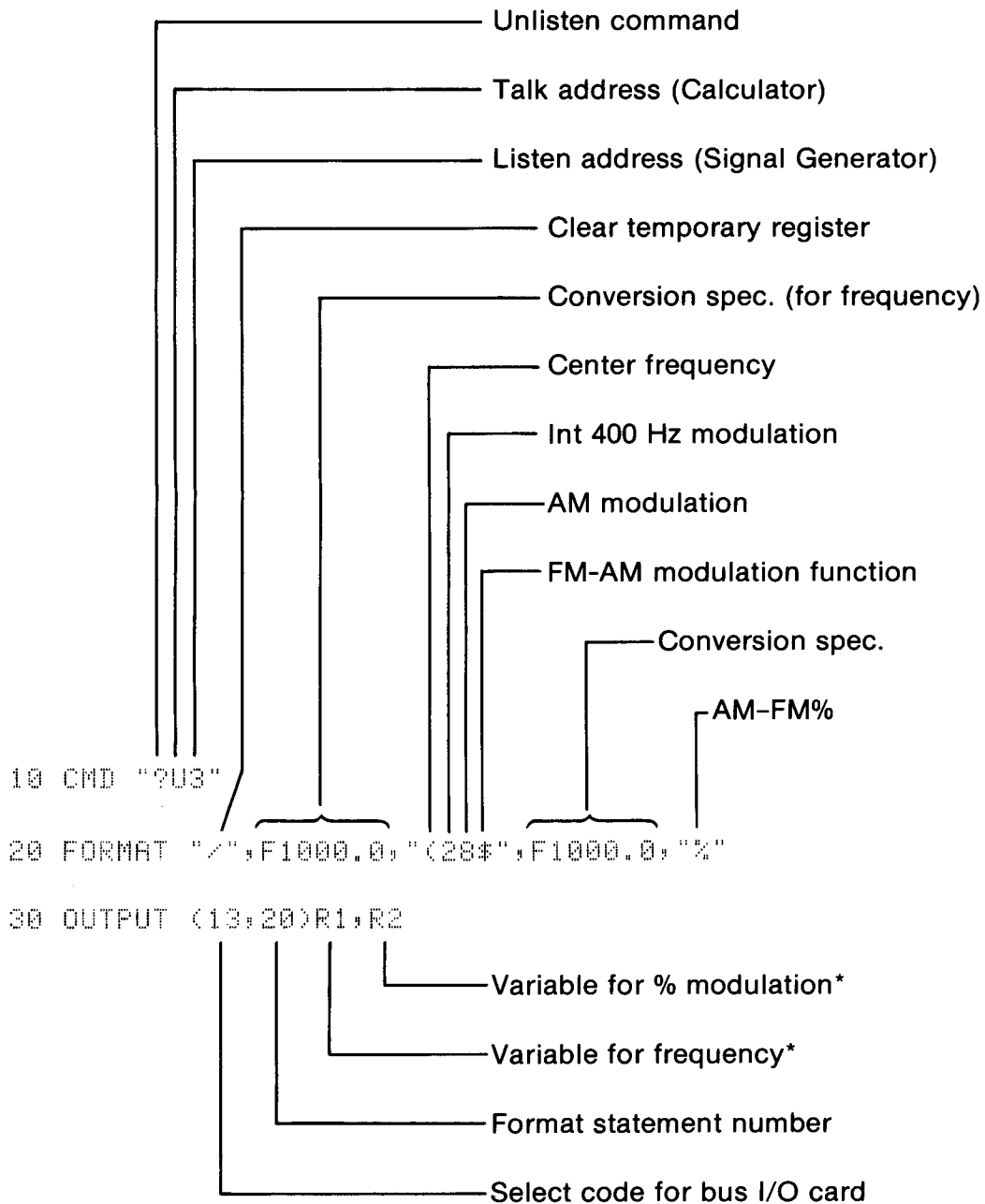
An example best illustrates using variable values as program codes. Suppose a system includes an HP 3330B Synthesizer with listen address "\$". To obtain a frequency output of 1000.0 Hz, we might use literals as program codes in the statement



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:

