Errata

Title & Document Type: 3852A Command Reference Manual Volume 1

Manual Part Number: 03852-90016V1

Revision Date: December 1987

HP References in this Manual

This manual may contain references to HP or Hewlett-Packard. Please note that Hewlett-Packard's former test and measurement, semiconductor products and chemical analysis businesses are now part of Agilent Technologies. We have made no changes to this manual copy. The HP XXXX referred to in this document is now the Agilent XXXX. For example, model number HP8648A is now model number Agilent 8648A.

About this Manual

We've added this manual to the Agilent website in an effort to help you support your product. This manual provides the best information we could find. It may be incomplete or contain dated information, and the scan quality may not be ideal. If we find a better copy in the future, we will add it to the Agilent website.

Support for Your Product

Agilent no longer sells or supports this product. You will find any other available product information on the Agilent Test & Measurement website:

www.tm.agilent.com

Search for the model number of this product, and the resulting product page will guide you to any available information. Our service centers may be able to perform calibration if no repair parts are needed, but no other support from Agilent is available.

Agilent Technologies

HP 3852A Data Acquisition/Control Unit

Command Reference Manual
Volume 1

HEWLETT
PACKARD

Copyright © Hewlett-Packard Company, 1986, 1987

Manual Part Number: 03852-90016
Microfiche Part Number: 03852-99016

Printed: DECEMBER 1987 Edition 2
Printed in U.S.A. E1287

Printing History

The Printing History shown below lists the printing dates of all Editions and Updates created for this manual. The Edition number changes as the manual undergoes subsequent revisions. Editions are numbered sequentially starting with Edition 1.

Updates, which are issued between Editions, contain individual replacement pages which the customer uses to update the current Edition of the manual. Updates are numbered sequentially starting with Update 1. When a new Edition is created, all Updates associated with the previous Edition are merged into the manual. Each new Edition or Update also includes a revised copy of this printing history page.

Many product updates and revisions do not require manual changes and, conversely, manual corrections may be done without accompanying product changes. Therefore, do not expect a one-to-one correspondence between product updates and manual updates.

Edition 1 (Part Number 03852-90008) NOVEMBER 1986
Update 1 (Part Number 03852-90090) JULY 1987
Edition 2 (Part Number 03852-90016) DECEMBER 1987

RESTRICTED RIGHTS LEGEND

Use, duplication, or disclosure by the Government is subject to restrictions as set forth in subdivision (b)(3)(ii) of the Rights in Technical Data and Computer Software clause at 52.227-7013.

Hewlett-Packard Company
3000 Hanover Street, Palo Alto, California 94304

HEWLETT PACKARD

Herstellerbescheinigung

Hiermit wird bescheinigt, daß das Gerät/System HP 3852A in Übereinstimmung mit den Bestimmungen von Postverfügung 1046/84 funkentstört ist.

Der Deutschen Bundespost wurde das Inverkehrbringen dieses Gerätes/Systems angezeigt und die Berechtigung zur Überprüfung der Serie auf Einhaltung der Bestimmungen eingeräumt.

Zusatzinformation fur Meß- und Testgeräte

Werden Meß- und Testgeräte mit ungeschirmten Kabeln und/oder in offenen Meßaufbauten verwendet, so ist vom Betreiber sicherzustellen, daß die Funk-Entstörbestimmungen unter Betriebsbedingungen an seiner Grundstücksgrenze eingehalten werden.

Manufacturer's declaration

This is to certify that the equipment HP 3852A is in accordance with the Radio Interference Requirements of Directive FTZ 1046/84. The German Bundespost was notified that this equipment was put into circulation, the right to check the series for compliance with the requirements was granted.

Additional Information for Test- and Measurement Equipment

If Test- and Measurement Equipment is operated with unscreened cables and/or used for measurements on open set-ups, the user has to assure that under operating conditions the Radio Interference Limits are still met at the border of his premises.

NOTICE

The information contained in this document is subject to change without notice.

HEWLETT-PACKARD MAKES NO WARRANTY OF ANY KIND WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Hewlett-Packard shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance or use of this material.

Hewlett-Packard assumes no responsibility for the use or reliability of its software on equipment that is not furnished by Hewlett-Packard.

This document contains proprietary information which is protected by copyright. All rights are reserved. No part of this document may be photocopied, reproduced or translated to another language without the prior written consent of Hewlett-Packard Company.

Copyright © 1985 by HEWLETT-PACKARD COMPANY

HEWLETT PACKARD

CERTIFICATION

Hewlett-Packard Company certifies that this product met its published specifications at the time of shipment from the factory. Hewlett-Packard further certifies that its calibration measurements are traceable to the United States National Bureau of Standards, to the extent allowed by the Bureau's calibration facility, and to the calibration facilities of other International Standards Organization members.

WARRANTY

This Hewlett-Packard instrument product is warranted against defects in materials and workmanship for a period of one year from date of shipment. During the warranty period, Hewlett-Packard Company will, at its option, either repair or replace products which prove to be defective.

For warranty service or repair, this product must be returned to a service facility designated by -hp-. Buyer shall prepay shipping charges to -hp- and -hp- shall pay shipping charges to return the product to Buyer. However, Buyer shall pay all shipping charges, duties, and taxes for products returned to -hp- from another country.

Duration and conditions of warranty for this instrument may be superceded when the instrument is integrated into (becomes a part of) other -hp- instrument products.

Hewlett-Packard warrants that its software and firmware designated by -hp- for use with an instrument will execute its programming instructions when properly installed on that instrument. Hewlett-Packard does not warrant that the operation of the instrument, or software, or firmware will be uninterrupted or error free.

LIMITATION OF WARRANTY

The foregoing warranty shall not apply to defects resulting from improper or inadequate maintenance by Buyer, Buyer-supplied software or interfacing, unauthorized modification or misuse, operation outside of the environmental specifications for the product, or improper site preparation or maintenance.

NO OTHER WARRANTY IS EXPRESSED OR IMPLIED. HEWLETT-PACKARD SPECIFICALLY DISCLAIMS THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

EXCLUSIVE REMEDIES

THE REMEDIES PROVIDED HEREIN ARE BUYER'S SOLE AND EXCLUSIVE REMEDIES. HEWLETT-PACKARD SHALL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, WHETHER BASED ON CONTRACT, TORT, OR ANY OTHER LEGAL THEORY.

ASSISTANCE

Product maintenance agreements and other customer assistance agreements are available for Hewlett-Packard products.

For any assistance, contact your nearest Hewlett-Packard Sales and Service Office. Addresses are provided at the back of this manual.

F

HEWLETT PACKARD

SAFETY SUMMARY

The following general safety precautions must be observed during all phases of operation, service, and repair of this instrument. Failure to comply with these precautions or with specific warnings elsewhere in this manual violates safety standards of design, manufacture, and intended use of the instrument. Hewlett-Packard Company assumes no liability for the customer's failure to comply with these requirements.

GROUND THE INSTRUMENT

To minimize shock hazard, the instrument chassis and cabinet must be connected to an electrical ground.

DO NOT OPERATE IN AN EXPLOSIVE ATMOSPHERE

Do not operate the instrument in the presence of flammable gases or fumes. Operation of any electrical instrument in such an environment constitutes a definite safety hazard.

KEEP AWAY FROM LIVE CIRCUITS

Operating personnel must not remove instrument covers. Component replacement and internal adjustments must be made by qualified maintenance personnel. Under certain conditions, dangerous voltages may exist even with the instrument switched off. To avoid injuries, always disconnect input voltages and discharge circuits before touching them.

DO NOT SERVICE OR ADJUST ALONE

Do not attempt internal service or adjustment unless another person, capable of rendering first aid and resuscitation, is present.

DO NOT SUBSTITUTE PARTS OR MODIFY INSTRUMENT

Because of the danger of introducing additional hazards, do not install substitute parts or perform any unauthorized modification to the instrument. Return the instrument to a Hewlett-Packard Sales and Service Office for service and repair to ensure that safety features are maintained.

DO NOT OPERATE A DAMAGED INSTRUMENT

Whenever it is possible that the safety protection features built into this instrument have been impaired, either through physical damage, excessive moisture, or any other reason, REMOVE POWER and do not use the instrument until safe operation can be verified by service-trained personnel. If necessary, return the instrument to a Hewlett-Packard Sales and Service Office for service and repair to ensure that safety features are maintained.

C

HEWLETT PACKARD

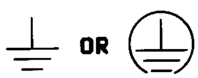
Operating and Safety Symbols

Symbols Used On Products And In Manuals

~ LINE — AC line voltage input receptacle.

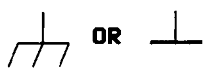
Instruction manual symbol affixed to product. Warns and cautions the user to refer to respective instruction manual procedures to avoid personal injury or possible damage to the product.

Indicates dangerous voltage – terminals connected to interior voltage exceeding 1000 volts.



Protective conductor terminal. Indicates the field wiring terminal that must be connected to earth ground before operating equipment – protects against electrical shock in case of fault.

Clean ground (low-noise). Indicates terminal that must be connected to earth ground before operating equipment – for single common connections and protection against electrical shock in case of fault.



Frame or chassis ground. Indicates equipment chassis ground terminal – normally connects to equipment frame and all metal parts.

Affixed to product containing static sensitive devices – use anti-static handling procedures to prevent electrostatic discharge damage to components.

NOTE

NOTE
Calls attention to a procedure, practice, or condition that requires special attention by the reader.

CAUTION

CAUTION
Calls attention to a procedure, practice, or condition that could possibly cause damage to equipment or permanent loss of data.

WARNING

WARNING
Calls attention to a procedure, practice, or condition that could possibly cause bodily injury or death.

WARNING, CAUTION, and NOTE Symbols

Some labels on the HP 3852A, HP 3853A, and plug-in accessories include an international warning symbol (triangle with subscripted number) which refers the reader to the manuals for further information. This table shows the warning symbols used for the HP 3852A/3853A and plug-in accessories. Refer to the manual set for specific information on WARNINGS, CAUTIONS, or NOTES referenced with a warning symbol.

HP 3852A WARNING, CAUTION, and NOTE Symbols

Symbol	Meaning	Location
⚠ 1	Shock hazard originating outside the instrument (field wiring)	. Analog Extender Connector on Power Supply Modules . Terminal modules on plug-in accessories . Component module covers on plug-in accessories
⚠ 2	Treat all channels as "one circuit" for safety purposes.	. Inside terminal modules on plug-in accessories . Metal cover on component modules of plug-in accessories
⚠ 3	Maximum number of certain plug-in accessories to be installed into an HP 3852A or HP 3853A.	. HP 44701A, HP 44702A/B, HP 44727A/B/C plug-in accessories
⚠ 4	If High-Speed FET multiplexers are used with the HP 44702A/B, ribbon cable may be connected.	. HP 44711A, 44712A, 44713A (referenced on HP 44702A and HP 44702B)
⚠ 5	The instrument should not be operated at a line frequency of 440Hz with a line voltage of 200 V or greater as the AC leakage current may exceed 3.5mA	. HP 3852A, HP 3853A Power Supply Modules

Table of Contents

Chapter 1 - USING THIS MANUAL

Chapter 2 - COMMANDS

Chapter 3 - COMMAND SUMMARY

Chapter 4 - GLOSSARY

Chapter 5 - USEFUL TABLES

Chapter 6 - ERROR MESSAGES

Chapter 1
Using This Manual

CONTENTS

Chapter 1

Using This Manual

Introduction

The HP 3852A Command Reference Manual contains an alphabetical description of the HP 3852A Data Acquisition and Control Unit command set. It includes commands for the HP 3852A mainframe and for HP 447XXX plug-in accessory commands.

Manual Contents

This manual has six chapters. An overview of Chapters 2 through 5 follows.

Chapter 2 - Commands

This chapter contains an alphabetical listing of commands for the HP 3852A mainframe and HP 447XXX plug-in accessories. Each command entry identifies the the applicable accessories (or mainframe) and includes a description of the command, prerequisites, syntax, parameters, remarks, and examples.

Chapter 3 - Command Summary

This chapter lists commands by functional group and indicates whether or not the command is allowed in a subroutine, whether it can be executed from the front panel while the mainframe is in remote, and if executing the command aborts the current or ongoing measurement requiring that it be retriggered. Functional groupings are organized according to mainframe commands, plug-in accessory commands, and math operations.

Chapter 4 - Glossary

The glossary describes addressing conventions, data destinations, and data formats for the HP 3852A. The glossary also contains detailed descriptions of parameters frequently used in the command set.

Chapter 5 - Useful Tables

This chapter includes tables of HP 3852A addressing conventions, mainframe and plug-in accessory power-on states, HP 3852A data formats, and conversion routines for various packed data formats.

Chapter 6 - Error Messages

This chapter lists the error messages by error number for the HP 3852A. It also includes a description of each error and possible causes for the error.

Chapter 2
Commands

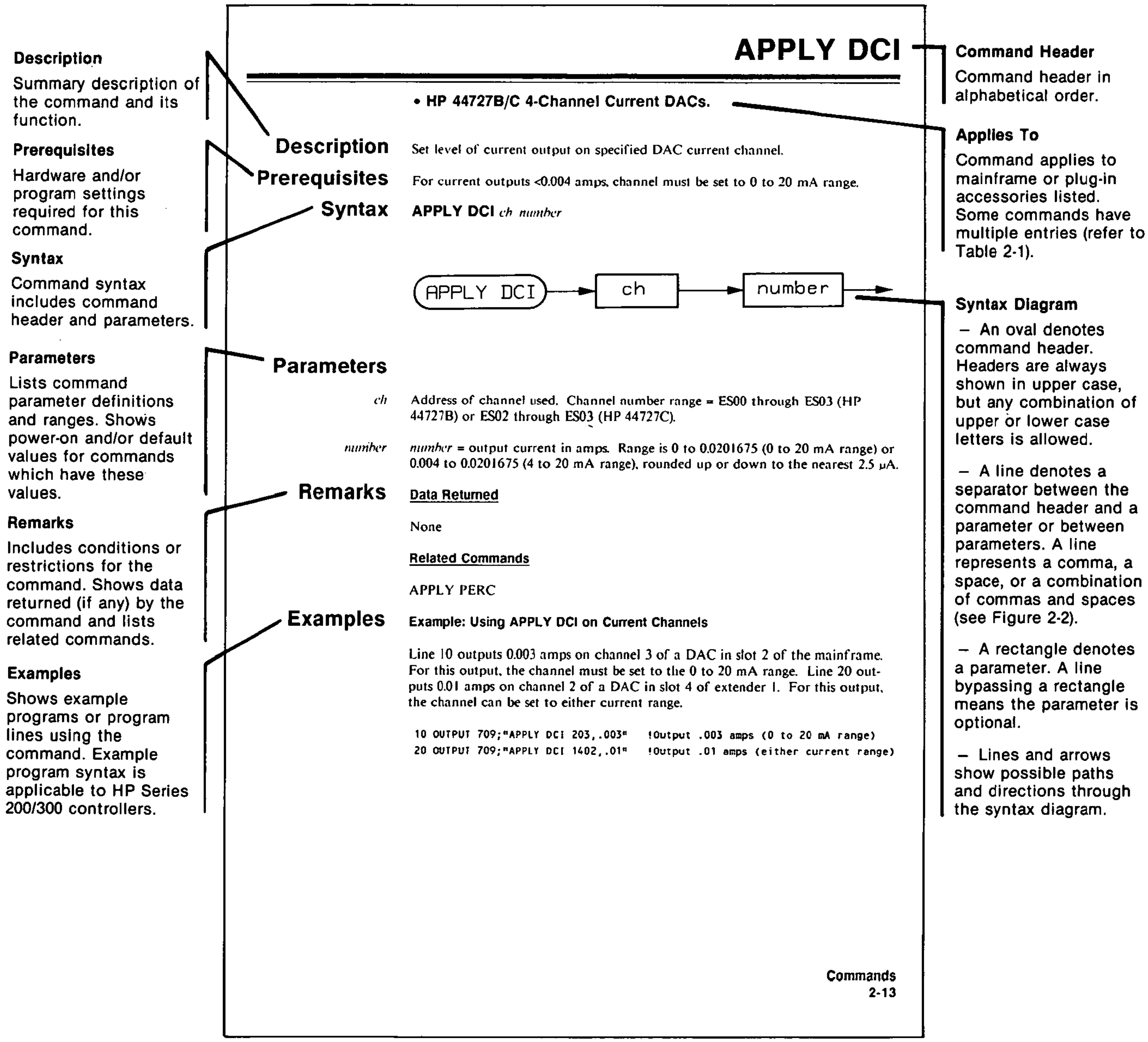
CONTENTS

Chapter 2 Commands

Commands

Reading a Command Reference Page

This chapter consists of an alphabetical listing of commands for an HP 3852A Data Acquisition/Control Unit and the HP 447XXX plug-in accessories. Each command is listed on a command reference page. Figure 2-1 shows a typical command reference page.



APPLY DCI

• HP 44727B/C 4-Channel Current DACs.

Description — Set level of current output on specified DAC current channel.

Prerequisites — For current outputs <0.004 amps, channel must be set to 0 to 20 mA range.

Syntax — APPLY DCI *ch number*

Parameters

ch — Address of channel used. Channel number range = ES00 through ES03 (HP 44727B) or ES02 through ES03 (HP 44727C).

number — *number* = output current in amps. Range is 0 to 0.0201675 (0 to 20 mA range) or 0.004 to 0.0201675 (4 to 20 mA range), rounded up or down to the nearest 2.5 μA.

Remarks

Data Returned

None

Related Commands

APPLY PERC

Examples

Example: Using APPLY DCI on Current Channels

Line 10 outputs 0.003 amps on channel 3 of a DAC in slot 2 of the mainframe. For this output, the channel must be set to the 0 to 20 mA range. Line 20 outputs 0.01 amps on channel 2 of a DAC in slot 4 of extender 1. For this output, the channel can be set to either current range.

```
10 OUTPUT 709;"APPLY DCI 203,.003"   !Output .003 amps (0 to 20 mA range)
20 OUTPUT 709;"APPLY DCI 1402,.01"   !Output .01 amps (either current range)
```

Commands
2-13

Figure 2-1. Typical Command Reference Page

Each command reference page consists of a command header entry, a description of the command, a summary of prerequisites for the command, a syntax ("railroad") diagram for the command, command parameters, remarks and program examples. If a command is continued on following page(s), "command header (cont'd)" is shown on the top of the following page(s).

All command reference pages have the same format. However, several commands have more than one entry. For example, the CHREAD command has four entries: one for the HP 44701A/HP 44702A/B; one for the HP 44715A; one for the HP 44721A/HP 44722A; and one for the HP 44723A.

Therefore, when referring to a command reference page, check the accessories listing shown at the top of the page to ensure that the command applies to the accessory you are programming.

Reading a Syntax Diagram

As shown in Figure 2-1, each command has an associated syntax ("railroad") diagram. Figure 2-2 is the syntax diagram representation for HP 3852A commands.

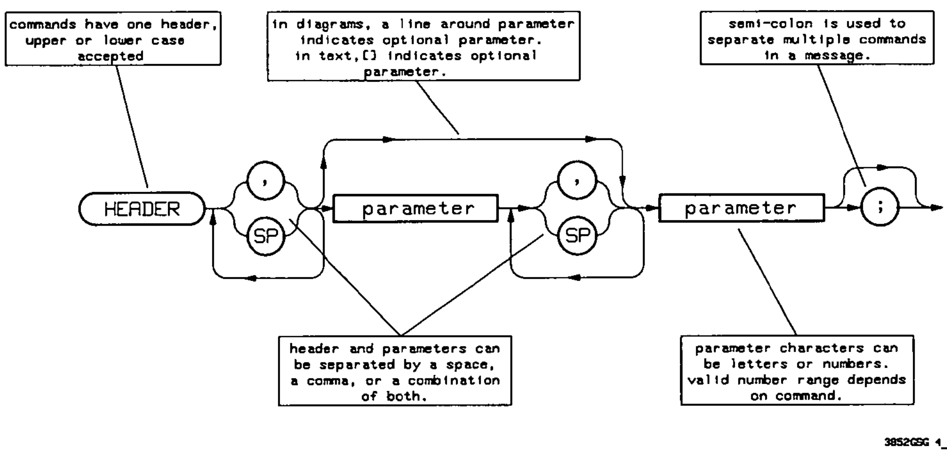


Figure 2-2. HP 3852A Syntax Diagram Format

Command Header

HP 3852A commands consist of a command header followed (usually) by parameters which can be required or optional. The command format has the form HEADER parameter [parameter], where brackets indicate an optional parameter.

Headers can be entered in upper or lower case. However, in this manual, headers appear in upper case and usually in bold (e.g., PACER where PACER is a command header).

Punctuation

You can use a comma, a space, or any combination of commas and spaces between the command header and a parameter or between any two parameters. More than one command can be sent in a single OUTPUT message if commands are separated by a semi-colon (;).

Although any combination of commas and/or spaces can be used between the header and parameters or between parameters, the diagrams in this manual use a line between the command header and a parameter and between parameters.

Command Parameters

Parameters can be entered in upper or lower case, but appear in this manual in lower case italics (e.g., *period* [*count*] where *period* is a required parameter and *count* is an optional parameter). Also, in this manual, optional parameters are enclosed by brackets, but the brackets are not part of the command syntax.

Required parameters must be specified each time the command is executed. Optional parameters can be omitted when executing the command. If an optional parameter is not specified, a default value for the parameter is used if one exists.

Numeric Expressions

Numeric expressions can be in integer, floating point, or exponential format. Numbers in floating point format are rounded to the nearest integer if the command requires an integer. A numeric expression can be a combination of math functions (+, -, *, /, ^, PI, ABS, EXP, FRACT, INT, LGT, LOG, SGN, or SQR), trigonometric operations (ATN, COS, or SIN), or binary functions (BINAND, BINCMP, BINEOR, BINIOR, or BIT).

Any parameter for which a number(s) is specified (such as channel address, for example) can be represented by a parenthesized numeric expression or by a variable or array. For example, SQR(4) can be used instead of the number 2 as a command parameter value.

Example: SCAN Command Syntax Diagram

For example (see Figure 2-3), in the command SCAN [*backplane_bus*] *ch_list* [NSCAN *number*], SCAN is the command header, [*backplane_bus*] and [NSCAN *number*] are optional parameters (indicated in the diagram by a line around the parameters), and *ch_list* is a required parameter.

For this command, you can use any combination of commas and/or spaces between the command header (SCAN) and the [*backplane_bus*] parameter, between SCAN and the *ch_list* parameter, between the [*backplane_bus*], *ch_list*, and [NSCAN *number*] parameters and so on.

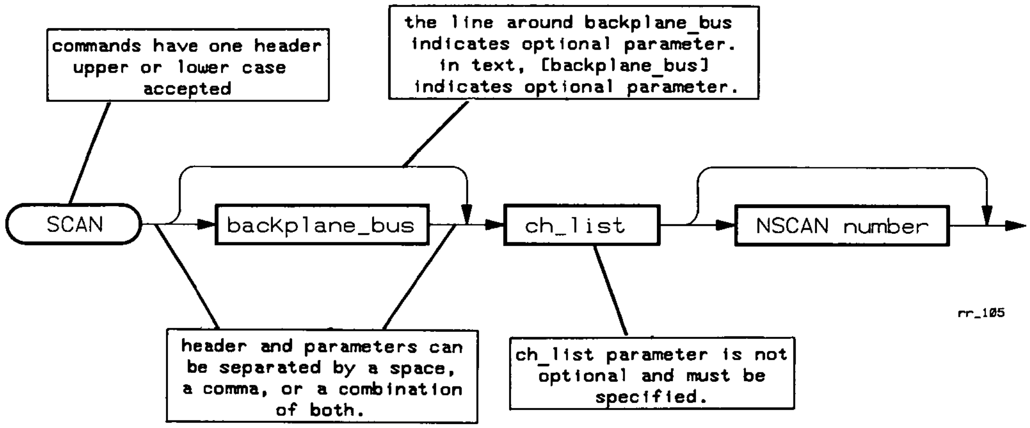


Figure 2-3. Example: SCAN Command Syntax Diagram

• **Mainframe**

Description

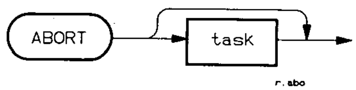
Aborts the currently executing subroutine called/activated by the specified task. Queued subroutines are not aborted.

Prerequisites

The ABORT command is used when the HP 3852A is in the multitasking mode. Multitasking is only available with mainframe firmware revision 3.0 or greater.

Syntax

ABORT [*task*]



Parameters

task Task from which the subroutine to be aborted was called. If a task is not specified, the subroutine containing the ABORT command is aborted.

TASK	DESCRIPTION
HPIB	Currently executing subroutine called from the HP-IB task is aborted.
KYBD	Currently executing subroutine called from the front panel task is aborted.
INTR	Currently executing subroutine called in response to an interrupt is aborted.
run task number	Number of the active/scheduled run task subroutine that is aborted.

Remarks

Active, Scheduled, and Queued Run Tasks

A subroutine directed to a run task in a multitasking system is either active, scheduled, or queued. For example, if subroutines A, B, C, and D are directed to run tasks 1, 2, and 3 in the following manner:

```
100   OUTPUT 709;"RUN 1 A"
110   OUTPUT 709;"RUN 1 B"
120   OUTPUT 709;"RUN 2 C"
130   OUTPUT 709;"RUN 3 D"
```

subroutines A, C, and D will begin to time-slice. While subroutine A is executing, it is termed "active", while subroutines C and D are "scheduled". When the system swaps to subroutine C, C becomes active while A and D are scheduled, and so on. Subroutine B is a "queued" subroutine behind subroutine A. Subroutine B will remain in the queue until A completes or is aborted. At that time, B will be swapped in and time-slice with C and D provided they have not finished.

Suspended Subroutines

Any subroutine whose execution has been suspended by the SUSPEND, WAITFOR SIGNAL, or PAUSE command can also be aborted.

Aborted Subroutines are not Deleted

When a subroutine has been aborted, the subroutine is still present in mainframe memory. Thus, if desired, the subroutine can be called or activated again.

Data Returned

None

Related Commands

ENABLE MULTI, RUN

Examples

Example: Aborting a Run Task Subroutine

This example shows how the ABORT command is used to abort a run task subroutine.

In the program, two subroutines are downloaded and directed to run tasks. Subroutine A (run task 1) counts from 0 to 199 and subroutine B (run task 2) repeatedly displays A B C. To demonstrate the use of the ABORT command, an IF ... END IF loop (lines 180-200) is inserted into subroutine A to monitor the count. When a count of 100 is reached, the ABORT command (line 190) is executed and the subroutine aborts itself. At that point, subroutine B runs continuously until it finishes.

```
10    !Set a 65 ms time-slice and set the number of run tasks and the
20    !queue size the system will allow. Place the HP 3852A in the
30    !multitasking mode. The reset caused by ENABLE MULTI will load
40    !the TSLICE and NTASKS parameters into the operating system.
50    !
60    OUTPUT 709;"TSLICE .065"
70    OUTPUT 709;"NTASKS 2,2"
80    OUTPUT 709;"ENABLE MULTI"
90    !
100   !Write and download the subroutines to be directed to run tasks.
110   !Subroutine A when active, begins counting from 0 to 199. When a
120   !count of 100 is reached, the subroutine aborts.
130   !
140   OUTPUT 709;"SUB A"
150   OUTPUT 709;"  INTEGER I"
160   OUTPUT 709;"  FOR I=0 TO 199"
170   OUTPUT 709;"    DISP I"
180   OUTPUT 709;"    IF I > 100 THEN"
190   OUTPUT 709;"      ABORT"
200   OUTPUT 709;"    END IF"
```

```
210   OUTPUT 709;"    WAIT .1"
220   OUTPUT 709;"  NEXT I"
230   OUTPUT 709;"SUBEND"
240   !
250   !Download subroutine B. When active, subroutine B repeatedly
260   !displays A B C. This run task will time-slice with subroutine A
270   !until subroutine A is aborted. At that point, subroutine B will
280   !execute by itself until it is finished.
290   !
300   OUTPUT 709;"SUB B"
310   OUTPUT 709;"  INTEGER K"
320   OUTPUT 709;"  FOR K=0 TO 49"
330   OUTPUT 709;"    DISP 'A'"
340   OUTPUT 709;"    WAIT .1"
350   OUTPUT 709;"    DISP 'B'"
360   OUTPUT 709;"    WAIT .1"
370   OUTPUT 709;"    DISP 'C'"
380   OUTPUT 709;"    WAIT .1"
390   OUTPUT 709;"  NEXT K"
400   OUTPUT 709;"SUBEND"
410   !
420   !Direct subroutines to run tasks. Direct subroutine A to
430   !run task 1 and subroutine B to run task 2. Each run task will
440   !execute one time.
450   !
460   OUTPUT 709;"RUN 1 A"
470   OUTPUT 709;"RUN 2 B"
480   END
```

As this program executes, A B C is repeatedly displayed along with an increasing count. When the count reaches 100, only A B C A B C ... is displayed since the subroutine of run task 1 was aborted.