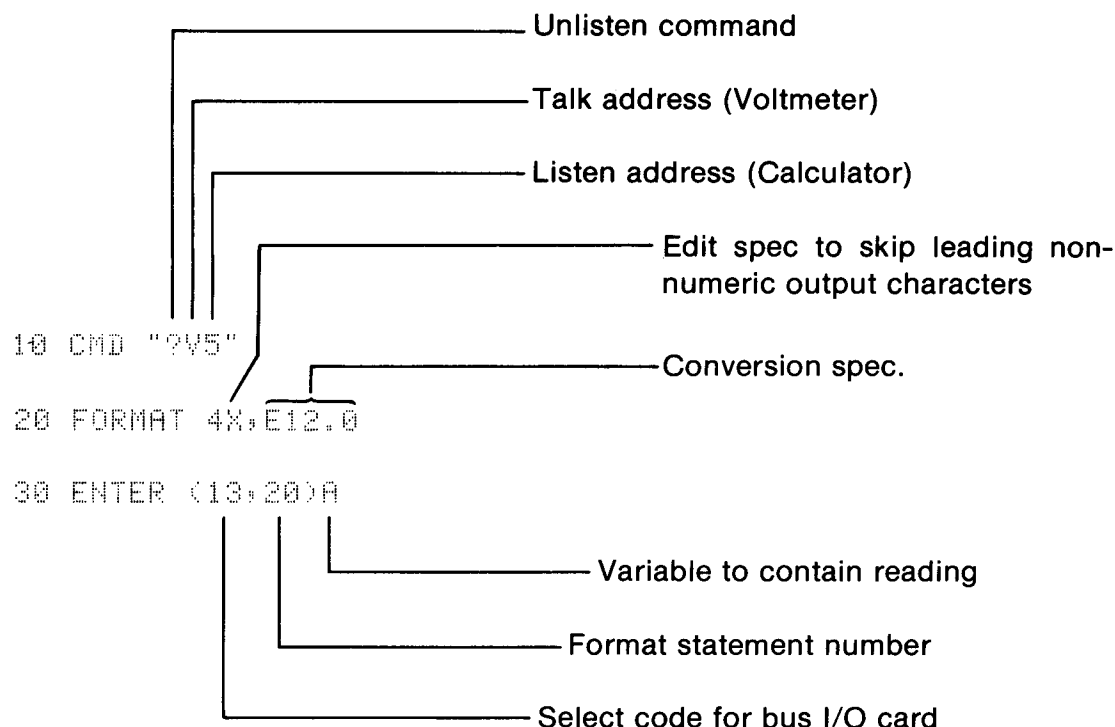
Here is another example of using variable values as program codes.

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.

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



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.

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

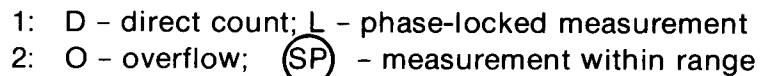


Diagram illustrating the program listing with annotations:

- Unlisten command
- Talk address (Counter)
- Listen address (Calculator)

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

- Variable to contain reading
- Variable to receive second character
- Variable to receive first character

Suppose the HP 5340A outputs the data string

L (SP) (SP) 0 8 1 6 4 0 4 7 E + 3 (CR) (LF)

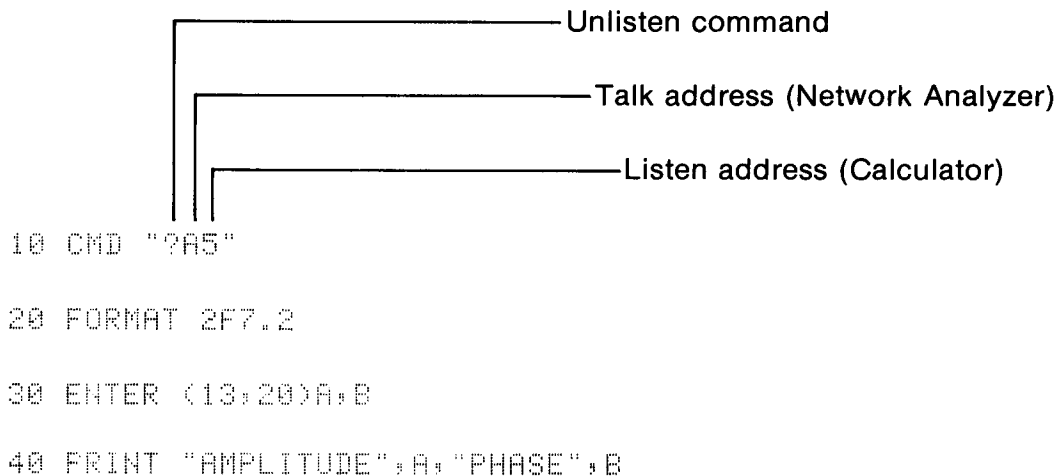
Variables A and B will receive the decimal equivalent for the first two characters of the data string. C will receive the value of the reading. In this case

A = 76 (decimal equivalent for "L")

B = 32 (decimal equivalent for "SP")

C = 8.164047E09 (value of counter's measurement)

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



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

- 0 3 4 . 4 8 , - 0 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

The first display will show

123.45

But the second display will show

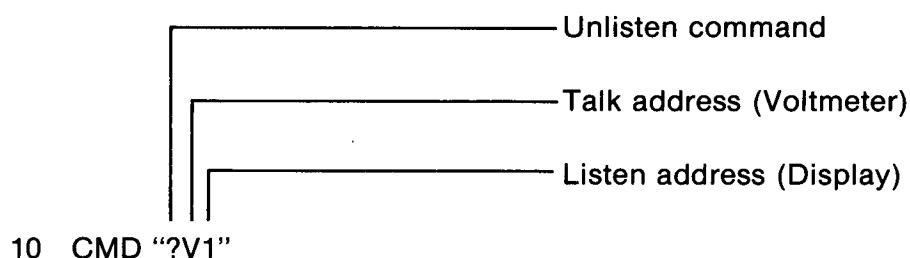
1.2345E 02

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



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

N (SP) D C + 0 8 3 2 5 0 E - 4 (CR) (LF)

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

083250E-4

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).

```
graph TD; A[HP 9830A CALCULATOR] --- B[BUS CABLES]; B --> C[HP 59303A D-A CONVERTER]; B --> D[HP 5345A ELECTRONIC COUNTER]; C --> E[VCO UNDER TEST]; E --> D;
```

The diagram illustrates the system configuration for testing a VCO. At the top is the HP 9830A CALCULATOR. A vertical line connects it to a horizontal line labeled BUS CABLES. This bus cable branches into two paths: one leading to the HP 59303A D-A CONVERTER on the left, and another leading to the HP 5345A ELECTRONIC COUNTER on the right. Below the D-A converter is the VCO UNDER TEST, which is connected to the electronic counter by a horizontal line. Arrows indicate the signal flow from the calculator to the D-A converter and counter, and from the D-A converter to the VCO, and from the VCO to the counter.

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)

Chapter 3, PROGRAMMING SYSTEM OPERATIONS □□□□□□□□□□□□□□□□

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.