ABS

• Math Function

Description

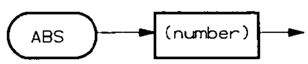
Math function. Returns the absolute value of its argument.

Prerequisites

The ABS function is only available with mainframe firmware revision 3.0 or greater.

Syntax

ABS (*number*)



R.ABS

Parameters

number Number or numeric expression.

Remarks

Data Returned

The data returned is the absolute value of the number or expression. The result will be the same type (REAL or INTEGER) as the argument. Note that ABS of the INTEGER -32768 returns ERROR 42: MATH ERROR - INTEGER OVERFLOW.

Related Commands

None

Examples

Example: Returning the Absolute Value of -32768

Stated in the Remarks section was that the absolute value of INTEGER -32768 returns an error message. The following programs demonstrate how an INTEGER value of -32768 might be obtained and cause an error message and when -32768 would not.

```
10   OUTPUT 709;"INTEGER A"
20   OUTPUT 709;"A=-32768"
30   OUTPUT 709;"VREAD ABS(A)"
40   END
```

```
10   OUTPUT 709;"VREAD ABS (-32767-1)"
20   ENTER 709;A
30   PRINT A
40   END
```

In these programs, the INTEGER -32768 is evaluated by the ABS function and the following error message is returned:

```
ERROR 42: VREAD: MATH ERROR - INTEGER OVERFLOW
```

In the following program, the - (minus sign) is interpreted as an operator rather than part of the number. Thus, no error message is returned.

```
10   OUTPUT 709;"VREAD ABS(-32768)"
20   ENTER 709;A
30   PRINT A
40   END
```

```
32768
```

ADDR

• **Mainframe**

Description

Set HP 3852A HP-IB address. ADDR can be executed from the front panel only.

Prerequisites

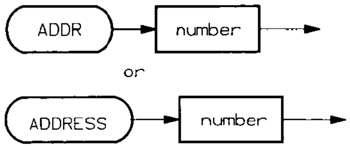
HP 3852A must be in LOCAL (front-panel) mode.

Syntax

ADDR *number*

or

ADDRESS *number*



Parameters

number — *number* from 0 to 30 sets HP 3852A HP-IB address. Factory-set address = 9. Default/power-on address = address previously set.

Remarks

Address Stored in Nonvolatile Memory

Address is stored in nonvolatile memory and does not change when mainframe is cleared or reset, or when power is cycled.

Data Returned

None

Related Commands

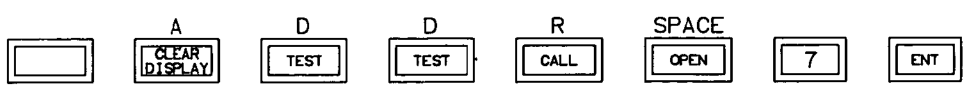
ADDR?

Examples

Example: Setting HP-IB Address

This example shows how to set HP-IB address 7. Note that the address can be changed from the front panel keyboard only.

Press:



- Mainframe

Description

See **ADDR** Command

ADDR?

- Mainframe

Description

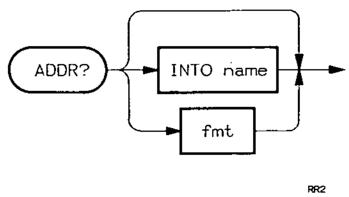
Read HP-IB Address.

Prerequisites

None.

Syntax

ADDR? [INTO *name*] or [*fmt*]



Parameters

INTO *name* — See Glossary.

fmt — See Glossary. Default format is IASC.

Remarks

Data Returned

HP 3852A HP-IB address.

Related Commands

ADDR

Examples

Example: Read HP-IB Address

To read the current HP-IB address, press:

If, for example, the current address is 9, ADDR? is displayed on the left display window and 9 is displayed on the right display window.

• Mainframe

Description

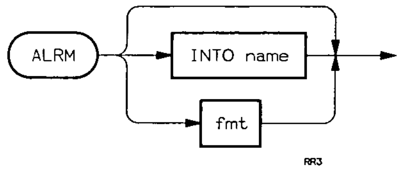
Returns the current setting of the real-time alarm, referenced to the system real-time clock.

Prerequisites

None.

Syntax

ALRM [INTO *name*] or [*fmt*]



Parameters

INTO *name* See Glossary.

fmt See Glossary. Default format is DASC.

Remarks

Time Set When Memory Lost

Alarm time is stored in battery backed-up memory. If memory is lost, alarm time is set to 0.

Data Returned

If the system real-time clock is set to 0 at 2400 hours (midnight) local time, value returned by ALRM is the number of seconds after midnight for the alarm to be initiated (if enabled). The resolution of ALRM is 0.001 second (1 msec).

Related Commands

SET ALRM, IALRM, ENABLE, ALRM, DISABLE ALRM, ON ALRM, CALL, OFF ALRM, RQS, STA?, SRQ

Examples

Example: Set and Read Alarm Setting

This program sets the real-time alarm to 06:56:40 hours and reads the time set. Note that the alarm will not actually be enabled at 06:56:40 hours unless the alarm was previously enabled with an ENABLE ALRM command. A typical return for this program is 06:56:40 on the controller CRT, for controllers with a TIME$ function.

```
10 OUTPUT 709;"RST"                !Reset the HP 3852A
20 OUTPUT 709;"SET ALRM 25000"     !Set alarm for 06:56:40
30 OUTPUT 709;"ALRM"               !Read alarm setting
40 ENTER 709;A                     !Enter alarm setting
50 PRINT TIME$(A)                  !Print real-time for 25000 sec
60 END
```

AND

- **Logical Operator**

Description

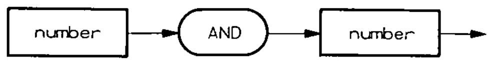
Returns a TRUE (1) or FALSE (0) depending on the logical AND of the numbers.

Prerequisites

The AND statement is only used in an IF...END IF or in a WHILE...END WHILE construct which, in turn, must be included in an HP 3852A subroutine.

Syntax

number **AND** *number*



RR_4

Parameters

number Constant, variable, or numeric expression. Result returned is TRUE (1) or FALSE (0) depending on the AND of the two *numbers* as shown, where A and B are INTEGER or REAL numbers.

A / B	= 0	<> 0
= 0	F	F
<> 0	F	T

Remarks

Rounding Errors with Real Numbers

When using logical operators, Real numbers should not be used due to rounding errors that may occur. For example, when the following subroutine is called, a beep will sound since A is assigned a value other than 0 due to the rounding which occurred.

```
10    OUTPUT 709;"SUB X"
20    OUTPUT 709;"  REAL A,B"
30    OUTPUT 709;"  A=.1+.1+.1-.3"
40    OUTPUT 709;"  B=1"
50    OUTPUT 709;"  IF A AND B THEN
60    OUTPUT 709;"    BEEP"
70    OUTPUT 709;"  END IF"
80    OUTPUT 709;"SUBEND"
```

AND Priority Higher than OR Priority

In firmware revision 3.5 and greater, the AND operator has a higher priority than the OR operator. Thus, an expression such as: A OR B AND C is interpreted as A OR (B AND C). In firmware revisions prior to 3.5, the AND and OR operators have the same priority.

Examples

Example: AND Statement in IF...END IF Loop

The following program shows one use of the AND statement in an IF...END IF loop. In line 40, the value of A AND B is calculated. If the A value <> 0 and the B value is <> 0, "TRUE" is is displayed. If either value = 0, "FALSE" is displayed.

On the first loop, since A = 3 and B = 2, A AND B is TRUE so "TRUE" is displayed. On the second loop, since A = 3 and B = 0, A AND B is FALSE and "FALSE" is displayed. FASTDISP OFF is set so that you can see the two displays.

```
10 OUTPUT 709;"RST"                  !Reset the HP 3852A
20 OUTPUT 709;"FASTDISP OFF"         !Turn fast display off
30 OUTPUT 709;"INTEGER A,B"          !Define variables A and B
40 OUTPUT 709;"SUB DEC"              !Start of subroutine
50 OUTPUT 709;"  IF A AND B THEN"    !Start loop
60 OUTPUT 709;"    DISP 'TRUE'"      !Disp TRUE if A <>0 and B <>0
70 OUTPUT 709;"  ELSE"
80 OUTPUT 709;"    DISP 'FALSE'"     !Disp FALSE if A = 0 or B = 0
90 OUTPUT 709;"  END IF"             !End loop
100 OUTPUT 709;"SUBEND"              !End subroutine
           .
           .
300 OUTPUT 709;"A=3;B=2"             !Define A and B values
310 OUTPUT 709;"CALL DEC"            !Call subroutine
           .
           .
500 OUTPUT 709;"B=0"                 !Redefine B value
510 OUTPUT 709;"CALL DEC"            !Call subroutine again
520 END
```

APPLY DCI

• **HP 44727B/C 4-Channel Current DACs.**

Description

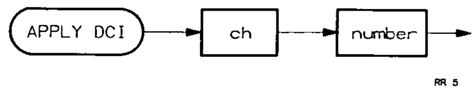
Set level of current output on specified DAC current channel.

Prerequisites

For current outputs <0.004 amps, channel must be set to 0 to 20 mA range.

Syntax

APPLY DCI *ch number*



Parameters

ch — Address of channel used. Channel number range = ES00 through ES03 (HP 44727B) or ES02 through ES03 (HP 44727C).

number — *number* = output current in amps. Range is 0 to 0.0201675 (0 to 20 mA range) or 0.004 to 0.0201675 (4 to 20 mA range), rounded up or down to the nearest 2.5 μA.

Remarks

Data Returned

None

Related Commands

APPLY PERC

Examples

Example: Using APPLY DCI on Current Channels

Line 10 outputs 0.003 amps on channel 3 of a DAC in slot 2 of the mainframe. For this output, the channel must be set to the 0 to 20 mA range. Line 20 outputs 0.01 amps on channel 2 of a DAC in slot 4 of extender 1. For this output, the channel can be set to either current range.

```
10 OUTPUT 709;"APPLY DCI 203,.003"    !Output .003 amps (0 to 20 mA range)
20 OUTPUT 709;"APPLY DCI 1402,.01"    !Output .01 amps (either current range)
```

- **HP 44726A 2-Channel Arbitrary Waveform DAC**

Description

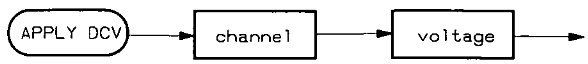
Sets the voltage level on the DAC OUT BNC terminal for the specified channel.

Prerequisites

Requires firmware revision 3.5 or greater and TARM OFF must be set.

Syntax

APPLY DCV *channel voltage*



Parameters

channel Address of the channel on which the voltage is applied. The channel range is ES00 through ES01.

voltage Voltage to be applied from the specified DAC channel. The range for voltage is -10.2396875V to +10.2396875V rounded up or down to the nearest 0.3125 mV.

Remarks

Voltages Are Applied From the DAC OUT BNC Terminals

When APPLY DCV is used, the voltage is output from the channel's DAC OUT BNC terminal. Note that the channels can output voltages simultaneously and the voltages can have different amplitudes.

Data Returned

None

Related Commands

APPLY PERC

APPLY DCV (cont)

Examples

Example: Supplying a DC Voltage with the APPLY DCV Command

The following program segment shows how the DAC accessory can be used to supply a DC voltage to a heater in order to maintain the temperature of a liquid.

```
100  OUTPUT 709;"SUB TEMP"
110  OUTPUT 709;"  REAL T"
120  OUTPUT 709;"  USE 700"
130  OUTPUT 709;"  CONF TEMPT"
140  OUTPUT 709;"  WHILE 1"
150  OUTPUT 709;"    MEAS TEMPT 200 INTO T"
160  OUTPUT 709;"    IF T<100 THEN"
170  OUTPUT 709;"      APPLY DCV 0 10"
180  OUTPUT 709;"    ELSE"
190  OUTPUT 709;"      APPLY DCV 0 0"
200  OUTPUT 709;"    END IF"
210  OUTPUT 709;"  END WHILE"
220  OUTPUT 709;"SUBEND"
230  OUTPUT 709;"RUN 0 TEMP"
   .
   .
   .
```

As this subroutine executes, the voltmeter in slot 7 of the mainframe is configured to make temperature measurements with a T type thermocouple (lines 120 and 130). Measurements are then taken to continuously monitor the temperature of the liquid (lines 140 and 150). When the temperature of the liquid falls below 100° C, channel 0 of the DAC in slot zero applies 10V to the heater (line 170). When the temperature reaches, and remains at 100°, there is no output from the DAC (line 190).

Note that the subroutine is executed as a run task (line 230) which indicates that it is part of a multitasking system.

- **HP 44727A/C 4-Channel Voltage DACs.**

Description

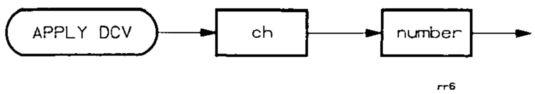
Sets level of output voltage on specified DAC voltage channel.

Prerequisites

For voltage outputs <0V, channel must be set to -10V to +10V range.

Syntax

APPLY DCV *ch number*



Parameters

ch Address of channel. Channel number range = ES00 through ES03 (HP 44727A) or ES00 through ES01 (HP 44747C).

number *number* = output voltage in volts. Range of *number* for -10V to +10V is -10.235 to + 10.235, rounded up or down to the nearest 2.5 mV. Range of *number* for 0V to +10V is 0 to 10.235, rounded up or down to the nearest 2.5 mV.

Remarks

Data Returned

None

Related Commands

APPLY PERC

Examples

Example: Using APPLY DCV on Voltage Channels

Line 10 outputs -5.000 volts on channel 3 of a DAC in slot 2 of the mainframe. For this output, the channel must be set to the -10V to 10V range. Line 20 outputs +4.000 volts on channel 2 of a DAC in slot 4 of extender 1. For this output, the channel can be set to either voltage range.

```
10 OUTPUT 709;"APPLY DCV 203,-5.000"  !Output -5 volts (-10V to 10V range)
20 OUTPUT 709;"APPLY DCV 1402,4.000"  !Output +4 volts (either voltage range)
```

APPLY PERC

• **HP 44726A 2-Channel Arbitrary Waveform DAC**

Description

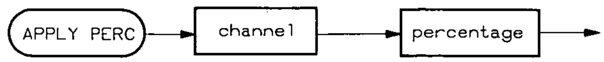
Sets the voltage level on the DAC OUT BNC terminal to a percentage of the DACs maximum range.

Prerequisites

Requires firmware revision 3.5 or greater and TARM OFF must be set.

Syntax

APPLY PERC *channel percentage*



Parameters

channel

Address of the channel on which the voltage is applied. The channel range is ES00 through ES01.

percentage

Percentage of the DACs maximum range that is applied to the specified DAC channel. The following percentages yield the following voltages. Note that the voltage is rounded up or down to the nearest 0.3125 mV.

PERCENTAGE	OUTPUT
0 to +100	0V to +10V
0 to -100	0V to -10V
+100 to +102.35	+10.0V to +10.235V
-100 to -102.35	-10.0V to -10.235V

Remarks

Voltages Are Applied From the DAC OUT BNC Terminals

When APPLY PERC is used, the voltage is output from the channel's DAC OUT BNC terminal. Using the APPLY PERC command twice, the channels can output voltages at the same time and each voltage can be a different percentage of the DACs maximum range.

Data Returned

None

Related Commands

APPLY DCV

Examples

Applying a Percentage of the DACs Maximum Range

In the following program, a subroutine is downloaded which continually monitors the temperature of a room. When the temperature rises above 70° F (21° C), the voltage applied to a fan is increased in order to circulate the air and cool the room. When the room cools, the voltage applied to the fan is decreased in order to maintain the temperature. The APPLY PERC command is used to set the output of the DAC based on the temperature sensed.

```
100  OUTPUT 709;"SUB FAN"
110  OUTPUT 709;"  REAL T"
120  OUTPUT 709;"  USE 700"
130  OUTPUT 709;"  CONF TEMPT"
140  OUTPUT 709;"  WHILE 1"
150  OUTPUT 709;"    MEAS TEMPT 200 INTO T"
160  OUTPUT 709;"    IF T>21 THEN"
170  OUTPUT 709;"      APPLY PERC 0 100"
180  OUTPUT 709;"    ELSE"
190  OUTPUT 709;"      APPLY PERC 0 50"
200  OUTPUT 709;"    END IF"
210  OUTPUT 709;"  END WHILE"
220  OUTPUT 709;"SUBEND"
230  OUTPUT 709;"RUN 1 FAN"
   .
   .
   .
```

As the subroutine executes, the voltmeter in slot 7 is configured to make temperature measurements with a T type thermocouple (lines 120 and 130). Measurements are then taken continuously to monitor the temperature of the room (lines 140 and 150). When the temperature rises above 21° C, 100% (10V) of the DACs maximum range is applied to the fan through channel 0. When the temperature falls below 21°, 50% (5V) of the DACs maximum range is applied to the fan.

Note that the subroutine is executed as a run task (line 230) which indicates that it is part of a multitasking system.

APPLY PERC

• **HP 44727A/B/C 4-Channel Current or Voltage DACs.**

Description

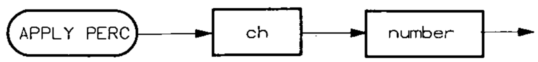
Sets percentage of maximum voltage output from DAC voltage channel or percentage of maximum current output from DAC current channel.

Prerequisites

For voltage outputs <0V, voltage channel must be set to -10V to +10V range. For current outputs <0.004 amps, current channel must be set to 0 to 20 mA range.

Syntax

APPLY PERC *ch number*



Parameters

ch Address of channel. Channel number range = ES00 through ES03.

number Percentage of maximum output for selected voltage or current range, rounded up or down to the nearest 2.5 μA (current channels) or to the nearest 2.5 mV (voltage channels).

Voltage/Current Output vs. Percentage

-10V to 10V	0V to 10V	0 to 20 mA	4 to 20 mA
V = n/10 n = -102.35 to 102.35	V = n/10 n = 0 to 102.35	I = n/5 n = 0 to 100.8375	I = 4 + n/6.25 n = 0 to 101.046875

V = output voltage in Volts
I = output current in mA
n = percentage of maximum output voltage or current

Remarks

Data Returned

None

Related Commands

APPLY DCI, APPLY DCV

Examples

Example: Using APPLY PERC on Voltage and Current Channels

Lines 10 through 40 output a percentage of maximum voltage or current on channel 3 of a DAC in slot 2 of the mainframe. Note that the channel must be set to the appropriate voltage or current range to output the actual voltages or currents shown.

```
10 OUTPUT 709;"APPLY PERC 203,-31.75"   !Output 31.75% max -V (-3.175 volts)
                                        !(-10V to 10V range only)

20 OUTPUT 709;"APPLY PERC 203,50.00"    !Output 50.00% max +V (+5.000 volts)
                                        !(either voltage range)

30 OUTPUT 709;"APPLY PERC 203,25"       !Output 25% max I (0.005 amps)
                                        !(0 to 20 mA range only)

40 OUTPUT 709;"APPLY PERC 203,62.5"     !Output 62.5% max I (0.014 amps)
                                        !(4 to 20 mA range only)
```

APPLY WFV

• HP 44726A 2-Channel Arbitrary Waveform DAC

Description

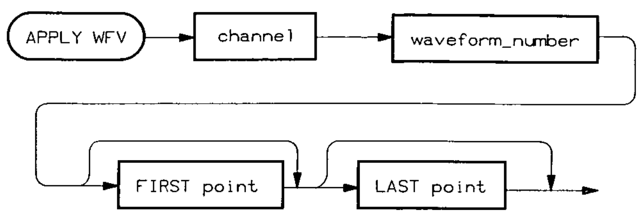
Selects the waveform in channel memory to be output from the DAC.

Prerequisites

Requires firmware revision 3.5 or greater. TARM OFF or TARM AUTO must also be set.

Syntax

APPLY WFV *channel waveform_number* [**FIRST** *point*] [**LAST** *point*]



Parameters

channel — Channel from which the waveform is selected. The range for *channel* is ES00 to ES01.

waveform_number — Number of the waveform which is selected in memory. Waveforms are numbered 0 through 63 per channel.

FIRST *point* — Portion of the waveform selected in memory beginning with the point specified through the LAST point specified, or through the end of the waveform. Note that waveform points are numbered starting with 0.

LAST *point* — Portion of the waveform selected in memory starting with the beginning of the waveform or from the FIRST point specified, through the last point specified.

If neither **FIRST** *point* or **LAST** *point* is specified, the entire waveform is selected.

Remarks

APPLY WFV Does Not Output the Waveform from the DAC

APPLY WFV selects the waveform, it does not output the signal from the DAC. Once a wave is selected, the channel must be armed (TARM) and trigger signals (TRIG) must be received to clock through that portion of memory which provides the data the DAC needs to generate the waveform.

Waveform Frequency

The frequency of waveforms generated by the DAC is dependent upon the time base specified and the number of time base intervals each amplitude point is held. When only a portion of a waveform is selected, the frequency is dependent upon the time base and the number of time base intervals each selected amplitude point is held. The time base used with each waveform is the time base stored by the WFWRITE or WFMOD command, or the time base set by the TBASE command if executed prior to TARM.

Viewing the Waveform

The waveforms generated by the following example programs are easily viewed by connecting the accessory's DAC OUT BNC to an oscilloscope. The plots shown were drawn using the HP 44458A - DACQ/300 software package. Refer to the "Viewing Waveforms" section of the Glossary for an example of how the software is used.

Data Returned

None

Related Commands

WFWRITE ACV, WFWRITE ARB, WFWRITE BIN, WFWRITE RPV, WFWRITE SQV, TARM, TRIG

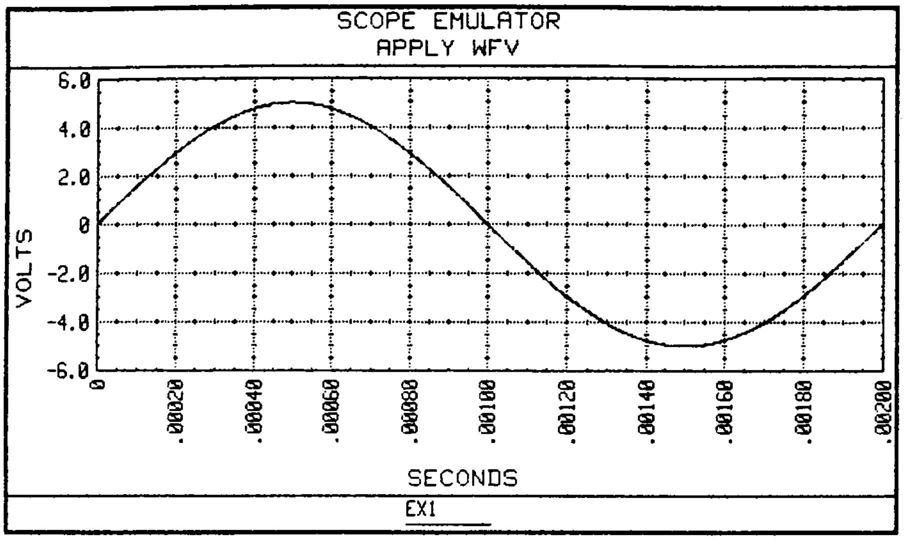
Examples

Example 1: Selecting a Waveform

The following program shows how APPLY WFV is used to select a waveform in memory. It also shows when APPLY WFV is executed relative to when the waveform is defined and the DAC triggered in order for the wave to begin.

```
10    !Reset the mainframe and the HP 44726A. Define and store a
20    !special function sine wave. The sine wave will be a 1000 point
30    !waveform, 10 Vp-p, with a frequency of 500 Hz. The waveform
40    !will be stored on channel 1.
50    !
60    OUTPUT 709;"RST"
70    WAIT 1
80    OUTPUT 709;"WFWRITE ACV 5 10 TBASE (1/(500*1E3)) USE 1"
90    !
100   !Select waveform 5 in channel 1 memory. Arm the channel
110   !so that when triggers are received, the data in that portion of
120   !memory is sent to the DAC which then generates the waveform.
130   !
140   OUTPUT 709;"APPLY WFV 1,5"
150   OUTPUT 709;"TRIG INT USE 1"
160   OUTPUT 709;"TARM AUTO USE 1"
170   END
```

A plot of the waveform is shown on the following page.



Example 2: Selecting a Portion of a Waveform

The following program shows how APPLY WFV is used to select a portion of a waveform in memory.

The waveform defined and stored is the same waveform developed in Example 1. However, instead of selecting the entire waveform, only the first 300 points are selected and output from the DAC.

Note in Example 1 that a 500 Hz waveform was defined:

```
TBASE = 1/(frequency*number of points) = 1/(500*1E3)
```

whereas the frequency of the wave in this example is 150 Hz:

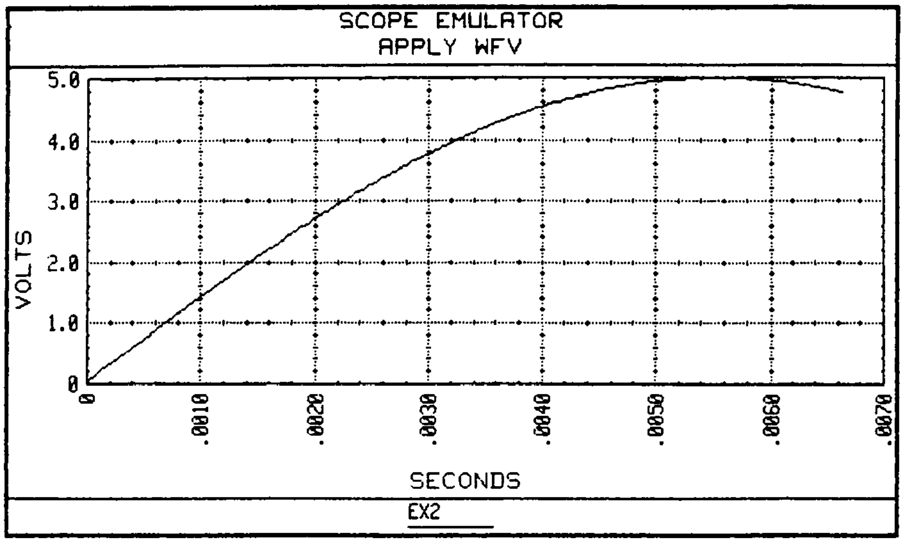
```
frequency = 1/(TBASE*number of points) = 1/(TBASE*300)
```

since the TBASE value used is calculated in line 80 of the following program and 300 points are selected.

```
10    !Reset the mainframe and the HP 44726A. Define and store a
20    !special function sine wave. The sine wave will be a 1000 point
30    !waveform, 10 Vp-p, with a frequency of 500 Hz. The waveform
40    !will be stored on channel 1.
50    !
60    OUTPUT 709;"RST"
70    WAIT 1
80    OUTPUT 709;"WFWRITE ACV 5 10 TBASE (1/(500*1E3)) USE 1"
90    !
100   !Select waveform 5 (points 0 through 299) in channel 1 memory.
110   !Arm the channel so that when triggers are received,
120   !the portion of the wave selected is generated by the DAC.
130   !
140   OUTPUT 709;"APPLY WFV 1,5 LAST 299"
```

```
150   OUTPUT 709;"TRIG INT USE 1"
160   OUTPUT 709;"TARM AUTO USE 1"
170   END
```

A plot of the waveform is shown below:



ARANGE

- **HP 44701A Integrating Voltmeter**

Description

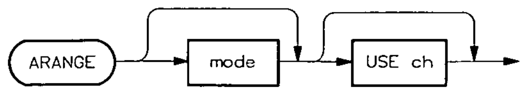
Turns the HP 44701A Integrating Voltmeter autorange mode OFF or ON.

Prerequisites

None

Syntax

ARANGE [*mode*] [USE *ch*]



rr8

Parameters

mode Autorange control mode. With autorange ON, the voltmeter samples the input signal and selects the appropriate range before each measurement. With autorange OFF, the range used for previous measurements is used for subsequent measurements. Power-on and default *mode* = ON.

Mode	Definition
OFF	Autorange disabled
ON	Autorange enabled

USE *ch* Voltmeter slot number. See Glossary.

Remarks

Measurement Speed

With autorange enabled (ARANGE ON), the voltmeter samples the input signal before each measurement, which may affect measurement speed.

Data Returned

None

Related Commands

RANGE, USE

Examples

This program performs DC voltage measurements on a sequence of channels using a fixed voltmeter range (RANGE command). The program then turns autorange on (ARANGE) before measuring a second sequence of channels.

```
10 OUTPUT 709;"RST"              !Reset HP 3852A and HP 44701A
20 OUTPUT 709;"USE 700"          !Use voltmeter in mainframe slot 7
30 OUTPUT 709;"CONF DCV"         !Configure the voltmeter for DC voltage
40 OUTPUT 709;"RANGE 10"         !Voltmeter range (maximum input 10V)
50 OUTPUT 709;"MEAS DCV 0-4"     !Measure DCV on chs 0-4 (slot 0)
60 OUTPUT 709;"ARANGE ON"        !Turn autorange ON for next sequence
70 OUTPUT 709;"MEAS DCV 5-9"     !Measure DCV on chs 5-9 (slot 0)
80 END
```

ARMODE

• **HP 44702A/B High-Speed Voltmeter** (System or Scanner Mode)

Description

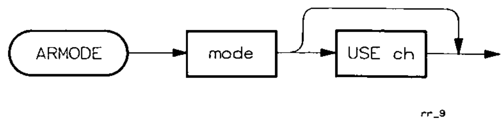
Set autorange mode. Select when autoranging will occur (before or after the Measure Trigger is received).

Prerequisites

When in the Scanner mode, SCTRIG must be set to HOLD.

Syntax

ARMODE *mode* [USE *ch*]



Parameters

mode Autoranging modes. Power-on *mode* = AFTER.

mode	Definition
BEFORE	For ribbon cable scans, autorange when channel advanced but before Measure Trigger is received.
AFTER	Autorange after Measure Trigger received.

USE *ch* Voltmeter slot number. See Glossary.

Remarks

Channel Advance and/or Settling Time

Due to the possibility of long channel advance and settling times, BEFORE should only be used for repeated single channel measurements, or fixed range measurements over the ribbon cable.

Data Returned

None

Related Commands

AZERO, CLWRITE, FUNC, RANGE, USE

Examples

Example: Enabling Autorange Mode

This program sets ARMODE BEFORE then makes 10 DC voltage measurements over the ribbon cable and stores them in mainframe memory.

```
 10 OUTPUT 709;"RST"                          !Reset HP 3852A and HP 44702A/B
 20 OUTPUT 709;"USE 600"                      !Use voltmeter in slot 6
 30 OUTPUT 709;"PACKED PACKSTOR(19)"          !Declare array (10 readings
 40                                           !2 bytes/reading)
 50 OUTPUT 709;"SCANMODE ON"                  !Set scanner mode operation
 60 OUTPUT 709;"CONF DCV"                     !Configure for DC voltage
 70 OUTPUT 709;"RANGE 10"                     !Set range (max. input 10V)
 80 OUTPUT 709;"ARMODE BEFORE"                !Autorange before measure trigger
 90 OUTPUT 709;"MEAS DCV, 500-509, INTO PACKSTOR"
100                                           !Take readings, store in
110                                           !mainframe memory
120 END
```

ASCAN

- **HP 44702A/B High-Speed Voltmeter (Scanner Mode only)**

Description

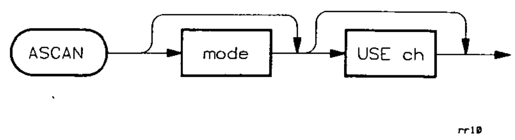
Select autoscan function. ASCAN OFF requires a separate Scan Trigger for each pass through the scan list. ASCAN ON requires only an initial Scan Trigger for the first pass through the scan list. Subsequent passes then start automatically.

Prerequisites

The HP 44702A/B High Speed Voltmeter must be set to Scanner Mode (SCANMODE ON) and SCTRIG must be set to HOLD before setting ASCAN.

Syntax

ASCAN [*mode*] [USE *ch*]



Parameters

mode Autoscan control modes. Power-on *mode* = OFF. Default *mode* = ON.

mode	Definition
OFF	Scan trigger (SCTRIG command) required for each pass through scan list.
ON	Scan trigger (SCTRIG command) required only for first pass through scan list. Subsequent passes start automatically.

USE *ch* Voltmeter slot number. See Glossary.

Remarks

Data Returned

None

Related Commands

SCANMODE, SCTRIG, USE

Examples

Example: Enabling Autoscan Mode

This program makes five scans through a list of 10 channels and stores the 50 readings in mainframe memory. ASCAN ON is set so that only one scan trigger is required (it starts the first scan).

```
 10 OUTPUT 709;"RST"                      !Reset HP 3852A and HP 44702A/B
 20 OUTPUT 709;"USE 600"                  !Use voltmeter in slot 6
 30 OUTPUT 709;"PACKED PRDGS(99)"         !Declare array (50 readings -
 40                                       !2 bytes/reading)
 50 OUTPUT 709;"SCANMODE ON"              !Enable Scanner Mode
 60 OUTPUT 709;"CONF DCV"                 !Configure for DC voltage
 70 OUTPUT 709;"CLWRITE SENSE 500-509"    !Ch list and cable connections
 80 OUTPUT 709;"PRESCAN 5"                !Make 5 scans through ch list
 90 OUTPUT 709;"ASCAN ON"                 !Requires only one scan trigger
100                                       !to make 5 passes through
110                                       !channel list
120 OUTPUT 709;"SCTRIG SGL"               !Start scan with one trigger
130 OUTPUT 709;"XRDGS 600 INTO PRDGS"     !Transfer readings from vm
140                                       !and store in mf memory
150 END
```

ASSIGN

• **HP 44788A HP-IB Controller**

Description

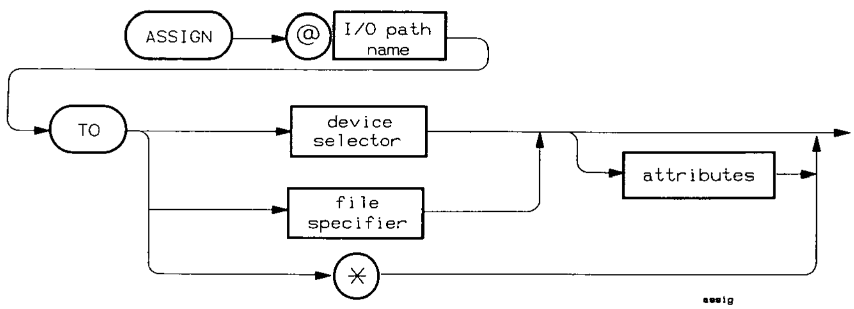
This statement is used to perform one of the following actions: assign an I/O path name and attributes to a device or mass storage file or close an I/O path name. This command cannot be used solely to change attributes.

Prerequisites

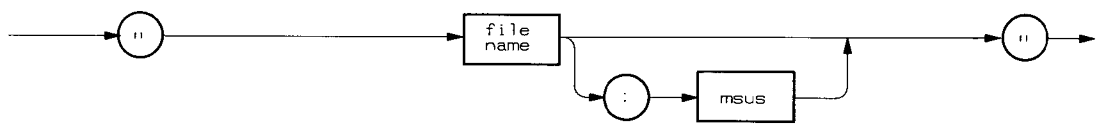
Requires firmware revision 3.5 or greater.

Syntax

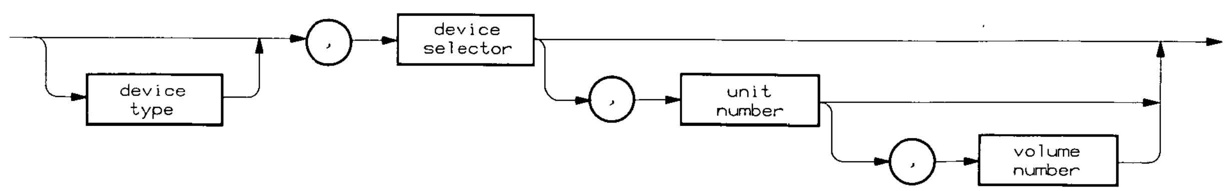
ASSIGN *@I/O path name* **TO** *device selector* or *"file specifier"* or *
[,FORMAT ON or **OFF] [,EOL OFF]**



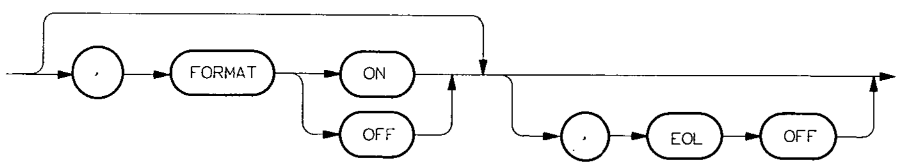
literal form of file specifier



msus:



attributes:



Parameters

@I/O path name The name of the bidirectional path assigned to a device or file.

device selector The HP-IB select code (i.e. ESnn) for the device being assigned or 1 for the HP 3852A display. The local HP-IB address is ESnn, where E = extender # (mainframe = 0), S = slot, and nn = device address.

***** Closes the I/O path when it is assigned to *.

file specifier The name of the file and optional disc address.

FORMAT Allows internal bit representation when OFF or ASCII byte representation when ON.

EOL When OFF, no CR/LF is sent. EOL is not applicable if assigned to the display (device selector = 1).

Remarks

ASSIGN Statement Purposes

The ASSIGN statement has three primary purposes. Its main purpose is to create an I/O path name and assign that name to an I/O resource and attributes that describe the use of that resource. Associated with an I/O path name is a unique data type that uses about 435 bytes of memory for a file and 160 bytes for a device.

Once an I/O path name has been assigned to a resource, OUTPUT and ENTER operations can be directed to that I/O path name. This provides the convenience of re-directing I/O operations in a program by simply changing tha appropriate ASSIGN statement. The resource assigned to the I/O path name may be a device or a mass storage file.

The FORMAT Attributes

Assigning the FORMAT ON attribute to an I/O path name directs the HP 3852A to use its ASCII data representation while sending and receiving data through the I/O path. Assigning the FORMAT OFF attribute to an I/O path name directs the computer to use its internal data representation when using the I/O path.

LIF ASCII format (similar to ASCII representation) is always used with ASCII files; thus, if either FORMAT ON or FORMAT OFF is specified for the I/O path name of an ASCII file, it will be ignored.

If a FORMAT attribute is not explicitly given to an I/O path, a default is assigned. If the resource is an ASCII file, the default is always ASCII format.If the resource is a BDAT file, the default FORMAT is OFF.

Using Devices

I/O path names are assigned to devices by placing the device selector after the keyword TO. For example, `ASSIGN @DISPLAY TO 1` creates the I/O path name "@DISPLAY" and assigns it to the HP 3852A display. The statement `ASSIGN @METERS TO 710` creates the I/O path name "@METERS" and assigns it to a device on HP-IB.

A device can have more than one I/O path name associated with it. Each I/O path name can have different attributes, depending upon how the device is used. The specific I/O path name used for an I/O operation determines which set of attributes is used for that operation.

Using Files

Assigning an I/O path name to a file name associates the I/O path with a file on the mass storage media. The mass storage file must be a data file, either ASCII or BDAT. The file must already exist on the media, as ASSIGN does not do an implied CREATE.

ASCII and BDAT files have a position pointer which is associated with each I/O path name. The position pointer identifies the next byte to be written or read, and the value of the position pointer is updated with each ENTER or OUTPUT that uses that I/O path name. The position pointer is reset to the beginning of the file when the file is opened. A file is opened by any ASSIGN statement that includes the file specifier. Open a file with only one I/O path name at a time.

BDAT files have an additional pointer for end-of-file. The end-of-file pointer is read from the media when the file is opened. The end-of-file pointer is updated on the media after each OUTPUT statement.

Data Returned

None

Related Commands

None

Examples

Example: Typical ASSIGN Statements

The following statements assign I/O paths and device selectors. The first two entries show how to enter the same command from either the HP 3852A front panel (no line number) or the system controller (line 100).

```
ASSIGN @FILE TO "MYFILE:,100,1"                 !HP 3852A entry
100 OUTPUT 709;"ASSIGN @FILE TO 'MYFILE:,100,1'" !System Controller entry

ASSIGN @DEVICE TO 122                           !Opens path to 122
ASSIGN @SOURCE TO 309,FORMAT OFF                !Opens path to 309

ASSIGN @SOURCE TO *                             !Closes @Source path
ASSIGN @DEV TO 401,FORMAT ON,EOL OFF            !EOL must follow FORMAT
```

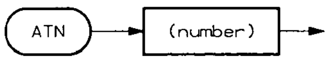
• Trigonometric Operation

Description

Numeric expression evaluated as a command parameter. Returns (in radians) the arctangent of its argument.

Syntax

ATN *(number)*



Parameters

number Number or numeric expression.

Remarks

None

Examples

Example: Using ATN

This program computes the arctangent of a number (.2018) and returns the value (.1991257 radians) to the controller CRT.

```
10 OUTPUT 709;"RST"
20 OUTPUT 709;"VREAD ATN (.2018)"
30 ENTER 709;Res
40 PRINT "Arctangent = ";Res;" radians"
50 END
```

A typical return is:

```
Arctangent = .1991257 radians
```

AUTOST IS

- **HP 44788A HP-IB Controller**

Description

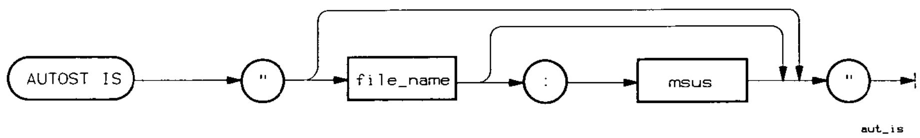
This statement sets up the auto start conditions (file name and device address).

Prerequisites

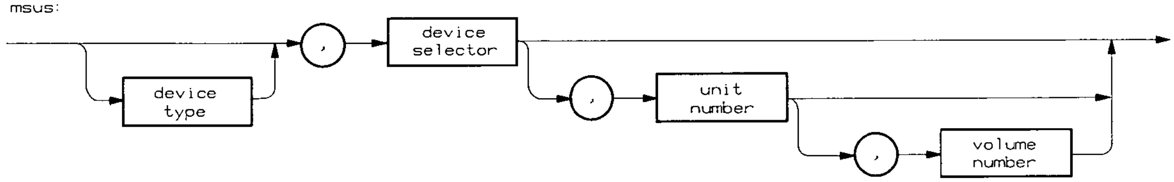
Requires firmware revision 3.5 or greater.

Syntax

AUTOST IS "[*file_name*[:*msus*]]".



msus:



Parameters

file_name	The name of the file containing the commands to be executed.
msus	**Mass storage unit specifier.** ESnn,n,n where ES is extender number (mainframe = 0) and slot number and *nn* is the address set on the drive. *,n,n* identifies the drive and volume number.

Remarks

The AUTOST IS command will accept only an ASCII file as the file type for an auto start file.

If *msus* is not specified in the command, the AUTOST IS command will use the current MSI.

Turning Off the Auto Start Feature

The auto start feature can be turned off by not specifying a file name and msus in the AUTOST IS command i.e. AUTOST IS " ".

Data Returned

None

Related Commands

CREATE ASCII, MSI

Examples

Example: Typical AUTOST Statements

The first statement is entered from a system controller, the second and third are entered from the HP 3852A front panel. The third cancels the auto start feature.

```
100 OUTPUT 709;"AUTOST IS 'AUTOSTART:,200,1'"  !Set file name, device addr
AUTOST IS "MYFILE:,200"                        !Uses the default drive
AUTOST IS " "                                  !Cancels auto start feature
```

- **HP 44701A Integrating Voltmeter**
- **HP 44702A/B High-Speed Voltmeter** (Scanner or System Mode)

Description

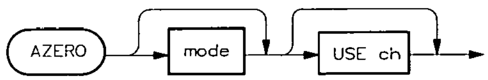
Autozero the voltmeter.

Prerequisites

AZERO ON cannot be used for ACV and autozero has no effect on resistance measurements.

Syntax

AZERO [*mode*] [USE *ch*]



rr_12

Parameters

mode **HP 44701A Integrating Voltmeter**

Autozero control mode. AZERO ON and AZERO OFF are used ONLY for DC voltage measurements with the HP 44701A Integrating Voltmeter. AZERO ON cannot be used for AC voltage measurements. Power-on *mode* = ON. Default *mode* = OFF/ONCE.

mode	Definition
OFF	Autozero once. Zero measurement is updated once, then only after a function, range, or integration time change.
ON	Autozero always. Zero measurement is updated after each measurement.
ONCE	Same as OFF.

HP 44702A/B High-Speed Voltmeter

Autozero control mode. Default *mode* = ONCE.

mode	Definition
ONCE	Autozero once

USE *ch* Voltmeter slot number. See Glossary.

Remarks

Autozero ON (Integrating Voltmeter only)

With AZERO ON, the voltmeter makes an *internal* zero measurement (with the input disabled) following every reading and subtracts the zero measurement from the reading. This more than doubles the time required per reading. When ACV is selected, the autozero mode is always OFF.

With AZERO ON, an NPLC line cycle zero is done. AZERO ON can be faster than AZERO OFF if FUNC/RANGE/NPLC changes are made.

Autozero OFF/ONCE

With AZERO OFF or AZERO ONCE, either voltmeter makes one *internal* zero measurement (upon receipt of the AZERO command) and subtracts it from each subsequent reading. This zero measurement is a 16 line cycle zero for the HP 44701A.

For the HP 44701A Integrating Voltmeter only, a new zero measurement is made whenever the function, range, or integration time is changed. For the HP 44702A/B High-Speed Voltmeter, the voltmeter zeroes on demand only.

Data Returned

None

Related Commands

FUNC, RANGE, USE

Examples

Example: Turning Autozero Off

Assuming an HP 44701A voltmeter is installed in slot 1, this program turns autozero off then measures DC voltage on 10 channels.

```
10  REAL Rdngs(0:9)                  !Declare controller array
20  OUTPUT 709;"RST"                 !Reset HP 3852A
30  OUTPUT 709;"USE 100"             !Use vm in slot 1
40  OUTPUT 709;"CONF DCV"            !Configure for DC voltage
50  OUTPUT 709;"AZERO OFF"           !Turns autozero off
60  OUTPUT 709;"MEAS DCV, 200-209"   !Measure DC voltage on chs 0-9 (slot 2)
70  ENTER 709;Rdngs(*)               !Enter readings into A
80  PRINT USING "K,/";Rdngs(*)       !Print readings
90  END
```

- **HP 44730A Track/Hold with Signal Conditioning**
- **HP 44732A 120 Ohm Dynamic Strain Gage FET Multiplexer**
- **HP 44733A 350 Ohm Dynamic Strain Gage FET Multiplexer**

Description

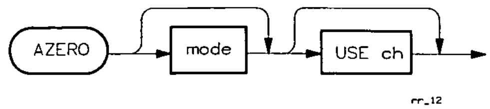
Perform an autozero on the specified channel.

Prerequisites

Command requires mainframe firmware revision 3.5 or greater. The system must not be scanning with the HP 44702A/B voltmeter when AZERO is executed.

Syntax

AZERO [*mode*] [USE *ch*]



Parameters

mode — Autozero control mode. Only *mode* is ONCE. Autozero is performed with channel inputs connected to an internal zero volt reference. After the AZERO command, amplifier contribution to channel offset errors is minimized.

USE *ch* — Specify channel to be used for the AZERO command. Channel number range = ES00-ES03.

Remarks

AZERO Does Not Null Channel Offset

The AZERO command does not null offset voltages present on the channel due to external sources. Refer to the NULL command for details on nulling channel offset errors.

Data Returned

None.

Related Commands

CAL, FUNC, NULL

Examples

Example: Performing Channel Autozero

This program uses FUNC to set channel 500 of an HP 44730A to amplify the input signal with gain = 10 and AZERO to autozero the channel. (The filter is not used.) Channel measurements are made with an HP 44702A/B voltmeter in slots 6 and 7 via the ribbon cable.

```
10  OUTPUT 709;"USE 500"              !Use ch 500
20  OUTPUT 709;"FUNC AMPLIFY,10"      !Ch 500 to amplify w/gain = 10
30  OUTPUT 709;"AZERO ONCE"           !Autozero on ch 500
40  OUTPUT 709;"USE 600"              !Use voltmeter in mf slot 6
```

AZERO (cont)

```
50  OUTPUT 709;"SCANMODE ON"            !Set Scanner mode
60  OUTPUT 709;"CONFMEAS DCV,500"       !Conf/meas DC volts, ch 500
70  ENTER 709;A                         !Enter reading
80  PRINT A                             !Display reading
90  END
```

For a 0.5 V, 5% input, a typical return is:

```
4.995
```

BEEP

• Mainframe

Description

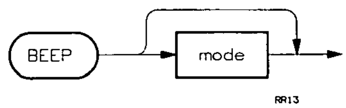
Enable/disable HP 3852A beeper mode. With beeper mode enabled, the mainframe will BEEP when an error is generated. With beeper mode disabled, error messages are displayed on the front panel but the BEEP is suppressed. Sending BEEP ONCE or BEEP with no parameter causes the mainframe to BEEP once.

Prerequisites

None

Syntax

BEEP [*mode*]



Parameters

mode Enables or disables the beeper mode. Power-on *mode* = ON. Default *mode* = ONCE.

mode	Definition
ON	Beeper mode enabled.
OFF	Beeper mode disabled.
ONCE	Beep once whether beeper mode enabled or disabled.

Remarks

Using BEEP ONCE

BEEP ONCE overrides the present ON/OFF state of the BEEP command and produces one BEEP. BEEP ONCE does not affect the enable/disable state.

Data Returned

None

Related Commands

None

Examples

Example: Produce One Beep

This program temporarily disables the beeper mode and produces one beep. The BEEP ONCE command overrides the BEEP OFF setting and causes one BEEP. Then, the *mode* returns to BEEP OFF.

```
10 OUTPUT 709;"BEEP OFF"     !Disable beeper mode
20 OUTPUT 709;"BEEP ONCE"    !Beep once and return to BEEP OFF mode
30 END
```

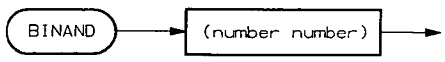
- **Binary Function**

Description

Numeric expression evaluated as a command parameter. Returns the value of a bit-by-bit logical AND of its arguments.

Syntax

BINAND (*number number*)



Parameters

number Number or parenthesized numeric expression which must evaluate within the range of -32768 to +32767.

Remarks

Within the HP 3852A processor, each bit in one *number* is logically ANDed with the corresponding bit in the other *number*, as shown in the following truth table.

num1 bit	num2 bit	result
0	0	0
0	1	0
1	0	0
1	1	1

Examples

Example: Using BINAND Function

This program computes the BINAND value of 9 and 12 and returns the result (8) to the controller CRT. The BINAND arithmetic is:

```
12 = 0000 0000 0000 1100
 9 = 0000 0000 0000 1001
--------------------------
 8 = 0000 0000 0000 1000
```

```
10 OUTPUT 709;"RST"                     !Reset HP 3852A
20 OUTPUT 709;"VREAD BINAND (9,12)"     !Compute and read BINAND value
30 ENTER 709;A                          !Enter value in controller
40 PRINT "BINAND Value = ";A            !Display value
50 END
```

A typical display for the BINAND of 12 and 9 is:

```
BINAND Value = 8
```

BINCMP

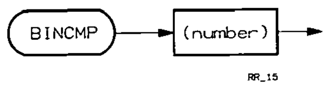
• **Binary Function**

Description

Numeric expression evaluated as a command parameter. Returns the bit-by-bit complement of its argument.

Syntax

BINCMP (*number*)



Parameters

number Number or parenthesized numeric expression that must evaluate within the range of -32768 to +32767.

Remarks

BINCMP Operation

Within the HP 3852A processor, each bit in the *number* is complemented and the resulting value returned.

Examples

Example: Using BINCMP Function

This program computes the BINCMP value of -13 and returns the result (12) to the controller CRT. The BINCMP arithmetic is:

```
-13 = 1111 1111 1111 0011
 12 = 0000 0000 0000 1100 = Complement
```

```
10 OUTPUT 709;"RST"                   !Reset HP 3852A
20 OUTPUT 709;"VREAD BINCMP (-13)"    !Compute and read BINCMP value
30 ENTER 709;A                        !Enter result to controller
40 PRINT "BINCMP Value = ";A          !Display result
50 END
```

A typical display for the BIMCMP of -13 is:

```
BINCMP Value = 12
```

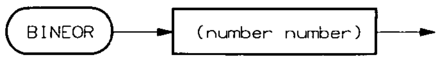
• **Binary Function**

Description

Numeric expression evaluated as a command parameter. Returns the value of the bit-by-bit exclusive-OR of its argument.

Syntax

BINEOR (*number number*)



rr_16

Parameters

number Number or parenthesized numeric expresssion which must evaluate within the range of -32768 to +32767.

Remarks

Within the HP 3852A processor, each bit in one *number* is exclusive-ORed with the corresponding bit in the second *number*, as shown in the following truth table.

num1 bit	num2 bit	result
0	0	0
0	1	1
1	0	1
1	1	0

Examples

Example: Using BINEOR Function

This program computes the BINEOR value of -9 and 12 and returns the result (-5) to the controller CRT. The BINEOR arithmetic is:

```
12 = 0000 0000 0000 1100
-9 = 1111 1111 1111 0111
------------------------
-5 = 1111 1111 1111 1011
```

```
10 OUTPUT 709;"RST"                      !Reset the HP 3852A
20 OUTPUT 709;"VREAD BINEOR (-9,12)"     !Compute and read BINEOR value
30 ENTER 709;A                           !Enter result to controller
40 PRINT "BINEOR Value = ";A             !Display result
50 END
```

A typical display for BINEOR of -9 and 12 is:

```
BINEOR Value = -5
```

BINIOR

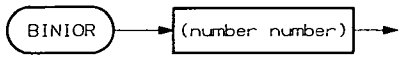
- **Binary Function**

Description

Numeric expression evaluated as a command parameter. Returns the value of the bit-by-bit inclusive-OR of its arguments.

Syntax

BINIOR (*number number*)



Parameters

number Number or parenthesized numeric expresssion which must evaluate within the range of -32768 to +32767.

Remarks

Within the HP 3852A processor, each bit in one *number* is inclusive-ORed with the corresponding bit in the second *number*, as shown in the following truth table.

num1 bit	num2 bit	result
0	0	0
0	1	1
1	0	1
1	1	1

Examples

<u>**Example: Using BINIOR Function**</u>

This program computes the BINIOR value of 9 and 12 and returns the result (13) to the controller CRT. The BINIOR arithmetic is:

```
12 = 0000 0000 0000 1100
 9 = 0000 0000 0000 1001
------------------------
13 = 0000 0000 0000 1101
```

```
10 OUTPUT 709;"RST"                    !Reset HP 3852A
20 OUTPUT 709;"VREAD BINIOR (9,12)"    !Compute and read BINIOR value
30 ENTER 709;A                         !Enter result to controller
40 PRINT "BINIOR Value = ";A           !Display result
50 END
```

A typical display for BINIOR of 9 and 12 is:

```
BINIOR Value = 13
```

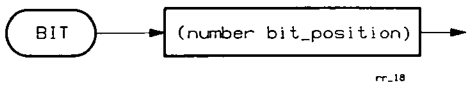
• **Binary Function**

Description

Numeric expression evaluated as a command parameter. Returns a 1 or 0 representing the value of the specified bit of its argument.

Syntax

BIT (*number bit_position*)



Parameters

number Number or numeric expression which must evaluate within the range of +32767 to -32768.

bit_position Number or parenthesized numeric expression which must evaluate within the range of 0 to 15.

Remarks

Within the HP 3852A processor, *number* is represented as a 16-bit two's-complement integer. Bit 0 is the least-significant bit and bit 15 is the most-significant bit.

Examples

Example: Using the BIT Function

This program determines the value (1 or 0) of bit 4 in the 16-bit two's complement representation of 17. Since 17 = 0000 0000 0001 0001, the program returns a "1".

```
10 OUTPUT 709;"RST"                !Reset HP 3852A
20 OUTPUT 709;"VREAD BIT (17,4)"   !Find and read value (1 or 0) of bit 4
30 ENTER 709;A                     !Enter value to controller
40 PRINT "Bit 4 value = ";A        !Display value
50 END
```

A typical display for this program is:

```
Bit 4 value = 1
```

BLOCKOUT

- **Mainframe**

Description

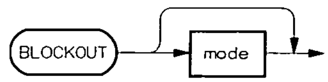
Block output mode. BLOCKOUT applies only to binary data. With BLOCKOUT OFF, binary data is output without preceding header. With BLOCKOUT ON, a data header precedes the data output.

Prerequisites

None

Syntax

BLOCKOUT [*mode*]



rr19

Parameters

mode Set block output mode for binary data output.

BLOCKOUT OFF (power-on)	Binary data (IN16, RL64, or PACK) is output without preceding header.
BLOCKOUT ON (default)	IEEE 728 Block A header precedes the binary data. Header contains four bytes: the # sign, the letter A, and two bytes which contain a 16-bit integer indicating the number of bytes to follow the header.