oo

Sample Program

```
5 DIM R[11]
10 CMD "?U"
20 FORMAT B
30 OUTPUT (13,20)768;
40 CMD "?U+","E0"
50 CMD "?U(","I2E8E911"
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 $\pm 10V$ 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 [11].
- 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

[illegible]

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.

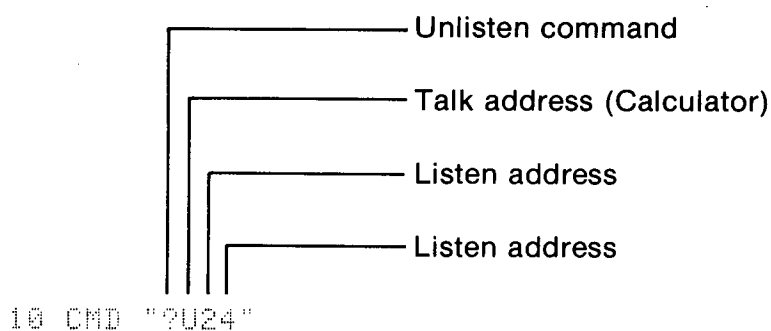
Table 4-1. Summary of Bus Commands

COMMAND		DECIMAL CODE	DESCRIPTION	GENERAL FORM AND EXAMPLES
UNADDRESS COMMANDS	UNL UNLISTEN	63	Clears bus of all listeners.	10 CMD "0"
	UNT UNTALK	96	Unaddresses the current talker so that no talker remains on the bus.	10 CMD "0"
UNIVERSAL COMMANDS	LLO LOCAL LOCKOUT	17	Disables front panel local-reset button on responding devices.	10 CMD "0"
	DCL DEVICE CLEAR	20	Returns all responding devices to predetermined states.	20 FORMAT 3B
	SPE SERIAL POLL ENABLE	24	Sets serial poll mode.	30 OUTPUT (13,20)256,C,512;
	SPD SERIAL POLL DISABLE	25	Terminates serial poll mode.	30 OUTPUT (13,20)256,C,512;
				command code Local Lockout
ADDRESSED COMMANDS	GTL GO TO LOCAL	1	Returns to local control all responding devices that are addressed to listen.	10 CMD "0" <Listen Addresses>
	SDC SELECTED DEVICE CLEAR	4	Returns to predetermined states all responding devices that are addressed to listen.	20 FORMAT 3B
	GET GROUP EXECUTE TRIGGER	8	Initiates a preprogrammed action by all responding devices that are addressed to listen.	30 OUTPUT (13,20)256,C,512;
			command code	Group Execute Trigger

UNADDRESS COMMANDS

UNLISTEN	UNL	63
UNTALK	UNT	95

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.



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.

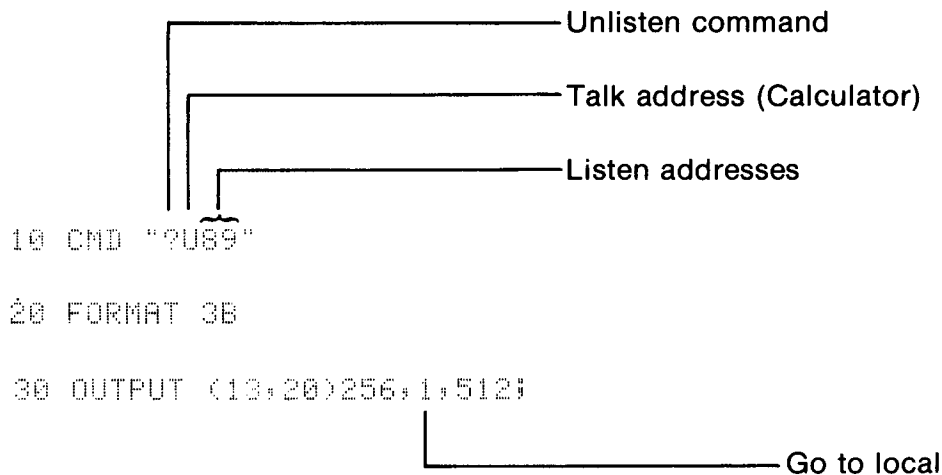
oo

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



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

[illegible]

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.

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.

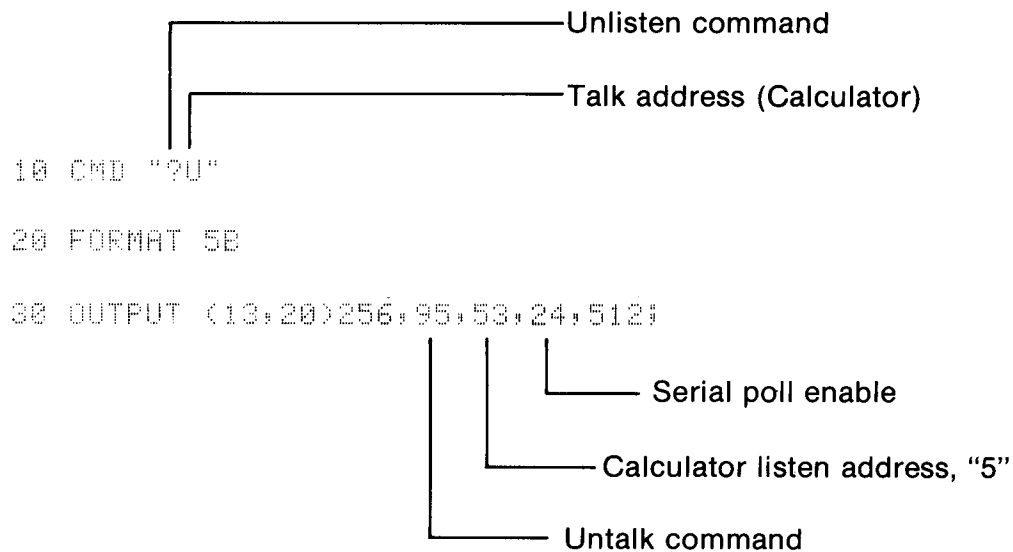
Responding to Service Requests

SERIAL POLLING

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

5-3



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