Remarks

BLOCKOUT Affects SYSOUT Data Header

With BLOCKOUT OFF, the "data format" parameter in the SYSOUT header is positive (+). With BLOCKOUT ON, this parameter is negative (-). Refer to SYSOUT command for details on SYSOUT data header.

Large Amounts of Data With BLOCKOUT ON

With BLOCKOUT ON, if >32760 bytes of data is to be returned, data is broken into several blocks (each with the IEEE 728 Block A header). All blocks except the last will have 32760 data bytes.

Data Returned

BLOCKOUT command returns no data. However, with BLOCKOUT OFF, binary data is output without a header. With BLOCKOUT ON, binary data is output with a data header.

Carriage Return/Line Feed (<CR> <LF>) terminates the block. Multiple data blocks have the form: <header> <data (32760 bytes)> <cr-lf> <header> <data> <cr-lf>...

Related Commands

SYSOUT and format parameter for any command returning data.

Examples

Example: Set Block Output Mode

This program line sets the HP 3852A for block output mode.

```
10 OUTPUT 709; "BLOCKOUT ON"
```

CAL

- **Mainframe**
- **HP 44701A Integrating Voltmeter**
- **HP 44702A/B High-Speed Voltmeter** (Scanner or System Mode)

Description

Calibrate the voltmeter. CAL is a service-related command. Refer to the HP 3852A Assembly Level Service Manual for calibration procedures.

- **HP 44730A Track/Hold with Signal Conditioning**
- **HP 44732A 120 Ohm Dynamic Strain Gage FET Multiplexer**
- **HP 44733A 350 Ohm Dynamic Strain Gage FET Multiplexer**

Description

Perform gain and offset calibration on the specified channel.

Prerequisites

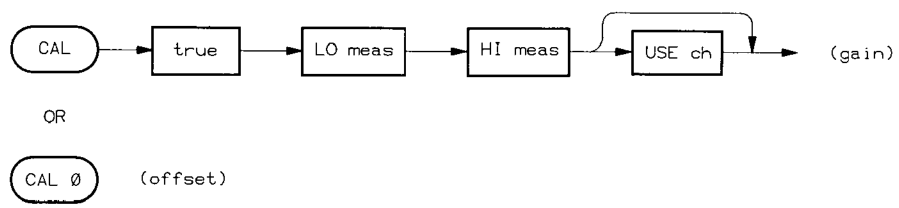
Command requires mainframe firmware revision 3.5 or greater. The system must not be scanning with the HP 44702A/B voltmeter when CAL is executed. Also, the channel to be calibrated must be set to the desired gain (1, 10, or 100) prior to the CAL command.

Syntax

CAL *true* **LO** *meas* **HI** *meas* [USE *ch*] (gain calibration)

OR

CAL 0 (offset calibration)



Parameters

true — The actual voltage input to the CAL terminals. For calibration, 9.00 Vdc $\leq$ (*true*)*(channel gain) $\leq$ 10.24 Vdc.

LO *meas* — Voltage measured on the channel after FUNC CALLO is executed. LO *meas* must be $\leq$(*true*)*(channel gain).

HI *meas* — Voltage value measured on the channel after FUNC CALHI is executed. HI *meas* must be $\geq$(*true*)*(channel gain).

USE *ch* — Specify channel to be used for the CAL command. Channel number range = ES00 through ES03.

Remarks

Use CAL 0 (Offset Calibration) to Zero Channel

The CAL command is used for gain calibration and requires a voltmeter and external calibration signal source. CAL 0 is used to zero the channel (offset calibration) and does not require a voltmeter or external zero reference. With CAL 0, the channel inputs are switched to an internal zero reference and the offset is automatically adjusted for minimum channel output.

Gain Parameter Required for FUNC CALLO and FUNC CALHI

When using FUNC CALLO and FUNC CALHI for gain calibration, the *gain* parameter must first be specified either by the FUNC [*gain*] parameter or by the GAIN command.

Calibration Procedure

Use the following steps for channel gain and offset calibration using an HP 44701A (or equivalent voltmeter) and an HP 44705A (or equivalent) multiplexer.

```
1.  Connect calibration source to CAL terminals.
2.  Also, connect source to HP 44705A channel.
3.  Set source for appropriate output.*
4.  Use GAIN to set channel gain.
5.  Use CAL 0 for offset calibration.

6.  Use CLOSE, CHREAD to measure source via HP 44705A.
7.  Set FUNC CALLO and measure channel LO value.
8.  Set FUNC CALHI and measure channel HI value.
9.  Use CAL for channel gain calibration.
10. Repeat (3) - (9) for other channels/gain settings.

* = 10.00 Vdc for gain = 1, 1.000 Vdc for gain = 10,
    0.1000 Vdc for gain = 100.
```

Data Returned

None.

Related Commands

FUNC, GAIN

Examples

Example: Offset and Gain Calibration

This program uses the CAL 0 command to perform gain and offset calibration on channel 500 of an HP 44730A, HP 44732A, or HP 44733A. The channel is set for amplification with gain of 10 and no filtering.

Channel measurements are made with an HP 44701A voltmeter in slot 1 via an HP 44705A multiplexer and the mainframe backplane. An external calibration source of about 1.00 Vdc is input to the CAL terminals. The source is also connected to channel 200 of an HP 44705A relay multiplexer. A description of the program follows.

Define CAL Parameter Variables (line 40)

Line 40 defines the mainframe variables to store the measured values. The actual value of the calibration source input is stored in INPT, the value resulting from FUNC CALLO is stored in LOW, and the value resulting from FUNC CALHI is stored in HIGH.

Perform Offset Calibration (lines 80 - 100)

In lines 80 - 100, channel 500 is set for amplification only with channel gain of 10. When the program executes, CAL 0 (line 100) performs an offset calibration on the channel.

Measure CAL *true* Value (lines 140 - 220)

In lines 140 - 220, the actual value of the calibration source is measured by the HP 44701A voltmeter (via the HP 44705A multiplexer in slot 200) and the value stored in INPT. This value will be used for the CAL *true* parameter value.

Measure LO *meas* Value (lines 260 - 320)

Lines 260 through 320 measure the channel output with the calibration circuitry set for minimum gain (about 0.99*channel gain). FUNC CALLO,10 sets channel 500 gain to 10 and selects the CAL terminals as the channel input. The channel output is measured and stored in variable LOW. This value will be used for the CAL LO *meas* parameter value. (Note that the [*gain*] value must be specified for FUNC CALLO, since the calibration is for channel gain of 10.)

Measure HI *meas* Value (lines 360 - 410)

Lines 360 through 410 measure the channel output with the calibration circuitry set for maximum gain (about 1.01*channel gain). FUNC CALHI,10 sets channel 500 gain to 10 and selects the CAL terminals as the channel input. The channel output is measured and stored in variable HIGH. This value will be used for the CAL HI *meas* parameter value. (Note that the [*gain*] value must be specified for FUNC CALHI, since the calibration is for channel gain of 10.)

Perform Gain Calibration (lines 450 - 460)

After the values to be used in the CAL command have been measured and stored, CAL performs a gain calibration on channel 500 using the stored values. CAL computes and stores the gain calibration constant for this gain setting and channel. For convenience, the measured value of the calibration source (CAL *true*), output low value (LO *meas*), and output high value (HI *meas*) are displayed.

NOTE

This program computes gain and offset calibration for channel 500 with gain = 10. To calibrate the channel for a gain of 1, change lines 90, 270, and 360 to reflect a gain of 1, change the calibration input to about 10.0 Vdc and rerun the program. To calibrate the channel for a gain of 100, change lines 90, 270, and 360 to reflect a gain of 100, change the calibration input to about 0.100 Vdc and rerun the program.

```
10   !
20   !Define mainframe variables
30   !
40   OUTPUT 709;"REAL INPT,LOW,HIGH"          !Define input, low, hi var
50   !
60   !Set up ch 500, make offset calibration
70   !
80   OUTPUT 709;"USE 500"                     !Use ch 500
90   OUTPUT 709;"FUNC AMPLIFY,10"             !Ch 500 to amplify w/gain 10
100  OUTPUT 709;"CAL 0"                       !Offset calibration
110  !
120  !Measure calibration source output
130  !
140  OUTPUT 709;"CLOSE 200,291"               !Close 44705A ch, iso relay
150  OUTPUT 709;"USE 100"                     !Use 44701A vm in slot 1
160  OUTPUT 709;"CONF DCV"                    !Conf for DC volts meas
170  OUTPUT 709;"TRIG SGL"                    !Trigger voltmeter
180  OUTPUT 709;"CHREAD 100 INTO INPT"        !Read/store input value
190  OUTPUT 709;"VREAD INPT"                  !Trans rdg to out buffer
200  ENTER 709;A                              !Enter rdg
210  PRINT "Input   =";A                      !Display input value
220  OUTPUT 709;"OPEN 200,291"                !Open 44705A ch, iso relay
230  !
240  !Measure ch 500 output low value
250  !
260  OUTPUT 709;"CLOSE 500,590"               !Close 44730A ch, iso relay
270  OUTPUT 709;"FUNC CALLO,10,USE 500"       !Set ch 500 to meas LO
280  OUTPUT 709;"TRIG SGL"                    !Trigger voltmeter
290  OUTPUT 709;"CHREAD 100 INTO LOW"         !Read/store low value
300  OUTPUT 709;"VREAD LOW"                   !Trans rdg to out buffer
310  ENTER 709;B                              !Enter rdg
320  PRINT "LO meas =";B                      !Display ch 500 LO value
330  !
340  !Measure ch 500 output high value
350  !
360  OUTPUT 709;"FUNC CALHI,10,USE 500"       !Set ch 500 to meas HI
370  OUTPUT 709;"TRIG SGL"                    !Trigger voltmeter
380  OUTPUT 709;"CHREAD 100 INTO HIGH"        !Read/store high value
390  OUTPUT 709;"VREAD HIGH"                  !Trans rdg to out buffer
400  ENTER 709;C                              !Enter rdg
410  PRINT "HI meas =";C                      !Display ch 500 high value
420  !
430  !Perform ch 500 gain calibration
440  !
450  OUTPUT 709;"CAL INPT,LO LOW,HI HIGH,USE 500"  !Gain calib on ch 500
460  OUTPUT 709;"OPEN 500,590"                !Open ch 500, iso relay
470  END
```

Since the channel gain is 10, a typical return for an approximate 1.00 Vdc calibration source input is as follows.

```
Input   = 1.001386   (input value)
LO meas = 9.91801    (output low value)
HI meas = 10.11938   (output high value)
```

• Mainframe

Description

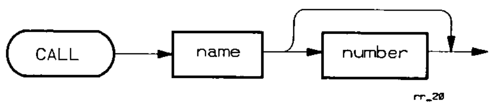
Call subroutine specified number of times.

Prerequisites

Specified subroutine must have been stored in HP 3852A mainframe memory. See SUB and SUBEND to create subroutines.

Syntax

CALL *name* [*number*]



Parameters

name Subroutine name as defined by the SUB command.

number Number of times subroutine to be called sequentially. Valid range is 1 to 2147483647. Numbers are rounded to the nearest integer. Default value = 1.

Remarks

Calling a Subroutine

When a subroutine is CALLed, the entire subroutine will execute before any command outside the subroutine can be executed. An error inside a subroutine aborts the subroutine. If a nested subroutine is aborted, the calling subroutine continues.

Subroutines with PAUSE Statement

Subroutines containing a PAUSE statement, cannot have *number* > 1.

Subroutine Arrays and Variables are Global

All arrays and variables are global. An array or variable defined within a subroutine is accessable outside the subroutine and vice-versa.

Data Returned

Data from a command which is part of a subroutine is returned to where CALL was entered (HP-IB or front panel). If CALL is entered from front panel, data is returned to the display. If CALL is entered from HP-IB, data is returned to HP-IB and to display if the DISP and MON modes are enabled.

Data from a subroutine called by ON ALRM, ON LMT, or ON INTR command is returned to where the ON command was entered.

Related Commands

SUB, SUBEND, STEP, CONT, PAUSE, ON event CALL name

Examples

Example: Creating and Calling Subroutine

```
 10 OUTPUT 709;"RST"            !Reset HP 3852A
 20 OUTPUT 709;"SUB BEEPER"     !Create subroutine BEEPER
 30 OUTPUT 709;"BEEP"
 40 OUTPUT 709;"SUBEND"         !End subroutine BEEPER
        .
        .
100 OUTPUT 709;"CALL BEEPER"    !Call BEEPER - HP 3852A will BEEP once
```

• Mainframe

Description

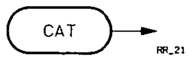
Returns a catalog list of all stored subroutines, arrays, and variables.

Prerequisites

None

Syntax

CAT



Parameters

None

Remarks

Potential Deadlock Problem

With INBUF OFF, the controller and the HP 3852A may deadlock if multiple commands (or terminators) are sent in a single command line and a command generates enough data to fill the output buffer. (CAT of 32 or more names will fill the output buffer). The best way to avoid potential deadlock is to send a single command per command line and read the results as soon as possible after a data-generating command is sent.

Turn FASTDISP OFF When Using CAT to Front Panel

Since the CAT command generates multiple "results", set FASTDISP OFF to be able to see all the strings on the HP 3852A display. The command returns quoted strings of 14 characters, including the quotes. However, quote marks are not shown on the display.

DELSUB, DELVAR, and SCRATCH Commands

The DELSUB and DELVAR commands recover storage used for subroutines and arrays (size is set to zero), but do not erase the name or type. SCRATCH, however, deletes all names in addition to recovering memory.

Data Returned

The CAT command returns the total number of subroutines, arrays, and variables stored, followed by one or more data lines which show the subroutine, array, or variable name, type, and size. Data is returned in reverse order of definition.

For a PACKED array, CAT returns the maximum number of bytes specified for the array if no data has been stored in the array. Once data is stored in the array, CAT returns the maximum number of readings which can be stored in the array in that format.

The maximum number of PACKED readings which can be stored is determined by dividing the maximum number of bytes by the number of bytes/reading for the data currently in the array. Note that different data formats can't be stored in the same PACKED array.

For example, a PACKED array with *max_index* = 99 contains 100 bytes of memory for storing data. If no data is in the array, CAT returns 100 (maximum number of bytes). If one PACKED reading is stored which requires 2 bytes, CAT returns 50 (= 100 bytes/2 bytes per reading).

The following table defines name, type, and size information for each category of data (subroutine, array, or variable). Data returned by CAT to the controller or to the front panel also includes the total number of subroutines, arrays, and variables stored in the HP 3852A memory.

CATEGORY	NAME	TYPE	SIZE
subroutine	sub name	SUB	number bytes used for sub
INTEGER array	array name	IARR	number of array elements
REAL array	array name	RARR	number of array elements
PACKED array	array name	PARR	max number of bytes (no data)
			or
		PARR	max number of readings (w/data)
INTEGER variable	var name	INT	0
REAL variable	var name	REAL	0

Related Commands

DIM, FASTDISP OFF, INTEGER, PACKED, REAL, SUB, DELSUB, DELVAR, SCRATCH

Examples

Example: Results Returned by CAT

The following program shows the results returned by the CAT command for subroutine BEEPER, INTEGER array A, and REAL variable R (a total of three subroutines, arrays, and variables defined).

```
 10 OUTPUT 709;"SCRATCH"         !Delete all sub, var names
 20 INTEGER N
 30 DIM Name$[20],Kind$ [20]
 40 OUTPUT 709;"SUB BEEPER"      !Start of sub BEEPER
 50 OUTPUT 709;"BEEP"
 60 OUTPUT 709;"SUBEND"          !End of sub BEEPER
 70 OUTPUT 709;"REAL R"          !Define R as real variable
 80 OUTPUT 709;"INTEGER A(20)"   !Define A as integer array (21 elements)
 90 OUTPUT 709;"CAT"
100 ENTER 709;N                  !Number of subs, arrays, vars in memory
110 PRINT "Subs, arrays, and variables in HP 3852A memory =  ";N
120 PRINT
130 PRINT "    Name       Type     Size"
140 PRINT
150 FOR I = 1 TO N
160 ENTER 709;Name$,Kind$
170 PRINT Name$,Kind$
180 NEXT I
190 END
```

Since the order of definition in the program was BEEPER, R, and A, the order of information returned by CAT is A, R, BEEPER. The following typical return shows that A is an INTEGER array with 21 elements, R is a REAL variable with size 0 (one element), and BEEPER is a subroutine which uses 110 bytes.

```
Subs, arrays, and variables in HP 3852A memory = 3

      Name              Type      Size

  "A         "          "IARR      21"
  "R         "          "REAL       0"
  "BEEPER    "          "SUB      110"
```

- **HP 44788A HP-IB Controller**

Description

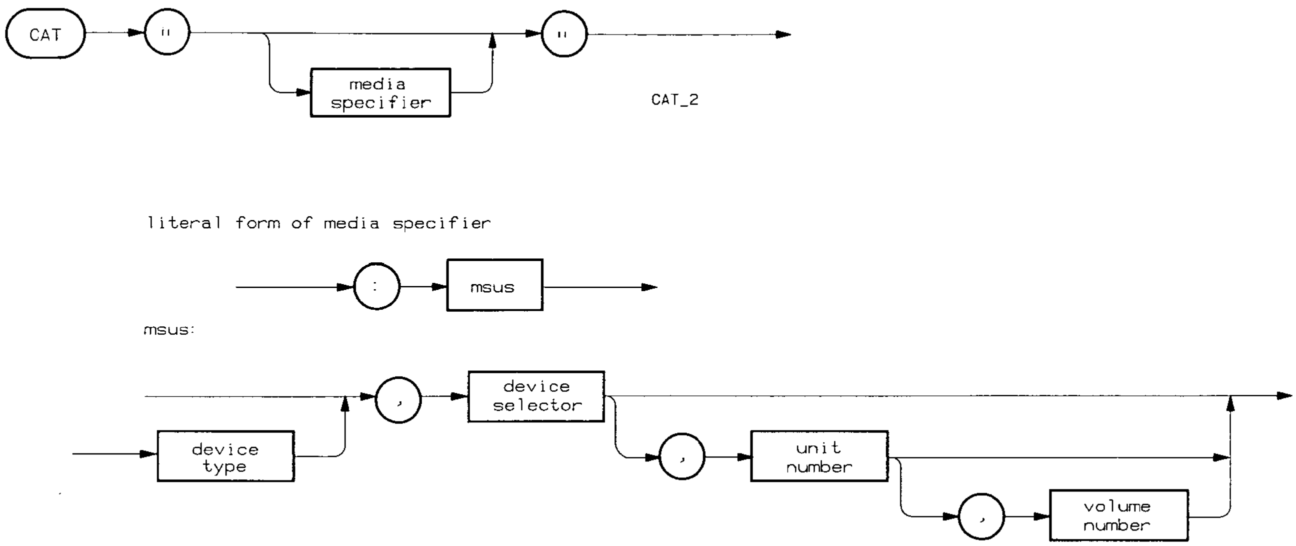
This statement lists all the contents of a mass storage directory on the HP 3852A display.

Prerequisites

Requires firmware revision 3.5 or greater.

Syntax

CAT "[*media specifier*]"



Parameters

media specifier The name of the mass storage device containing the directory. If no media specifier is stated (" "), the current MSI is catalogued.

Remarks

Media Specifier Declared with MSI

If the media specifier has been declared with the MSI command, the CAT command with a null string in the quotes will catalogue the directory specified by MSI. If quotes are omitted, the HP 3852A mainframe CAT command will be executed instead of the HP-IB Controller CAT command. Refer to the HP 3852A Command Reference Manual for information on the mainframe CAT command.

Data Returned

A directory entry is listed for each file on the media. The catalog shows the name of each file, the file type and length, and the number of bytes per logical record. The types recognized in BASIC are ASCII, BDAT (BASIC data), PROG (BASIC program), CODE (PASCAL code file), TEXT (PASCAL text file), or

SYSTM (language system). An ID number is listed for any unrecognized file types. The starting location (address) is also shown.

Related Commands

CREATE ASCII, CREATE BDAT, MSI

Examples

Example: Catalog Directory

The following program lines set catalog directory on drive 1 of disc at HP-IB address 00 with the HP 44788A in slot 3 of the mainframe. The first statement is entered from the HP 3852A front panel, while statements for lines 120 and 130 are entered from the system controller. In line 130, note that using CAT without quotes will execute the mainframe CAT command but not the HP 44788A CAT command.

```
CAT ":,300,1"                        !Catalog disc drive 300, drive 1
```

or

```
120 OUTPUT 709;"MSI ':,300,1'"       !Specify disc drive 300, drive 1
130 OUTPUT 709;"CAT ' '"             !Catalog current MSI
```

CHREAD

- **HP 44701A Integrating Voltmeter**
- **HP 44702A/B High-Speed Voltmeter (Scanner or System Mode)**

Description

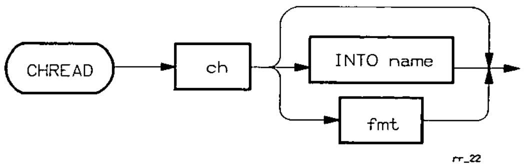
Transfer reading from voltmeter.

Prerequisites

The voltmeter addressed must have been triggered and data must be available before CHREAD will return a reading. When using the HP 44702A/B, RDGS SYS must be set.

Syntax

CHREAD *ch* [INTO *name*] or [*fmt*]



Parameters

ch Voltmeter slot number. See Glossary.

INTO *name* See Glossary.

fmt See Glossary. Default fmt is RASC.

Remarks

CHREAD Waits Until Results are Available

If a reading is not available when the CHREAD command is executed, the command waits until the reading is available.

CHREAD and Interrupts

CHREAD clears interrupts for the HP 44701A. CHREAD clears interrupts for the HP 44702A/B if the command reads the last reading in the voltmeter buffer.

Data Returned

The CHREAD command returns a reading from the specified voltmeter slot.

Related Commands

XRDGS, RDGS, TRIG

Examples

Example: Transfer Reading to Controller

This program configures the voltmeter in mainframe slot 1 to measure DC voltage using the external terminals on the voltmeter. The reading is then transferred to the controller and displayed.

```
10 OUTPUT 709;"RST"          !Reset the HP 3852A
20 OUTPUT 709;"USE 100"      !Use voltmeter in mainframe slot 1
30 OUTPUT 709;"CONF DCV"     !Select DC volts
40 OUTPUT 709;"TERM EXT"     !Select external terminals
50 OUTPUT 709;"TRIG SGL"     !Trigger once, take reading
60 OUTPUT 709;"CHREAD 100"   !Transfer reading to output buffer
70 ENTER 709;A               !Enter reading into A
80 PRINT A                   !Display reading
90 END
```

Example: Transfer Reading to Mainframe Memory

This program configures the voltmeter in mainframe slot 1 to measure DC voltage using the external terminals on the voltmeter. The reading is then transferred to variable R in mainframe memory.

```
10  OUTPUT 709;"RST"                !Reset the HP 3852A
20  OUTPUT 709;"USE 100"            !Use voltmeter in mainframe slot 1
30  OUTPUT 709;"CONF DCV"           !Select DC volts
40  OUTPUT 709;"TERM EXT"           !Select external terminals
50  OUTPUT 709;"REAL R"             !Create REAL variable R
60  OUTPUT 709;"TRIG SGL"           !Trigger once, take reading
70  OUTPUT 709;"CHREAD 100,INTO R"  !Transfer reading into R
80  END
```

CHREAD

• **HP 44715A 5-Channel Counter/Totalizer**

Description

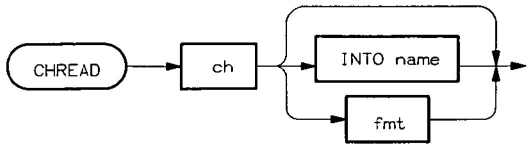
Read from channel.

Prerequisites

The channel addressed must have been triggered and data must be available before CHREAD will return a reading.

Syntax

CHREAD *ch* [INTO *name*] or [*fmt*]



Parameters

ch — Address of channel. Channel number range (depends on HP 44715A hardware configuration) = ES00 through ES04.

INTO *name* — See Glossary.

fmt — See Glossary. Default format = RASC

Remarks

Triggering and Interrupts for CHREAD

For TOTAL, TOTALM, UDC, UDCM, CD and CDM, the channel must be triggered prior to CHREAD. In addition, for the RAT, PER and PERD functions, and for Frequency configuration, the measurement must be available before CHREAD will return the reading. CHREAD clears interrupts on the channel for RAT, PER, and PERD functions, and for Frequency configuration.

Data Returned

For TOTAL, TOTALM, UDC, UDCM, CD, and CDM, the current count is returned without disturbing the counting function. For RAT, CHREAD returns the ratio of the A channel to the B channel counts. For PER and PERD, the period (in seconds) is returned. For Frequency configuration, the frequency (in Hz) is returned.

Related Commands

TRIG, FUNC, XRDGS, CHREADZ

Examples

Example: Read Channel Counts

This program reads the counts on channel 3 of a counter in slot 5 of the mainframe. The counter is set to the TOTAL function.

```
10 OUTPUT 709;"RST"           !Reset HP 3852A and HP 44715A
20 OUTPUT 709;"USE 503"       !Use ch 3 in slot 5 of mainframe
30 OUTPUT 709;"FUNC TOTAL"    !Set TOTAL function on channel
40 OUTPUT 709;"TRIG SGL"      !Single-trigger channel
50 OUTPUT 709;"CHREAD 503"    !Read counts on channel
60 ENTER  709;Counts          !Return count to controller
70 PRINT "Counts = ";Counts   !Display counts
80 END
```

A typical return (10 counts on channel) is:

```
Counts = 10
```

- **HP 44721A 16-Channel Digital Input**
- **HP 44722A 8-Channel Digital Input**

Description

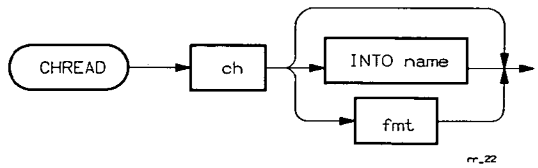
Read channel state or channel count.

Prerequisites

None

Syntax

CHREAD *ch* [INTO *name*] or [*fmt*]



Parameters

ch — Address of digital input channel. Channel number range = ES00 through ES31 (HP 44721A) or ES00 through ES15 (HP 44722A).

INTO *name* — See Glossary.

fmt — See Glossary. Default format is RASC.

Remarks

Data Returned

Channel numbers ES00 - ES15 (ES00 - ES07 for HP 44722A) returns count on channel. Channel numbers ES16 - ES31 (ES08 - ES15 for HP 44722A) returns state of channel. Channel state is returned as a 16-bit integer with a value of 1 (HIGH for DC input, ON for AC input) or 0 (LOW for DC input, OFF for AC input).

Channel count is number of edge transitions specified by EDGE since the channel was enabled by the EDGE command, unless preset with CNTSET or zeroed by the CHREADZ command. Counts are returned as a 32-bit two's complement integer.

Related Commands

READ, CHREADZ, XRDGS, EDGE, CNTSET

Examples

Example: Read Channel State and Count

The following program reads the state and the number of counts on channel 3 of a 16-channel digital input in slot 5 of the mainframe.

```
10 OUTPUT 709;"RST"                          !Reset HP 3852A and Dig. In.
20 OUTPUT 709;"CHREAD 503"                   !Read channel counts
30 ENTER 709;Counts                          !Enter counts
40 PRINT "Counts on channel 3 = ";Counts     !Display counts
50 OUTPUT 709;"CHREAD 519"                   !Read channel state
60 ENTER 709;State                           !Enter state
70 PRINT "State of channel 3 = ";State       !Display state
80 END
```

Typical returns (10 counts and channel state HIGH or ON) follow.

```
Counts on channel 3 = 10
State of channel 3 = 1
```

CHREAD

- **HP 44723A 16-Channel High-Speed Digital Sense/Control**

Description

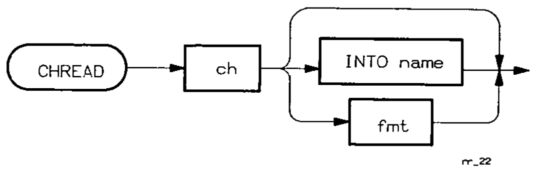
Returns the state of the specified input channel as read from the second rank input register.

Prerequisites

Requires mainframe firmware revision 3.0 or greater.

Syntax

CHREAD *ch* [**INTO** *name*] or [*fmt*]



Parameters

ch — Address of input channel. Channel number range = ES00 through ES15.

INTO *name* — See Glossary.

fmt — See Glossary. Default format is IASC.

Remarks

Data Returned

Returns the state (0 for LOW, 1 for HIGH) of the specified channel. Data is read from the second rank input register.

Related Commands

CHREADM, READ, READM

Examples

Example: Read Input Channel State

This program reads the current state of channel 203 of an HP 44723A in slot 2 of the mainframe. Since TRIG INT and SRTRIG INT are set, when CHREAD is executed the channel 203 input state is sampled and the result stored in the first rank input register. Next, the first rank input register contents are copied to the second rank input register. Then, CHREAD reads the second rank input register and returns the state of channel 203.

```
10  OUTPUT 709;"USE 203"              !USE ch is 203
20  OUTPUT 709;"TRIG INT"             !Sample input when CHREAD executed
30  OUTPUT 709;"SRTRIG INT"           !Copy data when CHREAD executed
40  OUTPUT 709;"CHREAD 203"           !Read ch 203 state
50  ENTER 709;A                       !Enter ch 203 state
60  PRINT "Channel 203 state =";A     !Display ch 203 state
70  END
```

A typical return for channel 203 state HIGH follows.

```
Channel 203 state = 1
```

- **HP 44721A 16-Channel Digital Input**
- **HP 44722A 8-Channel Digital Input**

Description

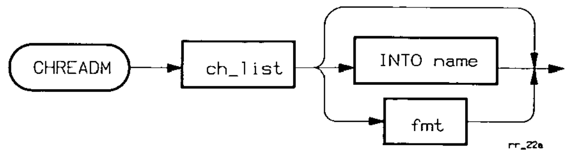
Read state of specified channel(s).

Prerequisites

Requires mainframe firmware revision 3.0 or greater.

Syntax

CHREADM *ch_list* [INTO *name*] or [*fmt*]



Parameters

ch_list — Channel list of digital input channel(s). Channel number range = ES15 through ES31 (HP 44721A) or ES08 through ES15 (HP 44722A).

INTO *name* — See Glossary.

fmt — See Glossary. Default format is IASC.

Remarks

Data Returned

CHREADM returns the state of the channel(s) specified by *ch_list*. The state is returned as a 1 (HIGH for DC input or ON for AC input) or a 0 (LOW for DC input or OFF for AC input) for each specified channel.

Related Commands

CHREAD, CHREADZ, CNTSET, EDGE, READ, XRDGS

Examples

Example: Read Channel States

This program reads the states of channels 500 and 502 through 505 for an HP 44721A in slot 5 of the mainframe.

```
10  INTEGER A(0:4)                     !Dim controller array
20  OUTPUT 709;"CHREADM 516,518-521"   !Read channel states
30  ENTER 709;A(*)                     !Enter counts
40  PRINT "Channel States"             !Display header
50  PRINT A(*)                         !Display states
60  END
```

A typical return follows. For this example, channels 500, 502, and 505 states are HIGH while channels 503 and 504 states are LOW.

```
Channel States
 1    1    0    0    1
```

CHREADM

- **HP 44723A 16-Channel High-Speed Digital Sense/Control**

Description

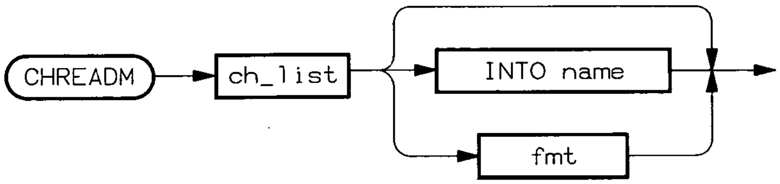
Returns the state of specified input channel(s) as read from the second rank input register.

Prerequisites

Requires mainframe firmware revision 3.0 or greater.

Syntax

CHREADM *ch_list* [**INTO** *name*] or [*fmt*]



r.chr

Parameters

ch_list Address of input channel(s). Channel number range = ES00 through ES15.

INTO *name* See Glossary.

fmt See Glossary. Default format is IASC.

Remarks

Data Returned

Returns the state (0 for LOW, 1 for HIGH) of each of the channels specified in *ch_list*. Data is read from the second rank input register.

Related Commands

CHREAD, READ, READM

Examples

Example: Read Input Channel States

This program reads the states of channels 203, 204, and 205 of an HP 44723A in slot 2 of the mainframe. Since TRIG INT and SRTRIG INT are set, when CHREADM is executed the channel 203 through 205 states are sampled and the result stored in the first rank input register. Next, the first rank input register contents are copied to the second rank input register. Then, CHREADM reads the contents of the second rank input register and returns the states of channels 203 through 205.

```
10  DIM A(0:2)                       !Dimension controller array
20  OUTPUT 709;"USE 200"             !USE ch is 200
30  OUTPUT 709;"TRIG INT"            !Sample inputs when CHREADM executed
40  OUTPUT 709;"SRTRIG INT"          !Copy data when CHREADM executed
50  OUTPUT 709;"CHREADM 203-205"     !Read ch 203-205 states
60  ENTER 709;A(*)                   !Enter states
70  PRINT A(*)                       !Display states
80  END
```

A typical return for channels 203 and 204 HIGH and channel 205 LOW follows.

where the leftmost bit is the channel 203 state and the rightmost bit is the channel 205 state.

1 1 0

CHREADZ

• HP 44715A 5-Channel Counter/Totalizer

Description

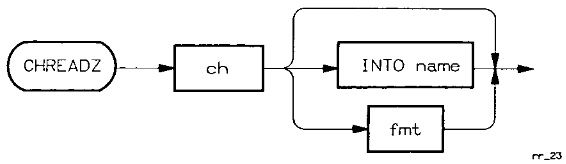
Read and zero channel count.

Prerequisites

The channel addressed must have been triggered and data must be available before CHREADZ will return a reading.

Syntax

CHREADZ *ch* [INTO *name*] or [*fmt*]



Parameters

ch Address of channel. Channel number range (depends on HP 44715A hardware configuration) = ES00 through ES04.

INTO *name* See Glossary.

fmt See Glossary. Default format is RASC

Remarks

Triggering and Interrupts for CHREADZ

For the TOTAL, TOTALM, UDC, UDCM, CD, and CDM functions, the channel must be triggered prior to CHREAD. In addition, for the RAT, PER, and PERD functions, the measurement must be available before CHREADZ will return the reading. CHREADZ clears interrupts on the channel for the RAT, PER, and PERD functions. CHREADZ may not be used for the Frequency configuration.

Data Returned

For the TOTAL, UDC, and CD functions, the current count is returned and the channel is then zeroed. For the TOTALM, UDCM, and CDM functions, the current count is returned without disturbing the count. For RAT, CHREAD returns the ratio of the A channel to the B channel counts. For PER and PERD, the period (in seconds) is returned.

Related Commands

TRIG, FUNC, XRDGS, CHREAD

Examples

Example: Read and Zero Channel Counts

The following program reads and zeroes the counts on channel 3 of a counter in slot 5 of the mainframe. The counter is set to the TOTAL function.

```
 10 OUTPUT 709;"RST"              !Reset HP 3852A and HP 44715A
 20 OUTPUT 709;"USE 503"          !Use ch 3 in slot 5
 30 OUTPUT 709;"FUNC TOTAL"       !Set TOTAL function on ch
 40 OUTPUT 709;"TRIG SGL"         !Single-trigger ch
 50 OUTPUT 709;"CHREADZ 503"      !Read/zero counts on ch
 60 ENTER  709;Counts             !Enter counts
 70 PRINT "Counts = ";Counts      !Display counts
 80 OUTPUT 709;"CHREADZ 503"      !Read/zero counts again
 90 ENTER 709;New                 !Return new counts
100 PRINT "New counts = ";New     !Display new counts
110 END
```

Typical returns (10 counts to first CHREADZ and 5 counts from first CHREADZ to second CHREADZ) follow.

```
Counts = 10
New counts = 5
```

- **HP 44721A 16-Channel Digital Input**
- **HP 44722A 8-Channel Digital Input**

Description

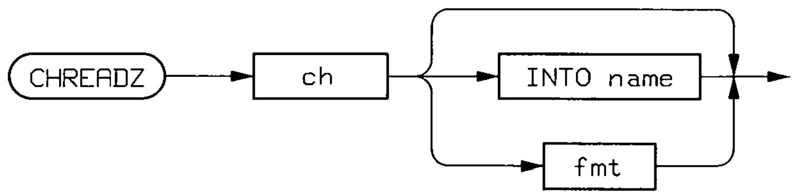
Read and zero channel count.

Prerequisites

Data must be available before the command will return a reading.

Syntax

CHREADZ *ch* [INTO *name*] or [*fmt*]



Parameters

ch — Address of digital input channel. Channel number range = ES00 through ES15 (HP 44721A) or ES00 through ES07 (HP 44722A).

INTO *name* — See Glossary.

fmt — See Glossary. Default format is RASC.

Remarks

Data Returned

Channel count is number of edge transitions specified by EDGE since the channel was enabled by the EDGE command, unless preset with CNTSET or zeroed by a previous CHREADZ command. CHREADZ returns channel count in 32-bit two's complement integer.

Related Commands

READ, CHREAD, XRDGS, EDGE, CNTSET

Examples

Example: Read and Zero Channel Count

This program reads and zeroes the number of counts on channel 3 of a digital input in slot 5 of the mainframe.

```
10 OUTPUT 709;"RST"            !Reset HP 3852A and HP 44721A/22A
20 OUTPUT 709;"USE 503"        !Use ch 3 in slot 5 in mainframe
30 OUTPUT 709;"EDGE HL"        !Count positive edges
40 OUTPUT 709;"CHREADZ 503"    !Read/zero counts on channel
50 ENTER 709;Counts            !Enter channel counts
60 PRINT "Counts = ";Counts    !Display counts
70 END
```

A typical return (10 counts on channel) follows.

```
Counts = 10
```

• **HP 44723A 16-Channel High-Speed Digital Sense/Control**

Description

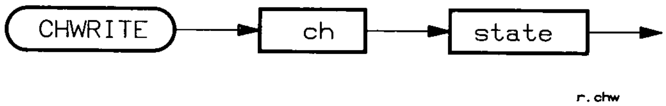
Write the state (0/1) to the specified output channel. State is written to the first rank output register. A second rank output trigger is required to transfer the new state to the user outputs.

Prerequisites

Requires mainframe firmware revision 3.0 or greater.

Syntax

CHWRITE *ch state*



Parameters

ch Address of output channel. Channel number range = ES16 through ES31.

state Specifies state for output channel. Any nonzero integer for *state* between -32768 and +32767 is interpreted as a "1" (HIGH). Zero is interpreted as a "0" (LOW).

Remarks

Data Returned

None

Related Commands

CHWRITEM, SRTRIG

Examples

Example: Write to Output Channel

This program sets channel 203 of an HP 44723A in slot 2 of the mainframe HIGH. When CHWRITE is executed, a "1" (set channel HIGH) is written to the first rank output register. Then, when an external trigger is received, the new state is transferred to the second rank output register and to the channel 203 output terminals.

```
10  OUTPUT 709;"USE 219"          !Use ch is 203
20  OUTPUT 709;"SRTRIG EXT"       !External source for sec rank out trig
30  OUTPUT 709;"CHWRITE 219,1"    !Write "1" to first rank output reg
40  END
```

CHWRITE

- **HP 44724A 16-Channel Digital Output**
- **HP 44725A 16-Channel General Purpose Switch**
- **HP 44728A 8-Channel Relay Actuator**
- **HP 44729A 8-Channel Power Controller**

Description

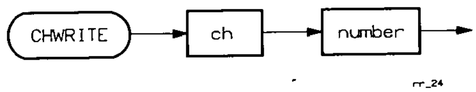
Write number to open or close single channel.

Prerequisites

None

Syntax

CHWRITE *ch number*



Parameters

ch Address of channel. Channel number range = ES00 through ES15 (HP 44724A and HP 44725A) or ES00 through ES07 (HP 44728A and HP 44729A).

number *number* = 0 opens channel addressed by *ch* while *number* <> 0 closes channel.

Remarks

CHWRITE Can Address Only One Channel

A single CHWRITE command can address only one channel. All non-addressed channels remain in previous state.

Data Returned

None

Related Commands

WRITE, CLOSE, OPEN

Examples

Example: Using CHWRITE to Set Digital Output Channels

The following program lines open channel 5 and close channel 13 of a 16-channel digital output in slot 2 of the mainframe.

```
100  OUTPUT 709; "CHWRITE 205, 0"   !Open channel 5 in slot 2
110  OUTPUT 709; "CHWRITE 213, 1"   !Close channel 13 in slot 2
```

• HP 44723A 16-Channel High-Speed Digital Sense/Control

Description

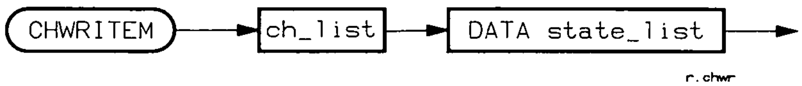
Write the state (0/1) to each specified output channel in the channel list. States are written to the first rank output register. A second rank output trigger is required to transfer the new state to the user outputs.

Prerequisites

Requires mainframe firmware revision 3.0 or greater.

Syntax

CHWRITEM *ch_list* **DATA** *state_list*



Parameters

ch_list — Address of output channel list. Channel number range = ES16 through ES31.

DATA *state_list* — Specifies the state(s) (0/1) for corresponding output channel(s) specified by *ch_list*. Any nonzero integer between -32768 and +32767 for *state_list* is interpreted as a "1" (HIGH). Zero is interpreted as a "0" (LOW). CHWRITEM uses one item from *state_list* for each channel OR channel range in *ch_list*.

Remarks

Data Returned

None

Related Commands

CHWRITE, SRTRIG

Examples

Example: Write to Output Channels

This program sets channels 200 and 203 HIGH and sets channels 206 through 209 LOW for an HP 44723A in slot 2 of the mainframe. Note that the first "1" in the DATA *state_list* parameter applies to channel 200, the second "1" to channel 203, and the "0" to channels 206 through 209. Since SRTRIG INT is set, the new channel states are automatically copied to the second rank output register and to the user output terminals.

```
10 OUTPUT 709;"USE 216"                               !Use channel is 200
20 OUTPUT 709;"SRTRIG INT"                            !Trans data on CHWRITEM
30 OUTPUT 709;"CHWRITEM 216,219,222-225 DATA 1,1,0"  !Write data
40 END
```

CHWRITEM

- **HP 44724A 16-Channel Digital Output**
- **HP 44725A 16-Channel General Purpose Switch**
- **HP 44728A 8-Channel Relay Actuator**
- **HP 44729A 8-Channel Power Controller**

Description

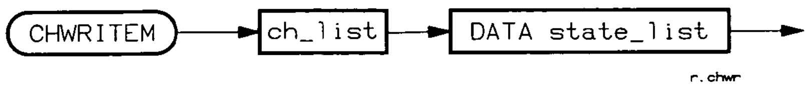
Write the state (0/1) to open or close specified channel(s).

Prerequisites

Requires mainframe firmware revision 3.0 or greater.

Syntax

CHWRITEM *ch_list* **DATA** *state_list*



Parameters

ch_list — Address of channel list. Channel number range = ES00 through ES15 (HP 44724A and HP 44725A) or ES00 through ES07 (HP 44728A and HP 44729A).

DATA *state_list* — Specifies the state (open or closed) for the corresponding channels specified by *ch_list*. A "0" for *state_list* opens the associated channel(s), while any nonzero integer between -32768 and +32767 closes the associated channel(s). CHWRITEM uses one item from DATA *state_list* for each channel OR channel range in *ch_list*.

Remarks

Data Returned

None

Related Commands

CHWRITE, CLOSE, OPEN, WRITE

Examples

Example: Setting Digital Output Channel States

This program line opens channels 200 and 204 and closes channels 205 through 209 of an HP 44724A in slot 2 of the mainframe. Note that the "1" parameter for DATA *state_list* applies to channels 205 through 209.

```
        .
        .
100  OUTPUT 709;"CHWRITEM 200,204,205-209 DATA 0,0,1"  !Open ch 200, 204,
        .                                              !close ch 205-209
        .
```

• **HP 44788A HP-IB Controller**

Description

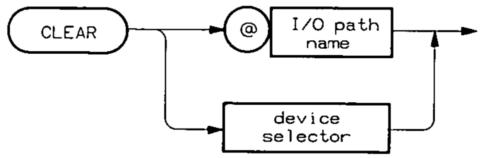
This statement allows the mainframe to put selected HP-IB devices into a defined, device-dependant state. The bus is reconfigured and the SDC (Selected Device Clear) message is sent to the device which is addressed by the LAG message.

Prerequisites

Requires firmware revision 3.5 or greater.

Syntax

CLEAR *@I/O path name* or *device selector*



CLEAR

Parameters

@I/O path name The name of the bidirectional path assigned to a device to be cleared.

device selector The HP-IB select code (i.e. ESnn) for the device that is to be cleared. The local HP-IB address for the mainframe is always ESnn, where E = extender number (mainframe = 0), S = slot and nn is the device address.

Remarks

Data Returned

None

Related Commands

ASSIGN

Examples

Example: Clear Device

This program segment assigns I/O path @HP3852 to the HP 3852A, then assigns I/O path @DVM to device 211, and then clears the device.

```
10 OUTPUT 709;"ASSIGN @HP3852 TO 709"      !Assign I/O path for HP 3852A
20 OUTPUT @HP3852;"ASSIGN @DVM TO 211"     !Assign I/O path @DVM
30 OUTPUT @HP3852;"CLEAR @DVM"             !Clear device 211
              .
              .
```

CLOSE

- **HP 44705A 20-Channel Relay Multiplexer**
- **HP 44705H 20-Channel High-Voltage Relay Multiplexer**
- **HP 44706A 60-Channel Relay Multiplexer**
- **HP 44708A 20-Channel Relay Multiplexer/TC**
- **HP 44708H 20-Channel High-Voltage Relay Multiplexer/TC**

- **HP 44709A 20-Channel FET Multiplexer**
- **HP 44710A 20-Channel FET Multiplexer/TC**

- **HP 44711A 24-Channel High-Speed FET Multiplexer**
- **HP 44712A 48-Channel High-Speed FET Multiplexer**
- **HP 44713A 24-Channel High-Speed FET Multiplexer/TC**

Description

Close multiplexer channels. CLOSE closes a single multiplexer channel or a list of multiplexer channels. CLOSE is a low-level command intended for individual switch control in special signal-routing applications.

CAUTION

The CLOSE command does not close channels in a break-before-make fashion. Therefore, the command can and will cause damage to the multiplexer accessory (relay or FET) and external system if it is used to force one channel open by closing another. This applies to channels in the same bank, in separate banks tied together by the tree relays, and to the tree relays themselves.

Before a channel is closed with the CLOSE command, use the OPEN command to open the channel that is currently closed. This prevents any two channels from being closed at the same time and reduces the risk of damaging your equipment.

NOTE

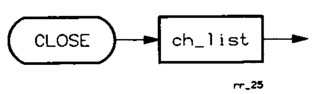
Using CLOSE is not the easiest way to do routine measurements since the tree switches are not automatically configured for the measurement.

Prerequisites

None

Syntax

CLOSE *ch_list*



Parameters

ch_list Address of channel list. See Glossary.

HP 44705A/HP 44705H/HP 44709A (20-Channel Multiplexers)

CC Range	Definitions
0 - 9	Bank A Switches
10 - 19	Bank B Switches
90	Isolation Relays (HP 44709A only)
91	Sense Bus Tree Switch (Bank A)
92	Sense Bus Tree Switch (Bank B)
93	Source Bus Tree Switch (Bank A)
94	Source Bus Tree Switch (Bank B)

HP 44706A (60-Channel Multiplexer)

CC Range	Definitions
0 - 59	Bank Switches/Sense Bus Tree Switch
91	Source Bus Tree Switch

HP 44708A/HP 44708H/HP 44710A (20-Channel Multiplexers/TC)

CC Range	Definitions
0 - 9	Bank A Switches
10 - 19	Bank B Switches
90	Isolation Relays (HP 44710A only)
91	Sense Bus Tree Switch (both banks)
92	Sense Bus Tree Switch (thermistor)
93	Source Bus Tree Switch (thermistor)
94	Source Bus Tree Switch (both banks)

HP 44711A/HP 44713A (24-Channel Multiplexers)

CC Range	Definitions
0 - 11	Bank A Switches
12 - 23	Bank B Switches
90	Isolation Relays
91	Source Bus Tree Switch (Bank A or Bank B)
92	Sense Bus Tree Switch (Bank A or Bank B)
93	2-Wire Ohms Configuration
94	4-Wire Ohms Configuration

HP 44712A (48-Channel Multiplexer)

CC Range	Definitions
0 - 47	Bank Switches
90	Isolation Relays
91	Source Bus Tree Switch
92	Sense Bus Tree Switch
93	2-wire ohms configuration

Remarks

Controlling Multiplexer Channels

Channels are closed in the order listed in the *ch_list* parameter, but are NOT closed in break-before-make (BBM) fashion. Only one channel per bank can be closed at a time. Closing a second channel in a bank opens any previously closed channel (see CAUTION). For the HP 44706, HP 44711A, HP 44712A, and HP 44713A, only one measurement channel per accessory can be closed (except for 4-wire ohms measurements with the HP 44711A/HP 44713A). All channels (and isolation relays for FET multiplexers) are opened at power-on. CLOSE waits for each channel closure to complete before advancing to the next channel in the *ch_list*.

Channel Closure Conditions

Channels closed by the CLOSE command remain closed until one of the following conditions occurs:

- Channels are opened with an OPEN command.
- Another channel in the same bank of a 20-channel accessory is closed (see CAUTION).
- Another channel on a 24, 48, or 60-channel accessory is closed (see CAUTION).
- A second tree switch of the same type (Source or Sense) on the same accessory is closed (see CAUTION).
- A RST or RST [*slot*] command is executed.
- A MEAS or SCAN command uses the same bank or accessory for tree switch (except Isolation Relays).
- Overvoltage on backplane (Isolation Relays).

Controlling Tree Switches

Tree switches control signal flow between the multiplexers and the HP 3852A backplane. Sense Bus tree switches provide voltage connections to the backplane for making voltage measurements. Source Bus tree switches provide current source connections (+I and -I) to the backplane for making resistance measurements.

Two tree switches of the same type cannot be closed simultaneously (e.g., only one of the Sense Bus tree switches can be closed at a time). Closing a second tree switch opens any previously closed tree switch of the same type.

Controlling Isolation Relays (FET Multiplexers)

FET multiplexers contain isolation relays which allow the accessory to be isolated from the HP 3852A backplane. Isolation relays can be used to reduce leakage current on the backplane for critical measurements or when the backplane is used for voltages greater than FET multiplexer specifications. These relays will open automatically if an overvoltage situation occurs.

Channel 90 controls the isolation relays for the FET multiplexers. For normal operation, the isolation relays must be closed to enable the accessory to be used for measurements on the backplane.

Data Returned

None

Related Commands

CLOSE?, OPEN, SCAN, MEAS

Examples

Example: Using CLOSE to Make Voltage Measurements

This program uses CLOSE to make DC voltage measurements using a 20-channel relay multiplexer in slot 2 of the mainframe and a voltmeter in slot 0 of the mainframe. Note that the OPEN command is used to open the currently closed channel before the next channel is closed.

NOTE

When making measurements using a FET multiplexer, the isolation relays (channel 90) must also be closed to connect the tree switches to the backplane.

```
 10 REAL Volts(0:1)                  !Declare controller array
 20 OUTPUT 709;"RST"                 !Reset HP 3852A and multiplexer
 30 OUTPUT 709;"USE 0"               !Use voltmeter in slot 0
 40 OUTPUT 709;"CONF DCV"            !Configure for DC volts
 50 OUTPUT 709;"CLOSE 200,291"       !Close ch 0 and tree switch (slot 2)
 60 OUTPUT 709;"TRIG SGL"            !Trigger voltmeter once, take reading
 70 OUTPUT 709;"CHREAD 0"            !Transfer reading to output buffer
 80 ENTER 709;Volts(0)               !Enter reading into controller
 90 OUTPUT 709;"OPEN 200"            !Open ch 0 before closing ch 1
100 OUTPUT 709;"CLOSE 201"           !Close channel 1 (slot 2)
110 OUTPUT 709;"TRIG SGL"            !Trigger voltmeter, take reading
120 OUTPUT 709;"CHREAD 0"            !Transfer reading to output buffer
130 ENTER 709;Volts(1)               !Enter reading into controller
140 OUTPUT 709;"OPEN 201,291"        !Open all channels
150 PRINT Volts(*)                   !Display readings
160 END
```

- **HP 44717A 10 Bridge 120Ω Static Strain Gage Relay Multiplexer**
- **HP 44718A 10 Bridge 350Ω Static Strain Gage Relay Multiplexer**
- **HP 44719A 10 Bridge 120Ω Static Strain Gage FET Multiplexer**
- **HP 44720A 10 Bridge 350Ω Static Strain Gage FET Multiplexer**

Description

Close strain gage multiplexer channels. CLOSE closes a single multiplexer channel or a list of multiplexer channels. This low-level command is necessary for accessing the "internal" strain gage channels and the tree/isolation relays associated with the configuration diagnostics. However, it can also be used to close bridge completion channels 0-9.

CAUTION

The CLOSE command does not close channels in a break-before-make fashion. Therefore, the command can and will cause damage to the multiplexer accessory (relay or FET) and external system if it is used to force one channel open by closing another. This applies to channels in the same bank, in separate banks tied together by the tree relays, and to the tree relays themselves.

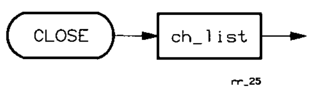
Before a channel is closed with the CLOSE command, use the OPEN command to open the channel that is currently closed. This prevents any two channels from being closed at the same time and reduces the risk of damaging your equipment.

Prerequisites

The strain gage accessories require mainframe firmware revision 2.0 or greater.

Synatx

CLOSE *ch_list*



Parameters

ch_list Address of the bridge completion channel, internal channel, or tree/isolation relay. Channel and diagnostic definitions for the strain gage accessories are given below:

Channel	Definition
0-9	Bridge Completion Channels. Closing channels 0-9 and 91 (and 90 if FET multiplexer) with the CLOSE command ties the bridge completion channels to the sense bus on the mainframe backplane. When performing 2-wire ohms measurements using low-level commands (e.g. OPEN, CLOSE), channel 94 must also be closed.
10	Bridge Excitation Voltage. Closing internal channels 10 and 91 (and 90 if FET multiplexer) with the CLOSE command enable the bridge excitation voltage to be

Channel	Definition
	measured using low-level commands (e.g. TRIG, CHREAD). Note that specifying STRVEX in the CONFMEAS, MEAS, and MONMEAS commands performs the same measurement.
11, 12	Shunt Verification. These channels are closed automatically by specifying either STRQTEN or STRQCOMP in the CONFMEAS or MEAS command. Specifying STRQTEN (tension shunt) closes channel 11 which places a resistor in parallel with the upper leg internal bridge resistor to simulate a known value of tensile strain. Specifying STRQCOMP (compression shunt) closes channel 12 which places a resistor in parallel with the strain gage on the bridge completion channel to simulate a known value of compressive strain.
13	Gage Isolation. Closing internal channels 13, 91, and 94 (and 90 if FET multiplexer) enable you to check the isolation between the strain gage and the test specimen. If the specimen is a conductor, the parallel combination of isolation on all bridge completion channels (0-9) is checked. If the specimen is not a conductor, only the isolation of channel 0 is checked.
14	Guard Voltage. Closing internal channels 14 and 91 (and 90 if FET multiplexer) enable you to measure the guard voltage to ensure that the guard lead is biased to the necessary potential to reduce stray leakage currents.
15, 16	Internal Half Bridge Voltage. Closing internal channels 15 or 16, and 91 (and 90 if FET multiplexer) enable you to measure the voltage on the upper and lower legs of the internal half bridge. This checks the stability of the 500Ω bridge completion resistors. Channel 15 measures the upper bridge leg and channel 16 measures the lower bridge leg.
17, 18, 19	Leadwire Resistance. Closing internal channels 17 or 18 or 19, and 91 (and 90 if FET multiplexer) enable you to indirectly determine the leadwire resistance of a 1/4 bridge arrangement by measuring the voltage on the leadwire from the strain gage to the bridge completion circuit. Knowing the leadwire resistance allows leadwire desensitization corrections to be made.

Channel	Definition
90	FET Isolation Relay. The FET strain gage multiplexers contain an isolation relay that isolates the accessory from the mainframe backplane. In addition to closing the tree switches (channels 91 and 94), the isolation relay must also be closed to tie the multiplexer channels to the backplane. (CONFMEAS, MEAS, and MONMEAS close channel 90 automatically.)
91	Volts Tree Relay/Switch. Closing channel 91 ties all accessory channels (internal and bridge completion) to the sense bus on the mainframe backplane. (Channel 91 is closed automatically by the CONFMEAS, MEAS, and MONMEAS commands.)
94	Resistance Tree Relay/Switch. Closing channel 94 ties all channels to the source bus on the mainframe backplane. This enables a voltmeter accessory to perform a 4-wire ohms measurement of gage isolation or 2-wire ohms measurements on the bridge completion channels. (Channel 94 is closed automatically by the CONFMEAS, MEAS, and MONMEAS commands when an OHMs function is specified.)