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

A = STAT13

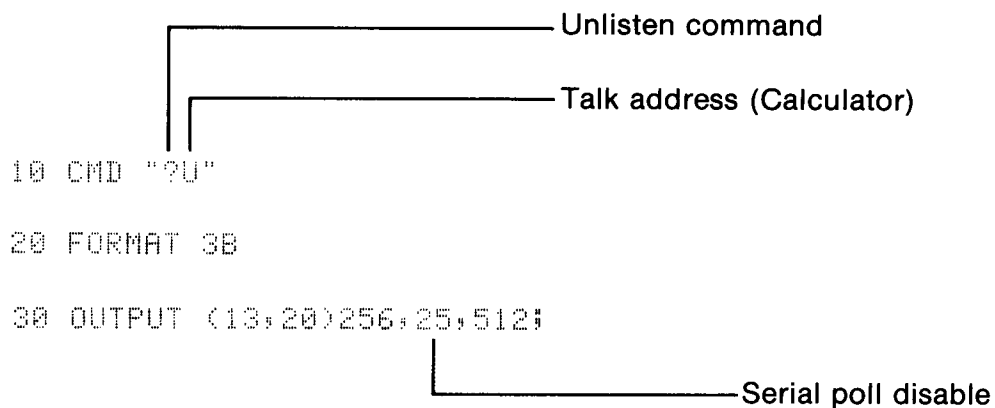
to poll a device, then the variable A will contain the decimal equivalent of the status byte for the device. Suppose the device has requested service and responds to the poll with a one in status bit 7 and zero in all other bits. Then we will find that

A=64

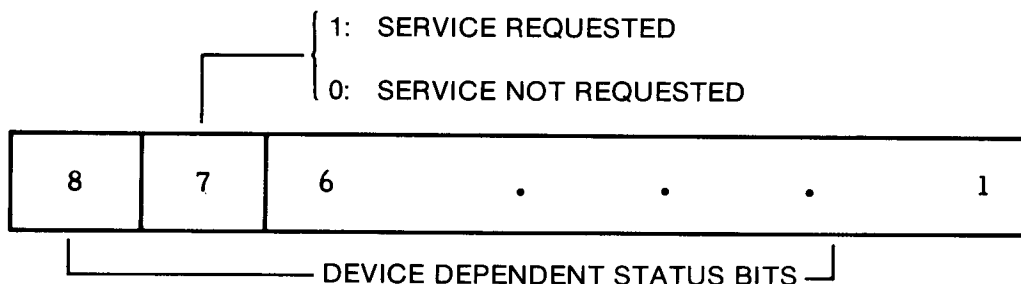
in this case. Other decimal values from 0 to 255 can be returned to the variable A depending on the device being polled and the state of the device.

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



STATUS BYTE:



TYPICAL HP USE:

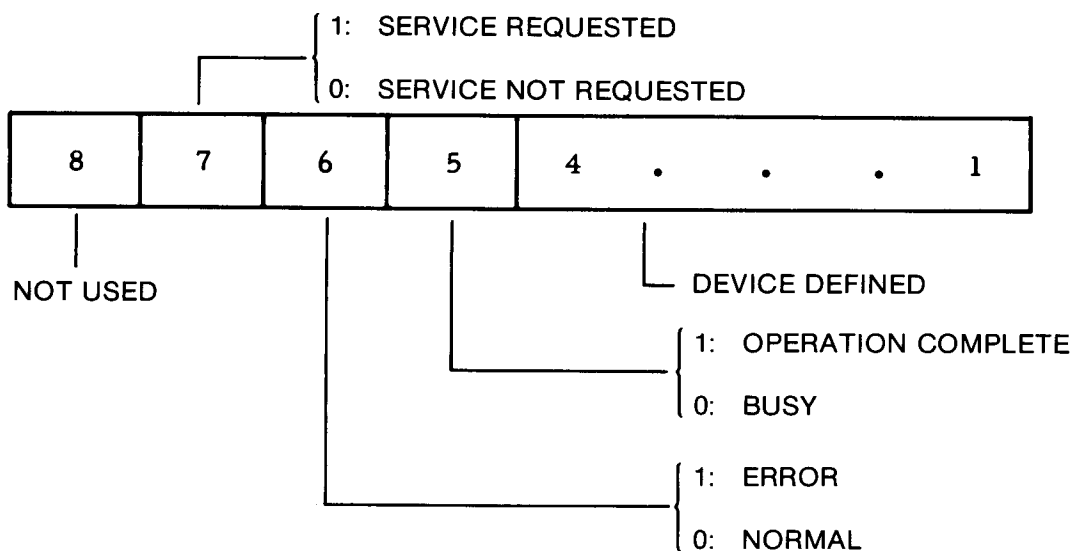


Figure 5-1. Device Poll Status Byte

oo

Polling Strategies

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 AC1]=RBYTE13
280 CMD "Y"
290 AC2]=RBYTE13
300 CMD "?U"
310 OUTPUT (13,240)256,25,512;
320 IF AC1]#64 THEN 340
330 GOSUB 530
340 IF AC2]#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

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.

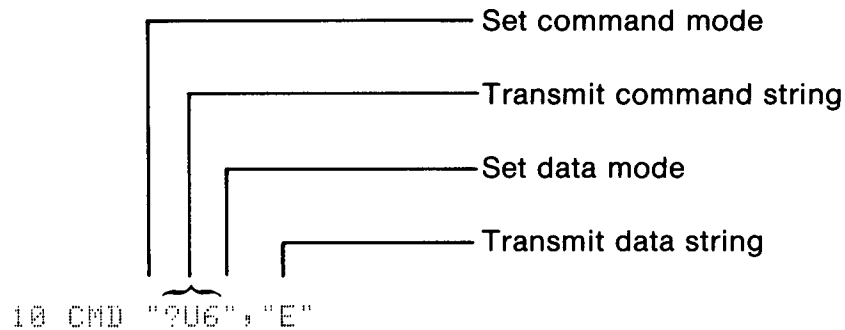
[illegible]

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.

The typical form for the BUS COMMAND statement is

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.

A-1



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.

[illegible]

The Data Mode

To return the bus to the data mode execute the statement

40 OUTPUT (13,20)512;

Code to set data mode

Format statement number

or any CMD statement, such as

10 CAD " "

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

[illegible]

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

Appendix B, ASCII CHARACTER CODES □□□□□□□□□□□□□□□□□□□□□□

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 "2U" 20 FORMAT B 30 OUTPUT (13,20)768;
	Set Local	Returns all devices to local control.	100 CMD "2U" 110 FORMAT B 120 OUTPUT (13,110)1024;
PROGRAMMING INSTRUMENT	Using literals as program codes	Calculator sends program codes to control listening device.	10 CMD "2U <Listen Address>"; " <Program Code String> " 10 CMD "2U\$"; "L1000.0="