Remarks

Closing Internal Strain Gage Channels

Internal channels are closed in the order specified by the *ch_list* parameter. Channels are not closed in a break-before-make (BBM) fashion. CLOSE waits for each channel closure to complete before advancing to the next channel in *ch_list*.

Bridge completion channels closed by the CLOSE command remain closed until opened with the OPEN command, forced open by another channel closing on the same bank (see CAUTION), or following the execution of the SCAN or MEAS command. Resetting the instrument (RST) or resetting the accessory (RST <slot>) also opens all closed channels.

Data Returned

None

Related Commands

OPEN, SCAN, SADV

Examples

Example: Using CLOSE to Measure the Internal Half Bridge Voltages

This program shows how the CLOSE command is used in conjunction with low-level voltmeter commands to measure the internal half bridge voltages to determine the stability of the 500Ω bridge completion resistors. Other commands are included to show the context in which the CLOSE command is used, and when a channel is closed relative to the measurement.

```
10 REAL Half_bridgeup               !REAL variables are declared
20 REAL Half_bridgelow
30 OUTPUT 709;"USE 700"             !use the voltmeter in slot 7
40 OUTPUT 709;"CONF DCV"            !configures the voltmeter
50 OUTPUT 709;"CLOSE 415,491"       !closes channel corresponding to
60                                  !the upper bridge leg
70 OUTPUT 709;"TRIG SGL"            !triggers the voltmeter
80 OUTPUT 709;"CHREAD 700"          !retrieves reading from voltmeter
90 ENTER 709;Half_bridgeup          !enter reading into controller
100 OUTPUT 709;"OPEN 415"           !open previously closed channel
110 OUTPUT 709;"CLOSE 416"          !closes channel corresponding to
120                                 !the lower bridge leg
130 OUTPUT 709;"TRIG SGL"           !triggers the voltmeter
140 OUTPUT 709;"CHREAD 700"         !retrieves reading from voltmeter
150 ENTER 709;Half_bridgelow        !enter reading into controller
160 OUTPUT 709;"OPEN 416,491"       !opens closed channels
170 Print Half_bridgeup             !upper bridge leg voltage displayed
180 PRINT Half_bridgelow            !lower bridge leg voltage displayed
190 END
```

Typical internal bridge voltages for a bridge excitation voltage of 5.4V are:

```
2.7051
2.705
```

Example: Using CLOSE to Measure the Guard Voltage

This program shows how the CLOSE command is used to close the internal channels necessary for the Guard Voltage diagnostic.

```
10 REAL Guard_voltage               !a REAL variable is declared
20 OUTPUT 709;"USE 700"             !use the voltmeter in slot 7
30 OUTPUT 709;"CONF DCV"            !configures the voltmeter
40 OUTPUT 709;"CLOSE 414,490,491"   !closes guard voltage channel
50                                  !(FET accessory)
60 OUTPUT 709;"TRIG SGL"            !triggers the voltmeter
70 OUTPUT 709;"CHREAD 700"          !retrieves reading from voltmeter
80 ENTER 709;Guard_voltage
90 PRINT Guard_voltage              !guard voltage is displayed
100 OUTPUT 709;"OPEN 414,490,491"   !opens channels 14, 90, and 91
110 END
```

The guard voltage should be 1/2 the bridge excitation voltage ±2%. A typical reading based on this program is:

```
2.521222
```

CLOSE

- **HP 44724A 16-Channel Digital Output**
- **HP 44725A 16-Channel General Purpose Switch**
- **HP 44728A 8-Channel Relay Actuator**
- **HP 44729A 8-Channel Power Controller**

Description

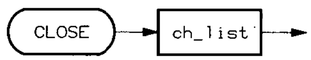
Close digital output and actuator channels.

Prerequisites

None

Syntax

CLOSE *ch_list*



Parameters

ch_list Address of channel list. See Glossary. Channel number range = ES00 - ES15 (HP 44724A and HP 44725A) or ES00 - ES07 (HP 44728A and HP 44729A).

Remarks

Channels Not Addressed by CLOSE

All channels not addressed by CLOSE remain in previous state. Channels are closed in the order given in *ch_list*. CLOSE waits for each channel closure to complete before advancing to the next channel in the *ch_list*.

Data Returned

None

Related Commands

CHWRITE, WRITE, OPEN

Examples

Example: Using CLOSE to Close Digital Output Channels

This program line uses the CLOSE command to close channel 0 and channels 3 through 6 of a digital output in slot 4 of mainframe.

```
160  OUTPUT 709;"CLOSE 400,403-406"
```

- **HP 44730A 4-Channel Track/Hold with Signal Conditioning**
- **HP 44732A 4-Bridge 120 Ohm Strain Gage FET Multiplexer**
- **HP 44733A 4-Bridge 350 Ohm Strain Gage FET Multiplexer**

Description

Close a single channel or a list of channels. CLOSE is a low-level command intended for individual switch control in special signal-routing applications.

CAUTION

CLOSE does not close channels in a break-before-make fashion. Therefore, the command can and will cause damage to an HP 44730A, HP 44732A, or HP 44733A and to the external system if CLOSE is used to force one channel open by closing another channel.

Before a channel is closed with the CLOSE command, use the OPEN command to open the channel which is currently closed. This prevents any two channels from being closed at the same time and reduces the risk of damaging your equipment.

NOTE

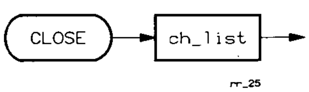
Using CLOSE is not the easiest way to do routine measurements since the isolation relays are not automatically configured for the measurement.

Prerequisites

Mainframe firmware revision 3.5 or greater is required to use CLOSE with an HP 44730A, HP 44732A, or HP 44733A accessory. The system must not be scanning with the HP 44702A/B voltmeter when CLOSE is executed.

Syntax

CLOSE *ch_list*



Parameters

ch_list Address of channel list. See Glossary.

CC Range	Definitions
0 - 3	User Inputs
4 - 7	Excitation Voltages
90	Isolation Relay

Remarks

Ribbon Cable Operation

CLOSE (and CLOSE? and OPEN) can be used with an HP 44730A, HP 44732A, or HP 44733A even if the ribbon cable is connected between the accessory and an HP 44702A/B voltmeter.

Controlling Accessory Channels

Channels are closed in the order listed in the *ch_list* parameter, but are NOT closed in break-before-make (BBM) fashion. Only one channel per accessory can be closed at a time. Closing a second channel in an accessory opens any previously closed channel (however, see CAUTION). All channels (and isolation relays) are opened at power-on or following a reset. CLOSE waits for each channel closure to complete before advancing to the next channel in the channel list.

Channel Closure Conditions

HP 44730A, HP 44732A, or HP 44733A channels closed by the CLOSE command remain closed until one of the following conditions occurs:

- Channels are opened with an OPEN command.
- A RST or RST [*slot*] command is executed.
- Overvoltage on backplane (Isolation Relays).

Controlling Isolation Relays

The HP 44730A, HP 44732A, and HP 44733A accessories contain isolation relays which allow the accessory to be isolated from the mainframe backplane. Isolation relays can be used to reduce leakage current on the backplane for critical measurements or when the backplane is used for voltages greater than FET multiplexer specifications. The isolation relays will open automatically if an overvoltage situation occurs.

Channel ES90 controls the isolation relays for the HP 44730A, HP 44732A, and HP 44733A accessories. For normal operation, the isolation relays must be closed to enable the accessory to be used for measurements via the backplane.

Data Returned

None

Related Commands

CLOSE?, MEAS, OPEN, SCAN

Examples

Example: Voltage Measurements Using CLOSE

This program uses an HP 44701A voltmeter to measure the +4.6 V source voltage for channel 500 of an HP 44730A, HP 44732A, or HP 44733A via the mainframe backplane. The CLOSE command is used to close the channel and the isolation relay. For this program, the EXC and EXC SENSE terminals must be connected.

NOTE

Although this measurement is via the backplane, since CLOSE and OPEN are used, the measurement can still be made even if the ribbon cable is connected between an HP 44730A, HP 44732A, or HP 44733A and an HP 44702A/B.

```
10  OUTPUT 709;"USE 100"                  !Use 44701A vm in slot 100
20  OUTPUT 709;"CLOSE 504,590"            !Close ch 500, iso relay
30  OUTPUT 709;"CONF DCV"                 !Set DC volts measurement
40  OUTPUT 709;"TRIG SGL"                 !Trigger voltmeter
50  OUTPUT 709;"CHREAD 100"               !Read ch 500 exc voltage
60  ENTER 709;A                           !Enter ch 500 exc voltage
70  PRINT "Ch 500 source =;A;"Volts"      !Display ch 500 exc voltage
80  OUTPUT 709;"OPEN 504,590"             !Open ch 500, iso relay
90  END
```

A typical return follows.

```
Ch 500 source = 4.62116 Volts
```

- **HP 44705A 20-Channel Relay Multiplexer**
- **HP 44705H 20-Channel High-Voltage Relay Multiplexer**
- **HP 44706A 60-Channel Relay Multiplexer**
- **HP 44708A 20-Channel Relay Multiplexer/TC**
- **HP 44708H 20-Channel High-Voltage Relay Multiplexer/TC**

- **HP 44709A 20-Channel FET Multiplexer**
- **HP 44710A 20-Channel FET Multiplexer/TC**

- **HP 44711A 24-Channel High-Speed FET Multiplexer**
- **HP 44712A 48-Channel High-Speed FET Multiplexer**
- **HP 44713A 24-Channel High-Speed FET Multiplexer/TC**

Description

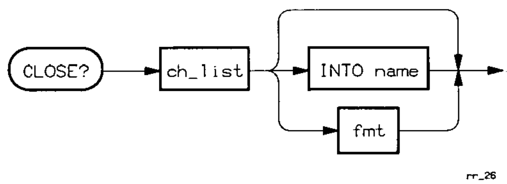
Query multiplexer channels. CLOSE? determines state of multiplexer channels. Command returns a number which shows if a channel is open or closed and to which bus the channel is connected.

Prerequisites

None

Syntax

CLOSE? *ch_list* [INTO *name*] or [*fmt*]



Parameters

ch_list Address of channel list. See Glossary.

HP 44705A/HP 44705H/HP 44709A (20-Channel Multiplexers)

CC Range	Definitions
0 - 9	Bank A Switches
10 - 19	Bank B Switches
90	Isolation Relays (HP 44709A only)
91	Sense Bus Tree Switch (BANK A)
92	Sense Bus Tree Switch (BANK B)
93	Source Bus Tree Switch (BANK A)
94	Source Bus Tree Switch (BANK B)

HP 44706A (60-Channel Multiplexer)

CC Range	Definitions
0 - 59	Bank Switches/Source Bus Tree Switch
91	Source Bus Tree Switch

HP 44708A/HP 44708H/HP 44710A (20-Channel Multiplexers/TC)

CC Range	Definitions
0 - 9	Bank A Switches
10 - 19	Bank B Switches
90	Isolation Relays (HP 44710A only)
91	Sense Bus Tree Switch (bank switches)
92	Sense Bus Tree Switch (thermistor)
93	Source Bus Tree Switch (thermistor)
94	Source Bus Tree Switch (bank switches)

HP 44711A/HP 44713A (24-Channel Multiplexers)

CC Range	Definitions
0 - 11	BANK A Switches
12 - 23	BANK B Switches
90	Isolation Relays
91	Source Bus Tree Switch (Bank A or Bank B)
92	Sense Bus Tree Switch (Bank A or Bank B)
93	2-Wire Ohms Configuration
94	4-Wire Ohms Configuration

HP 44712A (48-Channel Multiplexer)

CC Range	Definitions
0 - 47	Bank Switches
90	Isolation Relays
91	Source Bus Tree Switch
92	Sense Bus Tree Switch
93	2-wire ohms configuration

INTO *name* See Glossary.

fmt See Glossary. Default format is IASC.

Remarks

Potential HP 3852A/Controller Deadlock

With INBUF OFF, the controller and the HP 3852A may deadlock if multiple commands (or terminators) are sent in a single command line and a command generates enough data to fill the output buffer (CLOSE? can fill the output buffer). The best way to avoid potential deadlock is to send a single command per command line and to read the results as soon as possible after a data-generating command is sent.

FET Multiplexer Channel Considerations

When CLOSE? is used with FET multiplexers, the state of the Isolation Relays (channel 90) is ignored.

Data Returned

CLOSE? returns a 16-bit integer indicating the present state of the multiplexer channels, as shown.

20-Channel/24-Channel/48-Channel Multiplexers

Data Returned	Channel State
0	Channel Open
1	Channel Closed
2	Channel Closed - connected to Sense Bus
3	Channel Closed - connected to Source Bus
4	Channel Closed - connected to Both Buses

60-Channel Relay Multiplexer Accessories

Data Returned	Channel State
0	Channel Open
1	Channel Closed (Valid only for Source Bus tree switch - Channel 91)
2	Channel Closed - connected to Sense Bus
4	Channel Closed - connected to Sense Bus and Source Bus

Related Commands

CLOSE, OPEN

Examples

Example: Reading Channel State

This program uses the CLOSE? command to read the state of channels 0 through 4 of a 20-channel multiplexer in slot 2 of the mainframe.

```
10 OUTPUT 709;"RST 200"          !Reset multiplexer in slot 2
20 INTEGER State(0:4)            !Create INTEGER array State
30 OUTPUT 709;"CLOSE 203,291"    !Close ch 3 and Sense Bus tree sw (ch 91)
40 OUTPUT 709;"CLOSE? 200-204"   !Query channels 0 through 4
50 ENTER 709;State(*)            !Enter results into State
60 PRINT State(*)                !Display results
70 OUTPUT 709;"OPEN 203,291"     !Open channels
80 END
```

Typical output (channel 3 closed - connected to Sense Bus)

```
0    0    0    2    0
```

- HP 44724A 16-Channel Digital Output
- HP 44725A 16-Channel General Purpose Switch
- HP 44728A 8-Channel Relay Actuator
- HP 44729A 8-Channel Power Controller

Description

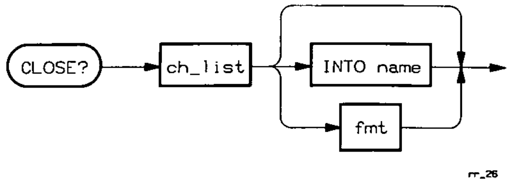
Returns state (open/closed) of each channel specified by *ch list*.

Prerequisites

None

Syntax

CLOSE? *ch_list* [**INTO** *name*] or [*fmt*]



Parameters

ch_list — Address of channel list. See Glossary. Channel number range = ES00 - ES15 (HP 44724A and HP 44725A) or ES00 - ES07 (HP 44728A and HP 44729A).

INTO *name* — See Glossary.

fmt — See Glossary. Default format is IASC.

Remarks

Potential Deadlock Problem

With INBUF OFF, the controller and the HP 3852A may deadlock if multiple commands (or terminators) are sent in a single command and a command generates enough data to fill the output buffer (CLOSE? can fill the output buffer). One way to avoid potential deadlock is to send a single command per command line and to read the results as soon as possible after a data-generating command is sent.

Data Returned

CLOSE? returns a 16-bit integer indicating the state (open/closed) of the channel, where 0 = channel open and 1 = channel closed.

Related Commands

READ, CLOSE, OPEN

Examples

Example: Using CLOSE? to Query Digital Output Channels

The following program uses CLOSE? to query all channels of an HP 44724A 16-Channel Digital Output in slot 2 of mainframe.

```
10 INTEGER State(0:15)                   !Declare INTEGER array
20 OUTPUT 709;"CLOSE? 200-215"           !Query all channels
30 ENTER 709;State(*)                    !Return channel states
40 PRINT "Channel states = ";State(*)    !Display states
50 END
```

A typical return for a 16-channel digital output with channels 2, 5, and 13 closed follows. The first number in the string = channel 0 state and the last number = channel 15 state.

```
Channel States = 0 0 1 0 0 1 0 0 0 0 0 0 0 1 0 0
```

- **HP 44730A 4-Channel Track/Hold with Signal Conditioning**
- **HP 44732A 4-Bridge 120 Ohm Strain Gage FET Multiplexer**
- **HP 44732A 4-Bridge 350 Ohm Strain Gage FET Multiplexer**

Description

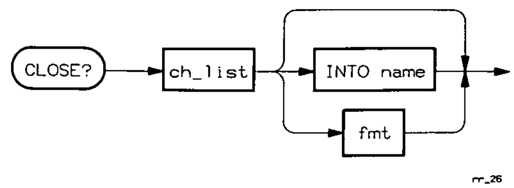
Query HP 44730A, HP 44732A, or HP 44733A channel state. CLOSE? returns a number which shows if a channel (or isolation relay) is open or closed and if the channel is connected to the sense bus.

Prerequisites

Mainframe firmware revision 3.5 or greater is required to use CLOSE? with an HP 44730A, HP 44732A, or HP 44733A accessory. The system must not be scanning with an HP 44702A/B voltmeter when CLOSE? is executed.

Syntax

CLOSE? *ch_list* [**INTO** *name*] or [*fmt*]



Parameters

ch_list Address of channel list. See Glossary.

CC Range	Definitions
0 - 3	User Inputs
4 - 7	Excitation Voltages
90	Isolation Relay

INTO *name* See Glossary.

fmt See Glossary. Default format for CLOSE? is IASC.

Remarks

Ribbon Cable Operation

CLOSE? (and CLOSE and OPEN) can be used with an HP 44730A, HP 44732A, or HP 44733A even if the ribbon cable is connected between the accessory and an HP 44702A/B voltmeter.

Potential HP 3852A/Controller Deadlock

With INBUF OFF, the controller and the HP 3852A may deadlock if multiple commands (or terminators) are sent in a single command line and a command generates enough data to fill the output buffer (CLOSE? can fill the output buffer). The best way to avoid potential deadlock is to send a single command per command line and to read the results as soon as possible after a data-generating command is sent.

Data Returned

CLOSE? returns an integer indicating the present state of the accessory channels, as shown.

Data Returned	Channel State	Channel Range
0	Channel Open	ES00 - ES07
1	Channel Closed (Isolation Relay)	ES90 only
2	Channel Closed (connected to Sense Bus)	ES00 - ES07

Related Commands

CLOSE, OPEN

Examples

Example: Reading Channel State

This program uses CLOSE? to read the state of channel 500 of an HP 44730A, HP 44732A, or HP 44733A. Note that although this measurement is via the backplane, since CLOSE, OPEN, and CLOSE? are used, the measurement can still be made even if the ribbon cable is connected between an HP 44730A, HP 44732A, or HP 44733A and an HP 44702A/B.

```
10 OUTPUT 709;"CLOSE 500,590"    !Close ch 500 and isolation relay
20 OUTPUT 709;"CLOSE? 500"       !Query ch 500 state
30 ENTER 709;A                   !Enter ch 500 state
40 PRINT A                       !Display ch 500 state
50 OUTPUT 709;"OPEN 500,590"     !Open ch 500 and isolation relay
60 END
```

Since channel 500 is closed and connected to the sense bus, a typical return is:

```
2
```

• Mainframe

Description

Clear the HP 3852A.

Prerequisites

None

Syntax

CLR

rr_27

Parameters

None

Remarks

The CLR command clears the HP 3852A. Sending CLR is equivalent to pressing the front panel CLEAR key or sending the HP-IB DCL (device clear) or SDC (selected device clear) command. CLR sets the following conditions:

NOTE

CLR will not be recognized until all previously entered commands have been executed. However, pressing the front panel CLEAR key or sending the HP-IB DCL or SDC commands will abort any currently executing commands or subroutines.

- Clears partially entered commands (HP-IB and front panel).
- Disables event interrupt recognition (DISABLE INTR SYS), limit interrupts (DISABLE LMT), and alarm interrupts (DISABLE ALRM). Accessory interrupts and channel logging remain enabled (if previously enabled).
- Sets RQS ON and RQS NONE (masks all bits in the HP 3852A status register).
- Clears the service request bit (bit 6) in the status register, the SRQ annunciator in the display, and the HP-IB SRQ line.
- Clears the HP-IB input and output buffers.
- If display enabled, displays READY on front panel.
- Attempts to <u>minimize</u> other state changes (relay settings, output states, memory values, etc.)

Data Returned

None

Related Commands

RST, CLROUT, DISABLE, RQS, STEP, PAUSE

Examples

```
OUTPUT 709;"CLR"   !Clears HP 3852A at address 9 when command executes.

CLEAR 7        !Immediately clears all devices (DCL) on bus (select code 7).

CLEAR 709      !Immediately clears device (SDC) at address 9 (select code 7).
```

• Mainframe

Description

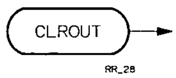
Clear HP-IB output buffer.

Prerequisites

None

Syntax

CLROUT



Parameters

None

Remarks

Using INBUF and OUTBUF with CLROUT

With OUTBUF OFF, it is usually not necessary to clear the output buffer, since new data will overwrite existing data in the buffer.

With INBUF ON, commands following the CLROUT command may be accepted and stored before CLROUT is executed. If the commands return data, the data returned may be the old data, the new data, or a combination of the two. See the Example "Clearing the Output Buffer" for details.

CLROUT Clears the DAV Bit in the Status Register

When executed, the CLROUT command will clear the DAV bit (bit 0) in the HP 3852A status register.

Data Returned

None

Related Commands

OUTBUF, INBUF, CLR

Examples

Example: Clearing the Output Buffer

As noted in Remarks, when using CLROUT with commands which return data, it is possible to receive old data, new data, or a combination. Two programs follow. The first program shows one way this can happen, while the second program shows a way to eliminate the possibility of incorrect data returns.

This Program May Return Incorrect Data

This program will return the results from channel 0 in slot 0 or channel 0 in slot 1 (or a mixture of both), depending on how quickly the controller gets to the ENTER statement vs. how quickly the HP 3852A gets to the CHREAD 100 statement. (This problem can occur even if CLROUT is not used).

```
10 OUTPUT 709;"RST"            !Reset the HP 3852A
20 OUTPUT 709;"INBUF ON"       !Set input buffer ON
30 OUTPUT 709;"CHREAD 0"       !Read channel 0 in slot 0
40 OUTPUT 709;"CLROUT"         !Clear output buffer
50 OUTPUT 709;"CHREAD 100"     !Read channel 0 in slot 1
60 ENTER 709;A                 !Read results
70 PRINT A                     !Display results
80 END
```

This Program Eliminates Incorrect Data Returns

To eliminate the possibility of incorrect data returns, the following program is better than the previous program. In this program the output buffer is cleared with the CLROUT command as before. However, the Wait for Ready statements in lines 30 and 40 halt the program until the buffer is clear. This ensures that the data in the output buffer is from CHREAD 100, not from CHREAD 0 (or a combination of data).

```
10 OUTPUT 709;"RST"                !Reset the HP 3852A
20 OUTPUT 709;"INBUF ON"           !Set input buffer ON
30 OUTPUT 709;"CHREAD 0"           !Read channel 0 in slot 0
40 OUTPUT 709;"CLROUT"             !Clear output buffer
50 WHILE NOT BIT (SPOLL (709),4))  !Wait for ready
60 END WHILE
70 OUTPUT 709;"CHREAD 100"         !Read channel 0 in slot 1
80 ENTER 709;A                     !Enter channel 0 read
90 PRINT A                         !Display channel 0 read
100 END
```

• **HP 44702A/B High-Speed Voltmeter** **(Scanner Mode only & TERM RIBBON)**

Description

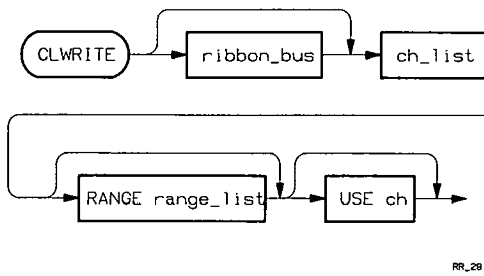
Sets channel and range lists to be scanned and used by the voltmeter.

Prerequisites

The voltmeter must be set to Scanner Mode (SCANMODE ON). TERM RIBBON and SCTRIG HOLD must also be set.

Syntax

CLWRITE [*ribbon_bus*] *ch_list* [RANGE *range_list*] [USE *ch*]



Parameters

ribbon_bus Specifies dedicated interface cable connections: SENSE (default), COM, or SEP.

ribbon_bus	Description
SENSE	Sets the interface cable for voltmeter SENSE bus only. Voltmeter current source is disconnected. Used for DC voltage measurements only.
COM	Both the voltmeter SENSE bus and the current source bus are connected in the multiplexer accessory. Used for 2-wire ohms measurements only.
SEP	Both the voltmeter SENSE bus and the current bus are available to the user. The voltage SENSE connects to Bank A and the current source connects to Bank B. Specify only the Bank A channels in the ch_list, since respective channels in Bank B will close automatically. Used for 4-wire ohms measurements only.

ch_list Address of channel list. See Glossary. All channels in the *ch_list* must be connected by ribbon cable to the HP 44702A/B High-Speed Voltmeter. Channel number ranges follow. Note that for measurements over the ribbon cable: NRDGS x (# of channels - 1) must be < 4095.

NOTE

The 20-channel and 60-channel multiplexers cannot be used with the dedicated interface (ribbon) cable.

Accessory Type	Range	Description
24-Channel Multiplexers	ES00 - ES11	A Bank
	ES12 - ES23	B Bank
48-Channel Multiplexers	ES00 - ES47	Channels

RANGE *range_list* List of voltmeter ranges that correspond to the channels to be scanned. Entries in *range_list* apply to corresponding entries in *ch_list*. One range can be used for the entire *ch_list* or separate ranges can be used for the corresponding channels and/or channel sequences.

The value(s) you specify for *range_list* should be the maximum signal amplitude expected on each channel (for voltage measurements) or the maximum resistance expected on each channel (for resistance measurements). The voltmeter then selects the correct range.

To select autorange for a channel and/or channel sequence, specify AUTO or 0 for *range_list*. In autorange mode, the voltmeter samples the input signal and selects the appropriate range before each measurement. Default setting = AUTO.

USE *ch* Voltmeter slot number. See Glossary.

Remarks

RANGE vs. CLWRITE Commands

If RANGE *range_list* is not specified in CLWRITE, autorange is used for all channels specified by CLWRITE.

If the RANGE command is executed after CLWRITE, the range set by the RANGE command is used for all channels specified by CLWRITE and the range values which were specified are cancelled.

Channel List vs. Range List

Each item in the *range_list* is associated with a corresponding channel or channel sequence (*ch_list*). If the number of items in the range list is less than the number of channels or channel sequences, the default range (autorange) is used for the remaining channels. If the number of items in the range list is greater than the number of channels or channel sequences, ERROR 21 - TOO MANY ARGUMENTS is generated. For example, OUTPUT 709;"CLWRITE 500-519,RANGE 2,10" generates ERROR 21 - TOO MANY ARGUMENTS since there are two range list entries (2 and 10) specified, but only one channel list sequence (500-519) specified.

Channel List Greater Than 10 Items

If a channel list greater than 10 items is desired when using the CLWRITE command, use an Integer array to store the channel numbers and then specify the array name for the *ch_list* parameter. When the array is declared, a valid channel address must be in each element of the array. There can be no empty or "extra" array elements.

Using multiple CLWRITE commands to specify a continous channel list will result in the previous channel list being overwritten in the voltmeter's scan RAM. Thus, the channel list in the RAM at the time of the measurement would be the channels specified by the last CLWRITE command.

Data Returned

None

Related Commands

RANGE, SCANMODE, USE

Examples

Example: Setting the Scan List and Voltmeter Range with CLWRITE

This program takes 100 DC voltage measurements and stores them in a PACKED array in the mainframe. The CLWRITE command specifies the channel list, and voltmeter range to be used for each channel. Other commands are included to show the context in which the CLWRITE command is often used.

```
10 OUTPUT 709;"RST"                          !Reset HP 3852A and HP 44702A/B
20 OUTPUT 709;"USE 500"                      !Use voltmeter in slot 5
30 OUTPUT 709;"PACKED PACKRDGS(199)"         !Declare array of 200 bytes
40                                           !(100 readings: 2 bytes/reading)
50 OUTPUT 709;"SCANMODE ON"                  !Set scanner mode
60 OUTPUT 709;"CONF DCV"                     !Configure for DC voltage
70 OUTPUT 709;"CLWRITE 400-409, RANGE 8"     !Set scan list and range
80 OUTPUT 709;"PRESCAN 10"                   !Make 10 passes through scan
90                                           !list before stop trigger is
100                                          !acknowledged
110 OUTPUT 709;"ASCAN ON"                    !Turn autoscan on (only one
120                                          !trigger is required
130                                          !to make 10 passes through
140                                          !the scan list)
150 OUTPUT 709;"ARMODE BEFORE"               !Autorange before measure
160                                          !trigger is received
170 OUTPUT 709;"SCTRIG EXTO"                 !Trigger the vm externally
180 OUTPUT 709;"XRDGS 500, INTO PACKRDGS"    !Transfer readings to mainframe
190                                          !when scan sequence completes
200 END
```

• HP 44715A 5-Channel Counter/Totalizer

Description

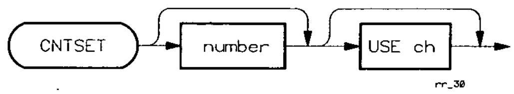
Presets counter channel to begin counting from a specified number of counts or to rollover after a specified number of counts.

Prerequisites

Specified channel must be set to the TOTAL function.

Syntax

CNTSET [*number*] [USE *ch*]



Parameters

number Number of counts at which channel starts counting OR number of counts required to cause counter rollover. Count range = 0 to 4294967296 and *number* range = -2147483648 to 2147483647. The power-on and default *number* = 0.

```
Preset Counter:

 number = counts                  (counts <2147483648)
 number = counts - 4294967296     (counts >=2147483648)

Rollover After Counts:

 number = -counts                 (counts <2147483648)
 number = 4294967296 - counts     (counts >=2147483648)
```

USE *ch* Specifies channel to count. Channel range = ES00 through ES04.

Remarks

CNTSET Reset and Triggering

CNTSET applies only to the TOTAL function and should be programmed after the FUNC TOTAL command. Unless the channel is set for internal triggering (with a TRIG AUTO command), a trigger is required after the CNTSET command to start totalizing. The CNTSET value is set only once so that if the totalize is aborted and then restarted, the count will begin at 0.

Data Returned

None

Related Commands

FUNC, TOTAL, TRIG, CHREAD, CHREADZ, XRDGS

Examples

Example: Preset/Rollover Channel (Counts <2147483648)

For this example, let counts = 1000. In line 10, *number* = counts = 1000 presets the counter to start counting at 1000 counts. In line 20, *number* = -1000 causes the counter to rollover after 1000 counts.

```
10 OUTPUT 709;"CNTSET 1000"      !Start count at 1000 counts
20 OUTPUT 709;"CNTSET -1000"     !Rollover after 1000 counts
```

Example: Preset/Rollover Channel (Counts >= 2147483648)

For this example, let counts = 3 000 000 000 (which is > 2147483648). Then, in line 30 *number* = counts - 4294967296 = -1294967296 presets the counter to start counting at 3 000 000 000 counts. In line 40, *number* = 4294967296 - counts = 1294967296 causes the counter to rollover after 3 000 000 000 counts.

```
30 OUTPUT 709;"CNTSET -1294967296"  !Start count at 3000000000 counts
40 OUTPUT 709;"CNTSET 1294967296"   !Rollover after 3000000000 counts
```

CNTSET

- **HP 44721A 16-Channel Digital Input**
- **HP 44722A 8-Channel Digital Input**

Description

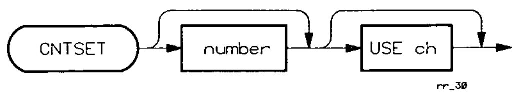
Presets digital input channel to begin counting from a specified number of counts or to rollover after a specified number of counts.

Prerequisites

None

Syntax

CNTSET [*number*] [USE *ch*]



Parameters

number Number of counts at which channel starts counting OR number of counts required to cause counter rollover. Count range = 0 to 4294967296 and *number* range = -2147483648 to 2147483647. The power-on and default *number* = 0.

```
Preset Counter:

  number = counts                 (counts <2147483648)
  number = counts - 4294967296    (counts >=2147483648)

Rollover After Counts:

  number = -counts                (counts <2147483648)
  number = 4294967296 - counts    (counts >=2147483648)
```

USE *ch* Specifies channel to count. Channel range = ES00 - ES15 (HP 44721A) or ES00 - ES07 (HP 44722A).

Remarks

Data Returned

None

Related Commands

EDGE

Examples

Example: Preset/Rollover Channel (Counts <2147483648)

For this example, let counts = 1000. In line 10, *number* = counts = 1000 presets the channel to start counting at 1000 counts. In line 20, *number* = -1000 causes the channel counter to rollover after 1000 counts.

```
10 OUTPUT 709;"CNTSET 1000"      !Start count at 1000 counts
20 OUTPUT 709;"CNTSET -1000"     !Rollover after 1000 counts
```

Example: Preset/Rollover Channel (Counts >= 2147483648)

For this example, let counts = 3 000 000 000 (which is > 2147483648). Then, in line 30 *number* = counts - 4294967296 = -1294967296 presets the channel to start counting at 3 000 000 000 counts. In line 40, *number* = 4294967296 - counts = 1294967296 causes the channel counter to rollover after 3 000 000 000 counts.

```
30 OUTPUT 709;"CNTSET -1294967296"  !Start count at 3000000000 counts
40 OUTPUT 709;"CNTSET 1294967296"   !Rollover after 3000000000 counts
```

COMPEN

- **Mainframe**

Description

Post process temperature and strain conversion. COMPEN enables you to measure the electrical parameters (i.e. resistance, voltage) of a thermistor, thermocouple, or strain gage, and later convert those parameters to corresponding temperatures or strain.

Prerequisites

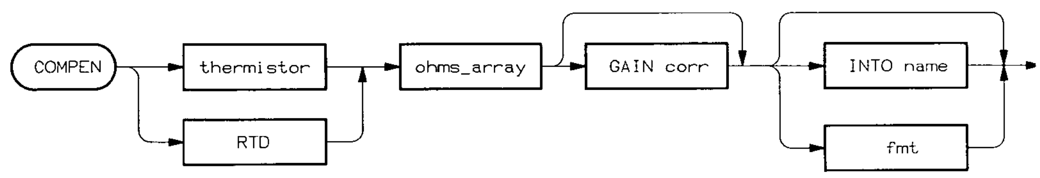
Requires firmware revision 3.5 or greater.

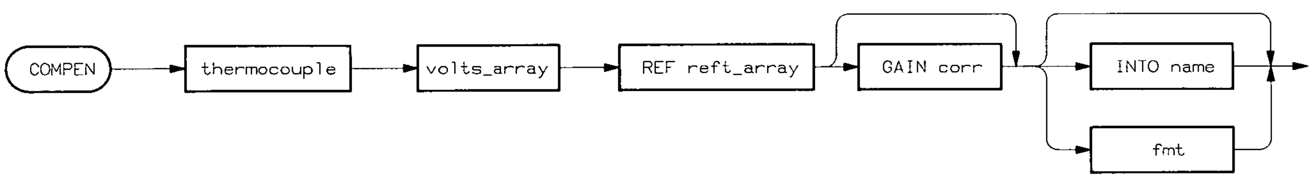
Syntax

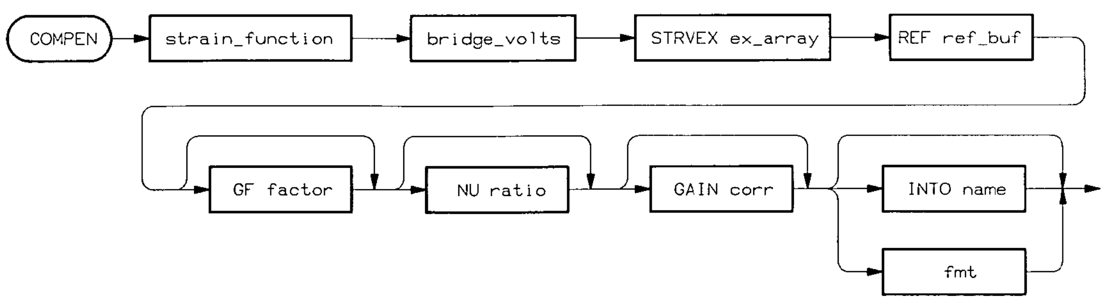
COMPEN *thermistor* or *RTD* *ohms_array* [**GAIN** *corr*] [**INTO** *name*] or [*fmt*]

COMPEN *thermocouple* *volts_array* **REF** *reft_array* [**GAIN** *corr*] [**INTO** *name*] or [*fmt*]

COMPEN *strain_function* *bridge_volts* **STRVEX** *ex_array* **REF** *ref_buf* [**GF** *factor*] [**NU** *ratio*] [**GAIN** *corr*] [**INTO** *name*] or [*fmt*]







Parameters

thermistor or *RTD* — Thermistor or RTD whose resistance is converted to temperature. Thermistors and RTDs supported by this command include:

```
THM2252
THM5K
THM10K
RTD85
RTD92
```

ohms_array — Real, Integer, or Packed array containing the resistance measurements of the thermistor or RTD used.

GAIN *corr* — Real or Integer array or a number containing values by which the readings in *ohms_array* are divided. If a number is specified, the value is divided into each of the readings in *ohms_array*. If an array is specified, there is a 1-for-1 correspondence between the correction array and *ohms_array*. If necessary, the correction array will wraparound and the correction factors will be used again.

thermocouple — Thermocouple whose voltage is converted to temperature. Thermocouples supported by this command include:

```
TEMPB       (B type thermocouple)
TEMPE       (E type thermocouple)
TEMPJ       (J type thermocouple)
TEMPK       (K type thermocouple)
TEMPN14     (N14 type thermocouple)
TEMPN28     (N28 type thermocouple)
TEMPR       (R type thermocouple)
TEMPS       (S type thermocouple)
TEMPT       (T type thermocouple)
```

volts_array — Real, Integer, or Packed array containing the voltage measurements of the thermocouple used.

REF *reft_array* — Real or Integer array or a number containing the measurement(s) of the isothermal block (REFT). If a number is specified, the single reference is used for all voltage measurements in the volts array. If an array is specified, there is a 1-for-1 correspondence between the reference array and the volts array. If necessary, the reference array will wraparound and the references used again.

GAIN *corr* — Real or Integer array or a number containing values by which the readings in *volts_array* are divided. If a number is specified, the value is divided into each of the readings in *volts_array*. If an array is specified, there is a 1-for-1 correspondence between the correction array and *volts_array*. If necessary, the correction array will wraparound and the correction factors will be used again.

strain_function — Bridge configuration whose output voltage is converted to strain. The strain functions which represent these configurations and which are supported by this command are:

STRQ	(1/4 bridge strain)
STRHB	(bending 1/2 bridge strain)
STRFB	(bending full bridge strain)
STRHP	(1/2 bridge Poisson strain)
STRFBP	(bending full bridge Poisson strain)
STRFP	(full bridge Poisson strain)

bridge_volts Real, Integer, or Packed array containing the bridge output voltage measurements.

STRVEX *ex_array* Real or Integer array or a number containing the measurement(s) of the bridge excitation voltage. If a number is specified, the single excitation voltage is used for all measurements in the bridge volts array. If an array is specified, there is a 1-for-1 correspondence between the excitation voltages and the bridge volts array. If necessary, the excitation voltage array will wraparound and the voltages will be used again.

REF *ref_buf* Real or Integer array or a number containing the unstrained reference measurement(s). If a number is specified, the single reference is used for all measurements in the bridge volts array. If an array is specified, there is a 1-for-1 correspondence between the references and the bridge volts array. The reference array will also wraparound if necessary.

GF *factor* Real or Integer array or a number containing a gage factor. A single gage factor is used for all readings in the bridge volts array, or there is a 1-for-1 correspondence depending on whether a number or an array is specified. The gage factor array will wraparound if necessary. If no gage factor is specified, a default gage factor of 2.0 is used. Specifying a gage factor with an exponent of -6 will return the converted readings in microstrain.

NU *ratio* Real or Integer array or a number containing a Poisson ratio. A single Poisson ratio is used for all readings in the bridge volts array, or there is a 1-for-1 correspondence depending on whether a number or an array is specified. The Poisson ratio array will wraparound if necessary. Note that when STRHP, STRFBP, or STRFP is specified, NU *ratio* is a required parameter of the COMPEN command.

GAIN *corr* Real or Integer array or a number containing values by which the readings in *bridge_volts* are divided. If a number is specified, the value is divided into each of the readings in *bridge_volts*. If an array is specified, there is a 1-for-1 correspondence between the correction array and *bridge_volts*. If necessary, the correction array will wraparound and the correction factors will be used again.

INTO *name* Array in which the converted readings are stored.

fmt See Glossary. The default format for all converted readings is RASC.

Remarks

COMPEN Allows for Faster Temperature and Strain Measurements

When measurements involve a thermistor or an RTD, the readings converted by COMPEN are resistance (OHM or OHMF) measurements. When measurements involve a thermocouple or a strain gage, the readings converted by COMPEN are voltage (DCV) measurements. Note that these measurements occur much faster since the conversion to temperature or strain does not occur at the time of the measurement.

Data Returned

Returned to mainframe memory or to the output buffer are the converted temperature or strain measurements.

Related Commands

None

Examples

Example 1: Using COMPEN to Convert Thermistor Measurements

The following program makes 30 resistance measurements (3 channels - 10 measurements/channel) with thermistors and uses the COMPEN command to convert those resistances to corresponding temperatures. Once the readings have been converted, they are entered into the controller and displayed. This program assumes the HP 44701A voltmeter is used.

```
10   !Reset the HP 3852A and turn off the display to increase
20   !command execution speed.
30   !
40   OUTPUT 709;"RST"
50   OUTPUT 709;"DISP OFF"
60   !
70   !Download the measurement channel list into a mainframe array.
80   !Ten measurements are to be taken on each channel (200, 201,
90   !202) in the list.
100  !
110  OUTPUT 709;"SUB LOADLIST"
120  OUTPUT 709;"  INTEGER C,CH_LIST(19)"
130  OUTPUT 709;"  FOR C=0 TO 9"
140  OUTPUT 709;"    VWRITE CH_LIST 200,-202"
150  OUTPUT 709;"  NEXT C"
160  OUTPUT 709;"SUBEND"
170  OUTPUT 709;"CALL LOADLIST"
180  !
190  !Perform the measurements using the desired thermistor or RTD.
200  !Connect the thermistor or RTD in a 4-wire configuration and
210  !specify a 4-wire resistance measurement (OHMF). Store the
220  !readings in a packed array for increased measurement speed.
230  !
240  OUTPUT 709;"PACKED OHM_RDGS(119)"
250  OUTPUT 709;"CONFMEAS OHMF CH_LIST USE 700 INTO OHM_RDGS(0)"
260  !
270  !Convert the resistance measurements to temperatures once all
280  !measurements have been made.
290  !
300  OUTPUT 709;"REAL COMPRDGS(29)"
310  OUTPUT 709;"COMPEN THM10K OHM_RDGS INTO COMPRDGS(0)"
320  !
330  !Read the converted readings into the output buffer. Enter and
340  !display those readings on the controller.
350  !
360  OUTPUT 709;"VREAD COMPRDGS"
370  REAL Rdgs(0:9,0:2)
```

```
380   ENTER 709;Rdgs(*)
390   PRINT USING "10(3(MDD.5D,3X),/)";Rdgs(*)
400   END
```

A typical output based on this program is shown below:

22.69238	24.35352	27.56836
22.69043	24.34668	27.48535
22.68945	24.34082	27.56445
22.68945	24.33496	27.63086
22.68848	24.3291	27.70898
22.6875	24.32324	27.78516
22.6875	24.31738	27.85352
22.6875	24.31152	27.92285
22.68652	24.30469	27.99219
22.68652	24.29688	28.06348

Example 2: Using COMPEN to Convert Thermocouple Measurements

By replacing lines 190 through 310 in the program of Example 1 with lines 190 through 312 shown below, 30 voltage measurements can be made using a T type thermocouple and the COMPEN command used to convert the voltages to corresponding temperatures. A typical output based on lines 190 through 312 is also shown below.

```
190   !Perform the measurements using the desired thermocouple. Measure
200   !the isothermal block for each channel. Specify a DC voltage
210   !measurement and store the readings in a packed array for increased
220   !measurement speed.
230   !
240   OUTPUT 709;"PACKED VOLT_RGS(119)"
250   OUTPUT 709;"REAL T(2)"
260   OUTPUT 709;"CONFMEAS REFT 200-202 USE 700 INTO T(0)"
270   OUTPUT 709;"CONFMEAS DCV CH_LIST USE 700 INTO VOLT_RGS(0)"
280   !
290   !Convert the voltage measurements to temperatures once all
300   !measurements have been made.
310   !
311   OUTPUT 709;"REAL COMPRDGS(29)"
312   OUTPUT 709;"COMPEN TEMPT VOLT_RGS REF T INTO COMPRDGS(0)"
```

Converted temperatures based on these program lines are shown below:

28.91992	28.81152	28.04492
28.81445	28.78418	28.01758
28.78906	28.77637	28.00879
28.81055	28.80273	28.03809
28.80664	28.78516	28.02246
28.7998	28.79297	28.02832
28.7998	28.81055	28.02441
28.81055	28.7959	28.03516
28.79883	28.79687	28.02539
28.7959	28.79883	28.00684

Example 3: Using COMPEN to Convert Strain Measurements

By replacing lines 190 through 310 in the program of Example 1 with lines 190 through 319 shown below, 30 voltage measurements can be made with a strain gage in a 1/4 bridge configuration and the COMPEN command used to convert the voltages to corresponding strains. A typical output based on lines 190 through 319 is also given below.

```
190  !Perform the measurements using the desired strain gage bridge
200  !configuration. Measure the unstrained reference voltage and
210  !the bridge excitation voltage. Apply stress to the specimen,
220  !then specify a DC voltage measurement (DCV) and measure the
230  !bridge output voltage. Store the readings in a packed array.
240  !
250  OUTPUT 709;"PACKED VOLT_RGS(119)"
260  OUTPUT 709;"REAL BRG_EXT, STR_REF"
270  OUTPUT 709;"CONF STRUN USE 700"
280  OUTPUT 709;"MEAS STRUN 200 USE 700 INTO STR_REF"
290  OUTPUT 709;"MEAS STRVEX 200 USE 700 INTO BRG_EXT"
300  OUTPUT 709;"RANGE 0.03"
310  OUTPUT 709;"NPLC 0"
311  OUTPUT 709;"AZERO OFF"
312  OUTPUT 709;"WAIT 2"
313  OUTPUT 709;"MEAS DCV CH_LIST USE 700 INTO VOLT_RGS(0)"
314  !
315  !Convert the voltage measurements to strain once all measurements
316  !have been made.
317  !
318  OUTPUT 709;"REAL COMPRDGS(29)"
319  OUTPUT 709;"COMPEN STRQ VOLT_RGS STRVEX BRG_EXT REF STR_REF GF 2E-6
INTO COMPRDGS(0)"
```

Converted strains based on these program lines are shown below:

347.4035	90.10919	261.624
347.4035	90.10919	261.624
347.4035	90.10919	218.7398
347.4035	90.10919	218.7398
347.4035	90.10919	218.7398
347.4035	132.9824	218.7398
347.4035	90.10919	218.7398
304.5119	47.23967	304.5119
304.5119	90.10919	261.624
347.4035	90.10919	218.7398

CONF

- **HP 44701A Integrating Voltmeter**
- **HP 44702A/B High-Speed Voltmeter (Scanner or System Mode)**

Description

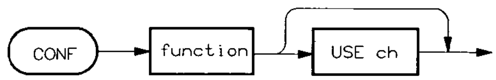
CONF configures an HP 44701A or HP 44702A/B voltmeter for a specified measurement function and selects preset values for voltmeter operation.

Prerequisites

CONF requires mainframe firmware revision 2.0 or greater for use with the strain functions (STRxx). The system must not be scanning with an HP 44702A/B voltmeter when CONF is executed.

Syntax

CONF *function* [USE *ch*]



cr_31

Parameters

function HP 44701A or HP 44702A/B voltmeter measurement function. Unless noted, *function* applies to both voltmeters.

function	Measurement	
ACV	AC voltage	(HP 44701A only)
DCV	DC voltage	
OHM	2-wire ohms	
OHM10K	2-wire ohms up to 10 kΩ	(HP 44702A/B only)
OHM100K	2-wire ohms up to 100 kΩ	(HP 44702A/B only)
OHM1M	2-wire ohms up to 1 MΩ	(HP 44702A/B only)
OHMF	4-wire ohms	
OHMF10K	4-wire ohms up to 10 kΩ	(HP 44702A/B only)
OHMF100K	4-wire ohms up to 100 kΩ	(HP 44702A/B only)
OHM1M	4-wire ohms up to 1 MΩ	(HP 44702A/B only)
TEMPtype	Thermocouple temperature (include ref temp) type = B, E, J, K, N14, N28, R, S, and T	
REFT	Reference temperature (isothermal block)	
THMtype	Thermistor (2-wire ohms) type = 2252 (2252 Ω thermistor) type = 5K (5 kΩ thermistor) type = 10K (10 kΩ thermistor)	
THMFtype	Thermistor (4-wire ohms) type = same as THMtype	

RTDtype	RTD (2-wire ohms) type = 85 (RTDs with α = 0.00385 Ω/Ω/°C) type = 92 (RTDs with α = 0.003916 Ω/Ω/°C)
RTDFtype	RTD (4-wire ohms) type = same as RTD type
STRVEX	Strain gage bridge excitation voltage
STRUN	Strain gage unstrained bridge output
STRQ	1/4 bridge strain
STRHB	Bending 1/2 bridge strain
STRFB	Bending full bridge strain
STRQTEN	Tension shunt verification
STRQCOMP	Compression shunt verification
STRHP	1/2 bridge Poisson strain
STRFBP	Full bridge bending Poisson strain
STRFP	Full bridge Poisson strain

USE *ch* Slot where the voltmeter is installed.

Remarks

Voltmeter Functions Set With CONF

Executing CONF is equivalent to executing the following commands in the sequence shown. The voltmeter configurations can then be altered by these low-level commands.

HP 44701A Integrating Voltmeter

Command	Description	Preset Value	Notes
STRIG	Scan trigger source	SCAN	
SADV	Scan advance source	SCAN	
TRIG	Trigger mode	HOLD	
DELAY	Measurement delay	set by CONF	[1]
AZERO	Autozero	ON	
FUNC	Measurement function	Set by CONF	[2]
RANGE	Voltmeter range	AUTO	[3]
TERM	Input terminals	BOTH	
NRDGS	Number of readings	1	
NPLC	Power line cycles	1	
OCOMP	Offset compensation	OFF	
DISABLE INTR	Disable interrupt	disabled	

[1] = Built-in default delays based on function/range/NPLC.

[2] = Function set depends on function parameter of CONF.

[3] = For mainframe firmware revisions 2.2 and greater, CONF TEMPtype sets RANGE AUTO. For earlier firmware revisions, the 30 mV range is selected for B, N28, R, S, and T type thermocouples, and the 300 mV range is selected for E, J, K, and N14 thermocouples.

CAUTION

For an HP 44701A voltmeter, CONF sets TERM BOTH. Thus, the mainframe backplane (i.e., multiplexers) and the voltmeter external (rear panel) terminals BOTH serve as inputs to the voltmeter. Thus, with CONF, the voltage on a closed multiplexer channel also appears on the external terminals of the voltmeter and any voltage on the external terminals appears on the multiplexer channel.

To prevent damage to the voltmeter accessory and to the system which may be connected, ensure that nothing is connected to the HP 44701A voltmeter external terminals when making measurements over the backplane. If measurements are performed using the external terminals only, it is recommended that TERM EXT be set.

HP 44702A/B High-Speed Voltmeter (System Mode)

Command	Description	Preset Value	Notes
STRIG	Scan trigger source	SCAN	
SADV	Scan advance source	SCAN	
TRIG	Trigger mode	HOLD	
FUNC	Measurement function	Set by CONF	[4]
RANGE	Voltmeter range	AUTO	[5]
TERM	Input terminals	INT	
NRDGS	Number of readings	1	
RDGS	Reading destination	SYS	
RDGSMODE	Interrupt condition	DAV	
ARMODE	Autorange mode	AFTER	
DELAY	Trig delay/sample rate	1 ms/10 µs	[6]
AZERO	Autozero mode	ONCE	

[4] = Function set depends on function parameter of CONF command.

[5] = For mainframe firmware revisions 2.2 and greater, CONF TEMPtype sets RANGE AUTO. For earlier firmware revisions, the 40 mV range is selected for B, N28, R, S, and T type thermocouples, and the 320 mV range is selected for E, J, K, and N14 thermocouples.

[6] = OHM1M and OHMF1M functions set a 6 msec delay.

HP 44702A/B High-Speed Voltmeter (Scanner Mode)

Command	Description	Preset Value	Notes
SADV	Scan advance source	SCAN	
STRIG	Scan trigger source	SCAN	
SCTRIG	Scan trigger (Scanner Mode)	HOLD	
TRIG	Measure trigger mode	INT	
STTRIG	Stop trigger source	INT	
FUNC	Measurement function	Set by CONF	[7]
RANGE	Voltmeter range	AUTO	[8]
TERM	Input terminals	RIBBON	
NRDGS	Number of readings	1	
RDGS	Reading destination	SYS	
RDGSMODE	Interrupt condition	DAV	
ARMODE	Autorange mode	AFTER	
ASCAN	Autoscan mode	OFF	
SCDELAY	Scan trigger delay	1 ms	[9]
SPER	Sample period	1 ms	[9]
PRESCAN	Scans before stop trig	1	
POSTSCAN	Scans after stop trig	0	
AZERO	Autozero mode	ONCE	

[7] = Function set depends on function parameter of CONF command.

[8] = For mainframe firmware revisions 2.2 and greater, CONF TEMPtype sets RANGE AUTO. For earlier firmware revisions, the 40 mV range is selected for B, N28, R, S, and T type thermocouples, and the 320 mV range is selected for E, J, K, and N14 thermocouples.

[9] = OHM1M and OHMF1M set a 6 msec scan trigger delay and a 6 msec sample period.

CONF Disables and Clears Interrupts

Executing CONF disables and clears interrupts on the measurement channel.

Data Returned

None

Related Commands

CONFMEAS, MEAS, MONMEAS

Examples

Example: DC Voltage Configuration (HP 44701A)

This program uses CONF to configure an HP 44701A voltmeter in slot 1 for a DC voltage measurement on channel 402 of an HP 44705A in slot 4. NRDGS is used to change the number of readings per trigger from 1 (as set with CONF) to 10.

```
10  DIM Volts(0:9)                  !Define controller array
20  OUTPUT 709;"USE 100"            !Use voltmeter in slot 1 of mainframe
30  OUTPUT 709;"CONF DCV"           !Configure for DC voltage
40  OUTPUT 709;"NRDGS 10"           !Set 10 readings/trigger
50  OUTPUT 709;"MEAS DCV,402"       !Measure DC volts on ch 402
60  ENTER 709;Volts(*)              !Enter readings
70  PRINT USING "K,/";Volts(*)      !Display readings
80  END
```

A typical return for a 9 V, 5% source is:

```
9.01487
9.01486
 .          10 readings
 .
9.01485
```

Example: DC Voltage Configuration (HP 44702A/B)

This program segment uses CONF to configure an HP 44702A/B in slots 5 and 6 for DC voltage measurements. SCTRIG EXT0 sets the scan trigger source to external trigger (EXT0 port). Since A is defined as a Packed mainframe array (2 bytes/reading), array size is 20 bytes (not elements) to store the 10 readings.

```
10  OUTPUT 709;"PACKED A(19)"              !Define mainframe array
20  OUTPUT 709;"USE 500"                   !Use HP 44702A/B in slot 5
30  OUTPUT 709;"SCANMODE ON"               !Set Scanner mode
40  OUTPUT 709;"CONF DCV"                  !Configure for DC voltage
50  OUTPUT 709;"CLWRITE SENSE 400-409"     !Set ch 400-409 as ch list
60  OUTPUT 709;"SCTRIG EXT0"               !Set external scan trigger
70  OUTPUT 709;"XRDGS 500 INTO A"          !Transfer readings to mainframe
            .
            .
```

Example: 1/4 Bridge Strain Measurements

This program makes ten strain gage measurements at 10 µsec intervals on channel 500 of an HP 44732A or HP 44733A using an HP 44702A/B voltmeter in slots 6 and 7. Channel 500 is set for amplification with gain = 100, no filtering, and is configured for 1/4 bridge strain measurements. The gage factor (GF) assumed = 2.

The HP 44702A/B voltmeter is set for Scanner mode, 1 prescan, no postscans, 10 readings per channel, and internal scan, measure, and stop triggers. Since measurement is via the ribbon cable, the ribbon cable must be connected between the HP 44732A or HP 44733A and the HP 44702A/B. Also, the EXC SENSE and

EXC terminals must be connected (H to +E and L to -E2) and the bridge completion jumper must be set to the 1/2,1/4 position.

When the program executes, the bridge offset is first nulled with the NULL command. Next, the excitation voltage is measured with CONFMEAS STRVEX and stored in mainframe variable Exc. The HP 44702A/B is configured for strain measurements and the program is then paused to allow time for the specimen to be strained. Press the Continue key (or equivalent) to continue the program after the specimen is strained.