	Using variable values as program codes	Calculator transmits program code string containing variable values to control listening device.	10 CMD "2U <Listen Address> " 20 FORMAT <format list> 30 OUTPUT (13,20) <variable list> 10 CMD "2U\$" 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 "2V5"
	Calculator-to-Instrument	The calculator sends data to other devices on the bus.	20 FORMAT 4X:E12.0 30 ENTER (13,20)A 10 CMD "2U2*"
	Instrument-to-Instrument	The calculator commands one device to send data to other devices.	20 FORMAT <format list> 30 OUTPUT (13,20) <variable list> 10 CMD "? <Talk Address> <Listen Addresses> " 20 FORMAT F7.2 30 OUTPUT (13,20)A 10 CMD "2V1"
	Interface Clear	Halts all I/O operations on the bus; unaddresses talker and listeners.	Press STOP key.

BUS COMMANDS

	COMMAND	DECIMAL CODE	DESCRIPTION	GENERAL FORM AND EXAMPLES
UNADDRESS COMMANDS	UNL UNLISTEN	63	Clears bus of all listeners.	10 CMD "2"
	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 "2U" 20 FORMAT 3B 30 OUTPUT (13,20)256,C,512; command code _____ Local Lockout _____
	DCL DEVICE CLEAR	20	Returns all responding devices to predetermined states.	10 CMD "2U" 20 FORMAT 3B 30 OUTPUT (13,20)256,8,512;
	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 "2U <Listen Addresses> " 20 FORMAT 3B 30 OUTPUT (13,20)256,C,512; command code _____
	SDC SELECTED DEVICE CLEAR	4	Returns to predetermined states all responding devices that are addressed to listen.	10 CMD "2U89" 20 FORMAT 3B 30 OUTPUT (13,20)256,8,512;
	GET GROUP EXECUTE TRIGGER	8	Initiates a preprogrammed action by all responding devices that are addressed to listen.	Group Execute Trigger _____

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 "2U" 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 AC11=RBYTE13
	Ending a Poll	Serial Poll Disable terminates the serial poll mode.	60 CMD "2U" 70 FORMAT 3B 80 OUTPUT (13,20)256,25,512;

hewlett·packard **i**nterface **bus**
users guide

9830A

Scan Copyright ©
The Museum of HP Calculators
www.hpmuseum.org

Original content used with permission.

Thank you for supporting the Museum of HP
Calculators by purchasing this Scan!

Please to not make copies of this scan or
make it available on file sharing services.