Since CONF sets SCDELAY .001, when the program is continued, a 10 msec delay occurs. Since SPER 10E-6 is set, 10 measurements are then made at 10 µsec intervals. The results are stored in mainframe array Strn. Since the channel is configured for 1/4 bridge measurements, the strain is computed by the COMPEN command for 1/4 bridge strain measurements (COMPEN STRQ). Results are displayed in microstrain.

In the COMPEN command, note that GAIN 100 is used which divides all measured voltages by 100. Since the user inputs have been multiplied by 100 with the FUNC AMPLIFY,100 command, it is also necessary to divide the measured voltages by 100 to arrive at correct results.

```
10   !
20   !Define arrays and variables
30   !
40   DIM A(0:9)                                   !Define controller array
50   OUTPUT 709;"REAL Strn(9)"                    !Define HP 3852A REAL array
60   OUTPUT 709;"REAL Exc"                        !Define HP 3852A REAL var
70   !
80   !Null bridge offset on channel 500
90   !
100  OUTPUT 709;"USE 500"                         !44732A/33A use ch is 500
110  OUTPUT 709;"FUNC AMPLIFY,100"                !Set ampl w/gain of 100
120  OUTPUT 709;"FILTER OFF"                      !Ch 500 no filtering
130  OUTPUT 709;"TRIG HOLD"                       !No trigger on 44732A/33A
140  OUTPUT 709;"NULL"                            !Null ch 500 offset voltage
150  !
160  !Measure ch 500 exc voltage
170  !
180  OUTPUT 709;"USE 600"                         !Use 44702A/B vm in slot 6
190  OUTPUT 709;"SCANMODE ON"                     !Set Scanner mode
200  OUTPUT 709;"CONFMEAS STRVEX,500,INTO Exc"    !Meas and store exc voltage
210  !
220  !Set HP 44702A/B for strain measurements
230  !
240  OUTPUT 709;"CONF DCV"                        !Conf for DC volts
250  OUTPUT 709;"NRDGS 10"                        !Make 10 readings
260  OUTPUT 709;"SPER 10E-6"                      !10 usec between meas trigs
270  OUTPUT 709;"CLWRITE 500"                     !Set ch 500 as scan list
280  !
290  !Pause program (apply strain to specimen)
300  !
310  PAUSE                                        !Pause - strain specimen
320  !
```

```
330  !Make 10 strain measurements
340  !
350  OUTPUT 709;"SCTRIG INT"                       !Internal scan trig
360  OUTPUT 709;"XRDGS 600 INTO Strn"              !Trans rdgs to mf array
370  !
380  !Compute and display microstrain
390  !
400  OUTPUT 709;"COMPEN STRQ Strn STRVEX Exc,REF 0,GF 2.E-6,GAIN 100" !Comp
410  ENTER 709;A                                   !Enter microstrains
420  PRINT "Reading    Microstrain"                !Display header
430  PRINT                                         !Space
440  FOR I=0 TO 9                                  !Start computation loop
450    PRINT I+1,A(I)                              !Display microstrain
460  NEXT I                                        !Increment count
470  END
```

A typical return is:

```
Reading        Microstrain

   1           298.5461
   2           325.9725
   3           291.0972
   4           324.2795
   5           333.7605
   6           545.0994
   7           521.961
   8           383.2004
   9           329.02
   10          299.5618
```

- **HP 44715A 5-Channel Counter/Totalizer**

Description

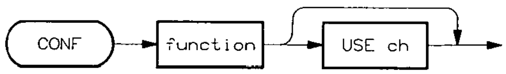
Configure counter channel counting/measurement function.

Prerequisites

Double-input functions (all except Ungated Total Counts and Ungated Total Counts, Modulo NPER) can only be programmed on channels configured for double inputs.

Syntax

CONF *function* [USE *ch*]



rr_31

Parameters

function Set counter channel function. Sets channel to one of ten *functions*, as shown in the following table, where Function = description of channel function, function = parameter of CONF command, D = double-input function, and S = single-input function. Refer to NPER command for details on Modulo NPER functions.

Function	function	Ch	Description	Application
Ungated Total Counts	TOTAL	S	Totalizes A input.	Total counts on single input.
Gated Total Counts	TOTAL	D	Totalizes A input, gated by B input.	Total counts on single input, gated by second input.
Ungated Total Counts, Mod NPER	TOTALM	S	Totalizes A input, count modulo NPER.	Total counts on single input, modulo NPER.
Gated Total Counts, Mod NPER	TOTALM	D	Totalizes A input, gated by B input, count modulo NPER.	Total counts on single input, gated by second input, count modulo NPER.
Up/Down Counts	UDC	D	Count up on A input, count down on B input. Result is (A-B) counts.	Count difference between counts of two inputs.

Function	function	Ch	Description	Application
Up/Down Counts, Mod NPER	UDCM	D	Count up on A input, count down on B input. Result is (A-B) counts, modulo NPER.	Count difference between counts of two inputs, modulo NPER.
Counts/ Direction	CD	D	Count A input up or down. B input controls direction.	Count relative number of up counts and down counts.
Counts/ Direction, Mod NPER	CDM	D	Count A input up or down. B input controls direction. Count modulo NPER.	Count relative number of up counts and down counts, modulo NPER.
Ratio	RAT	D	Count A input until NPER periods to B input occur.	Measure ratio of A input to B input counts.
Period	PER	D	Measure average of NPER periods of A input.	Measure average value of NPER periods of input
Delayed Period	PERD	D	Measure NPERth period of A input, as gated by B input.	Measure value of single period of input, delayed by NPER periods.
Frequency	FREQ	S	Measure frequency of A input.	Measure frequency of single input.

USE *ch* Address of channel to be used. Channel number (depends on HP 44715A hardware configuration) = ES00 through ES04.

Remarks

CONF Disables and Clears Interrupts

Executing the CONF command disables and clears all interrupts enabled on the measurement channel.

Counter Preset Values

Executing CONF is equivalent to executing the following commands in the sequence shown. The counter configuration can be subsequently altered by these low level commands.

All Functions Except FREQ

Command	Description	Preset Value	Note
TRIG	Counter trigger mode	HOLD	
FUNC	Counter function	Set by CONF	[a]
TERM	Counter input terminals	ISO	
EDGE	Counted/gated edge	HL,HL	
NPER	Measurement period/reset	10	
CNTSET	Start count/rollover	0	[b]
DISABLE INTR	Disable interrupt	Disabled	
SPER	Sample period	1 us	

[a] = TBASE = AUTO is also set. TBASE is specified for the PER and PERD functions only.

[b] = Command applies to the TOTAL function only.

FREQ Function

Command	Description	Preset Value
TRIG	Counter trigger mode	HOLD
TBASE	Time base	AUTO
TERM	Counter input terminals	ISO
EDGE	Counted/gated edge	HL
DISABLE INTR	Disable interrupt	Disabled
SPER	Sample period	1 us

Counter Functions Vs. Frequency Configuration

When the accessory configuration jumper is set to the FREQ position, all five channels simultaneously measure frequency and no other functions (e.g. TOTAL, PER, etc.,) can be set.

Data Returned

None

Related Commands

FUNC

Examples

Example: Set Counter Channel to TOTAL Function

This program configures the counter accessory to totalize inputs to the channel. The counter is then triggered and once the count is taken, the reading is transferred to the controller and displayed. For this example, the Card Configuration jumper is set to the TOTAL position.

```
10 OUTPUT 709;"RST"                !Reset the HP 3852A and HP 44715A
20 OUTPUT 709;"USE 103"            !Use channel 3 in slot 1
30 OUTPUT 709;"CONF TOTAL"         !Configure counter to totalize
40 OUTPUT 709;"TRIG SGL"           !Issue a single trigger
50 Wait 5                          !Totalize for 5 seconds
60 OUTPUT 709;"CHREAD 103"         !Return count to output buffer
70 ENTER 709;A                     !Enter reading into controller
80 PRINT A                         !Display reading
90 END
```

- **HP 44721A 16-Channel Digital Input**
- **HP 44722A 8-Channel Digital Input**

Description

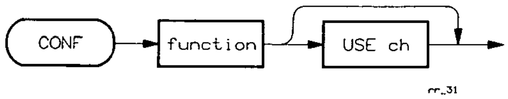
Configure digital input function. The digital inputs can be configured to measure/sense input levels or totalize.

Prerequisites

None

Syntax

CONF *function* [USE *ch*]



Parameters

function Digital input function.

function	Description
LVL	Configure channel to measure input level. LOW (0) or HIGH (1) for DC inputs. OFF (0) or ON (1) for AC inputs.
TOTAL	Totalize inputs on the specified channel.

USE *ch* Address of digital input channel used. For function LVL, channel range = ES16-ES31 (HP 44721A); range = ES08-ES15 (HP 44722A). For function TOTAL, range = ES00-ES15 (HP 44721A); range = ES00-ES07 (HP 44722A).

Remarks

CONF Disables and Clears Interrupts

Executing CONF disables and clears all interrupts enabled on the measurement channel.

Digital Input Preset Values

Executing CONF is equivalent to executing the following commands in the sequence shown. The digital input configuration can be subsequently altered by these low level commands.

Command	Description	Preset Value	Notes
DISABLE INTR	Disable interrupt	Disabled	
EDGE	Count/interrupt edge	LH	
CNTSET	Begin count/rollover	0	[a]

[a] = Set for the TOTAL function only.

Data Returned

None

Related Commands

None

Examples

Example: Set Channel to Read Input State

The following program CONFigures the HP 44721A digital input accessory to sense the input level on channel 3. After a 1 second pause, the input level on the channel is read and displayed on the front panel. For a high level, 1.000000 is displayed. For a low level, 0.000000 is displayed.

```
10 OUTPUT 709;"RST"           !Reset HP 3852A and digital input acc.
20 OUTPUT 709;"USE 519"       !Use ch 3 in slot 5 (lvl function)
30 OUTPUT 709;"CONF LVL"      !Configure ch to sense input level
40 WAIT 1                     !Wait 1s before reading channel state
50 OUTPUT 709;"CHREAD 519"    !Read state of channel
60 END
```

- HP 44701A Integrating Voltmeter
- HP 44702A/B High-Speed Voltmeter (Scanner or System Mode)

Description

See the **CONFMEAS** Command

CONFMEAS

- **HP 44701A Integrating Voltmeter**
- **HP 44702A/B High-Speed Voltmeter (Scanner or System Mode)**

Description

CONFMEAS configures an HP 44701A or HP 44702A/B voltmeter for a specified measurement function and then initiates a scan and measurement of specified channels. CONFMEAS is equivalent to sending CONF immediately followed by MEAS.

Prerequisites

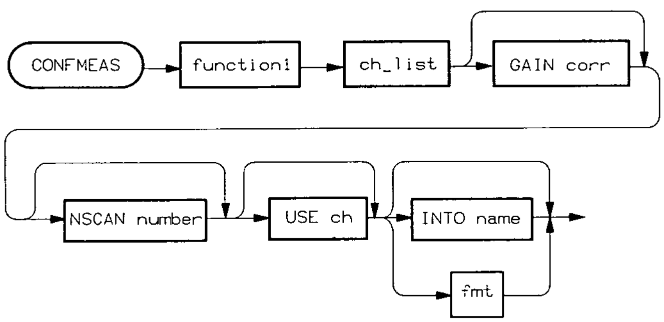
CONFMEAS requires mainframe firmware revision 2.0 or greater for use with an HP 44717A, HP 44718A, HP 44719A, or HP 44720A and requires mainframe firmware revision 3.5 or greater for use with an HP 44730A, HP 44732A, or HP 44733A. The NSCAN parameter is available with firmware revision 2.2 or greater and the GAIN parameter is available with firmware revision 3.5 or greater. The system must not be scanning with an HP 44702A/B voltmeter when CONFMEAS is executed.

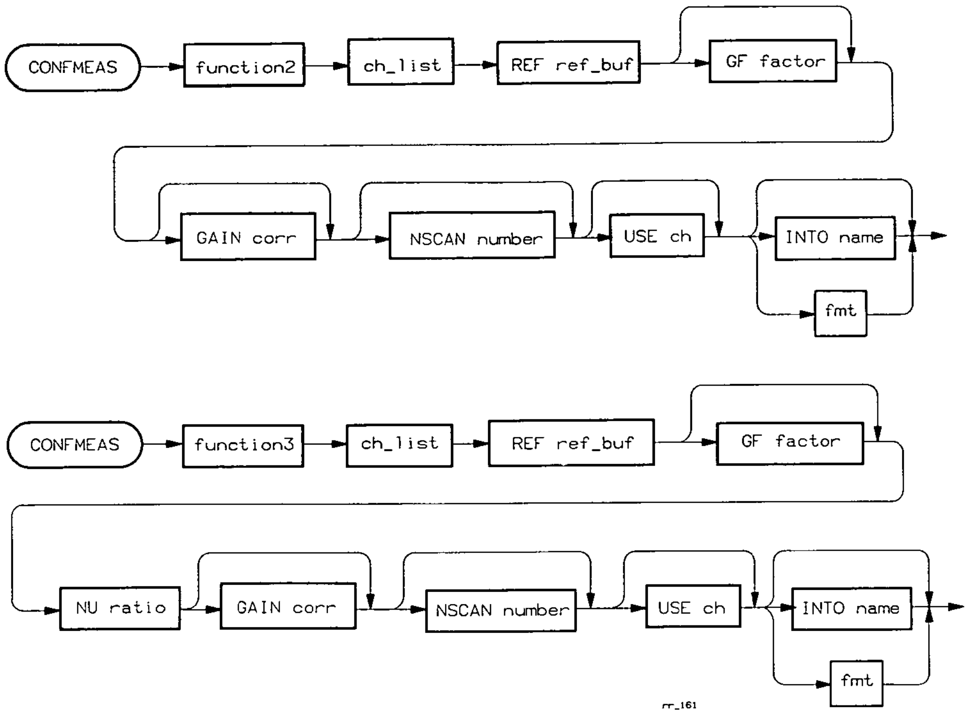
Syntax

CONFMEAS *function1 ch_list* [GAIN *corr*] [NSCAN *number*] [USE *ch*] [INTO *name*] or [*fmt*]

CONFMEAS *function2 ch_list* REF *ref_buf* [GF *factor*] [GAIN *corr*] [NSCAN *number*] [USE *ch*] [INTO *name*] or [*fmt*]

CONFMEAS *function3 ch_list* REF *ref_buf* [GF *factor*] NU *ratio* [GAIN *corr*] [NSCAN *number*] [USE *ch*] [INTO *name*] or [*fmt*]





Parameters

function HP 44701A or HP 44702A/B voltmeter measurement function. The *function1*, *function2*, or *function3* parameter specified causes the voltmeter to be configured for and make the measurement shown. Unless noted, *function* applies to both voltmeters. *function1* is used for general measurements (voltage, resistance, thermocouple, thermistor, RTD, and strain references), *function2* is used for non-Poisson arrangement strain gage measurements, and *function3* is used for Poisson arrangement strain measurements.

function1	Measurement	Note
ACV	AC voltage	[1]
DCV	DC voltage	
OHM	2-wire ohms	
OHM10K	2-wire ohms up to 10 kΩ	[2]
OHM100K	2-wire ohms up to 100 kΩ	[2]
OHM1M	2-wire ohms up to 1 MΩ	[2]
OHMF	4-wire ohms	
OHMF10K	4-wire ohms up to 10 kΩ	[2]
OHMF100K	4-wire ohms up to 100 kΩ	[2]
OHM1M	4-wire ohms up to 1 MΩ	[2]
TEMPtype	Thermocouple temperature (include ref temp) type = B, E, J, K, N14, N28, R, S, and T	

REFT	Reference temperature (isothermal block)	
THMtype	Thermistor (2-wire ohms) type = 2252 (2252 Ω thermistor) type = 5K (5 kΩ thermistor) type = 10K (10 kΩ thermistor)	
THMFtype	Thermistor (4-wire ohms) type = same as THMtype	
RTDtype	RTD (2-wire ohms) type = 85 (RTDs with α = 0.00385 Ω/Ω/°C) type = 92 (RTDs with α = 0.003916 Ω/Ω/°C)	
RTDFtype	RTD (4-wire ohms) type = same as RTD type	
STRVEX	Strain gage bridge excitation voltage	[3]
STRUN	Strain gage unstrained bridge output	[4]

function2	Measurement	Note
STRQ	1/4 bridge strain	[4]
STRHB	Bending 1/2 bridge strain	[4]
STRFB	Bending full bridge strain	[4]
STRQTEN	Tension shunt verification	[5]
STRQCOMP	Compression shunt verification	[5]

function3	Measurement	Note
STRHP	1/2 bridge Poisson strain	[4]
STRFBP	Full bridge bending Poisson strain	[4]
STRFP	Full bridge Poisson strain	[4]

[1] = HP 44701A voltmeter only.
[2] = HP 44702A/B voltmeter only.
[3] = HP 44717A/44718A/44719A/44720A/44730A/44732A/44733A only.
[4] = HP 44717A/44718A/44719A/44720A/44732A/44733A only.
[5] = HP 44717A and HP 44718A only.

ch_list Address of channel list. See Glossary.

REF *ref_buf* For CONFMEAS when *function2* or *function3* is used, array or number containing the unstrained bridge output voltage (reference voltage) measurements for the corresponding channel list (*ch_list*). When a one-element array, a variable, or a number is declared or specified for *ref_buf*, the single reference is used for all channels in *ch_list*. The contents of the array or variable are typically generated by CONFMEAS STRUN ... INTO *ref_buf*.

GF *factor* For CONFMEAS when *function2* or *function3* is used, array or number containing or representing a gage factor. When several gage factors are stored in an array, the gage factor in array element 0 is used for the strain measurement on the first channel in the list, the gage factor in array element 1 is used for the second channel in the list, etc.

If the number of channels on which strain is measured is greater than the maximum index of the array, the array "wraps around" and the factors are used again as necessary. When a one-element array, variable, or number is declared or specified for the gage factor, the single gage factor is used for all channels in the channel list.

Default gage factor = 2.0. When a gage factor is specified with an exponent of -6 (e.g., GF 2.E-6), the result is returned in microstrain.

NU *ratio* When *function3* is used (for Poisson configuration only), array or number containing or representing a Poisson ratio. When several ratios are stored in an array, the Poisson ratio in array element 0 is used for the strain measurement on the first channel in the list, the ratio in array element 1 is used with the second channel in the list, etc.

If the number of channels on which strain is measured is greater than the maximum index of the array, the array "wraps around" and the ratios are used again as necessary. When a one-element array, variable, or number is declared or specified for the Possion ratio, the single ratio is used for all channels in the channel list.

GAIN *corr* Real or Integer array or a number containing value(s) by which the measured readings are divided. The corrected values are stored in the mainframe or are sent to the output buffer and/or to the display in RASC format. Default GAIN *corr* = no correction and thus no format conversion.

If the number of channels on which strain is measured is greater than the maximum index of the array, the array "wraps around" and the factors are used again as necessary. When a one-element array, variable, or number is declared or specified for the gain correction, the single correction factor is used for all channels in the channel list.

The channel gain can be from an external source or from the amplification function of an HP 44730A, HP 44732A, or HP 44733A (see the GAIN command).

NSCAN *number* Number of scans to be made through the *ch_list*. Default NSCAN = 1. NSCAN is only available with mainframe firmware revision 2.2 or greater. (NSCAN *number*)*(number of channels in *ch_list*)* (NRDGS *number*) must result in 67,108,863 readings or less. For an HP 44702A/B in scanner mode and RDGS GPIO set, NSCAN *number* range = 1 to 2147483647.

When measuring thermocouples, the reference is measured prior to each pass through the channel list. When measuring strain (using *function2* or *function3*), the bridge excitation voltage (STRVEX) is measured prior to each pass through the channel list.

USE *ch* Slot where the voltmeter is installed.

INTO *name* Mainframe array or variable in which data is stored. For strain gage reference measurements (STRUN), the array should be Real. When references are stored into this array, each bridge completion channel reference is stored sequentially into the array starting at the current index. (See **REF** *ref_buf*.)

fmt Data format in which the measured function is displayed or returned to the output buffer. The default format is RASC. For STRVEX, the PACK format is the voltmeter packed format. For all other strain functions (STRxx), the PACK format is RL64. If the GAIN *corr* parameter is specified, *fmt* is always RL64.

Remarks

Voltmeter Functions set by CONFMEAS

CONFMEAS sets an HP 44701A or HP 44702A/B voltmeter to a predefined state (refer to CONF for settings). Note that the preset conditions cannot be changed once CONFMEAS is executed, since the configuration phase (CONF) is immediately followed by the measurement (MEAS) phase.

CONFMEAS Disables and Clears Interrupts

Executing CONFMEAS disables and clears all interrupts enabled on the measurement channel(s).

4-Wire Ohms Measurements

For 4-wire ohms measurements with CONFMEAS, *ch_list* identifies the "sense" input channels and the "current source" channels are automatically selected. Inclusive channel specifiers will "skip" over "current source" channels.

Using CONFMEAS with the HP 44701A

When using CONFMEAS with the HP 44701A, the CONFiguration phase sets TERM BOTH so the mainframe backplane (i.e., multiplexer) and voltmeter external (rear panel) terminals both serve as inputs to the voltmeter. When multiplexer channels are closed, the voltage on the channel also appears on the external terminals and any voltage on the external terminals appears on the closed multiplexer channel.

CAUTION

To prevent damage to the voltmeter accessory and to the system connected to the multiplexer channel(s) during backplane scans (using CONFMEAS), ensure that no voltage (or system) is connected to the HP 44701A voltmeter external terminals.

Potential Deadlock

With INBUF OFF, the controller and the HP 3852A can deadlock if multiple commands (or command terminators) are sent in a single command line and the command generates enough data to fill the output buffer (CONFMEAS can fill the output buffer). The best way to avoid a deadlock is to send a single command per command line and to read the results as soon as possible after a data-generating command is sent.

Static vs. Dynamic Strain Measurements

With static strain measurements, the specimen is strained to a final position and the difference in bridge output ratios between the unstrained and strained states is measured. Generally, static strain measurements are made with an HP 44701A Integrating Voltmeter (or equivalent) for greater measurement accuracy. When CONFMEAS is used, maximum reading rate for an HP 44701A is about 25 readings per second.

With dynamic strain measurements, the specimen undergoes a changing strain as the measurement is made. Generally, the HP 44702A/B voltmeter is used for high-speed dynamic strain measurements, since 100 kHz reading rates are available. However, when CONFMEAS is used, maximum measurement rate for strain measurements with an HP 44702A/B voltmeter is about 500 readings/second so CONFMEAS cannot be used for high-speed dynamic strain measurements. See CONF for for an example of high-speed dynamic strain measurements.

Declaring REF *ref_buf*, GF *factor*, and NU *ratio* Arrays

If arrays are declared for the reference measurement, gage factor, and Poisson ratio, the maximum index of the array should be equal to the number of channels (*ch_list*) on which strain measurements will be taken. The array should also be filled with references, factors, or ratios for the channels in the corresponding channel list.

When an array is declared, each element of the array contains the number 0 until data is written to, or stored in that particular element. For example, if strain is measured on 10 channels and the reference array (*ref_buf*) with a maximum index of 9 (starting index is 0) contains only five reference readings, the reference used to determine the strain measured on the last five channels will be 0, resulting in invalid measurements on those channels.

Data Returned

Data returned by an HP 44701A or HP 44702A/B using CONFMEAS follows. Note that OHM10K, OHMF10K, OHM100K, OHMF100K, OHM1M, and OHMF1M apply only when an HP 44702A/B voltmeter is used. Also, ACV applies only when an HP 44701A voltmeter is used.

function	Data Returned
Voltage	
ACV	AC voltage (volts)
DCV	DC voltage (volts)
Resistance	
OHM	Resistance (ohms)
OHM10K	Resistance (≤ 10 kΩ)
OHM100K	Resistance (≤ 100 kΩ)

OHM1M	Resistance ($\leq$ 1 MΩ)
OHMF	Resistance (ohms)
OHMF10K	Resistance ($\leq$ 10 kΩ)
OHMF100K	Resistance ($\leq$ 100 kΩ)
OHMF1M	Resistance ($\leq$ 1 MΩ)
Temperature	
TEMPtype	Thermocouple temp (deg C)
REFT	Ref temp of isothermal block (deg C)
THMtype	Thermistor temp (deg C)
THMFtype	Thermistor temp (deg C)
RTDtype	RTD temp (deg C)
RTDFtype	RTD temp (deg C)
Strain	
STRVEX	Bridge excitation voltage
STRUN	Unstrained bridge output
STRQ	1/4 bridge strain
STRHB	Bending 1/2 bridge strain
STRFB	Bending full bridge strain
STRQTEN	Tension shunt verification
STRQCOMP	Compression shunt verification
STRHP	1/2 bridge Poisson strain
STRFBP	Full bridge bending Poisson strain
STRFP	Full bridge Poisson strain

Related Commands

CONF, MEAS

Examples

Example: DC Voltage Measurements

This program uses CONFMEAS to configure an HP 44701A voltmeter in mainframe slot 1 to its preset values and make eight DC voltage measurements on channels 400 through 407 of an HP 44705A multiplexer.

```
10 DIM Volt(0:7)                       !Dim controller array
20 OUTPUT 709;"USE 100"                !Use voltmeter in mainframe slot 1
30 OUTPUT 709;"CONFMEAS DCV,400-407"   !Conf/meas DC volts on ch 400-407
40 ENTER 709;Volt(*)                   !Enter readings
50 PRINT USING "K,/";Volt(*)           !Display readings
60 END
```

A typical return for a set of 9 V, 5% inputs (values in volts) is:

```
8.9454
8.9333
8.9995
9.0112
```

```
8.9223
8.9349
8.9741
8.9945
```

Example: Multiple Scans Through Channel List

This program uses CONFMEAS to make 100 DC voltage measurements on channels 0 through 9 of an HP 44705A using an HP 44701A or HP 44702A/B in mainframe slot 7. NSCAN sets 10 scans through the channel list, so 100 readings are made. For this program, the input is assumed to have been externally amplified by 10, so GAIN 10 divides each measurement by 10 before storing the value in mainframe array A.

```
10 DIM B(0:99)                          !Dim controller array
20 OUTPUT 709;"USE 700"                 !Use voltmeter in slot 7
30 OUTPUT 709;"REAL A(99)"              !Declare 100-element array
40 OUTPUT 709;"CONFMEAS DCV,0-9,GAIN 10,NSCAN 10,INTO A"  !Conf/meas
50 OUTPUT 709;"VREAD A"                 !Trans rdgs to output buffer
60 ENTER 709;B(*)                       !Enter rdgs
70 PRINT USING "K,/";B(*)               !Display rdgs
60 END
```

A typical return for 5V, 5% inputs to each channel (actual voltage of 0.5V multiplied by external gain of 10) follows.

```
.4995
.50025
 .          100 readings
 .
.499
```

Example: 1/4 Bridge Strain Measurement

This program uses CONFMEAS to measure 1/4 bridge strain on channels 400 through 402 of an HP 44717A, HP 44718A, HP 44719A, or HP 44720A using an HP 44701A voltmeter in slot 1 of the mainframe. Results are returned in microstrain.

The PAUSE statement (line 80) halts the program after the reference voltages have been measured to allow time for the specimens to be strained. Press the Continue (or equivalent) key to continue the program after the specimens are strained.

```
10   DIM A(0:2)                                      !Define cont array
20   !
30   !Make ref measurements and pause program
40   !
50   OUTPUT 709;"REAL Str_ref(2)"                    !Define reference array
60   OUTPUT 709;"USE 100"                            !Use HP 44701A in slot 1
70   OUTPUT 709;"CONFMEAS STRUN,400-402,INTO Str_ref"   !Meas ref voltage
80   PAUSE                                           !Pause - strain specimens
90   !
100  !Measure strain on channels
```

```
110  !
120  OUTPUT 709;"CONFMEAS STRQ,400-402,REF Str_ref" !Meas strains
130  !
140  !Enter and display strains (in microstrain)
150  !
160  ENTER 709;A(*)                                  !Enter strain values
170  PRINT "Channel   Microstrain"                   !Display header
180  PRINT                                           !Space
190  FOR I=0 TO 2                                    !Start display loop
200    PRINT 400+I,A(I)*1.E+6                        !Display microstrain
210  NEXT I                                          !Increment loop
220  END
```

A typical return (expressed in microstrain) is:

```
Channel     Microstrain

 400        455.2426
 401        1077.261
 402        1696.887
```

- **Mainframe**

Description

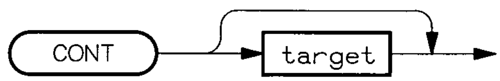
Continues a paused (PAUSE command) or stepped (STEP command) HP 3852A subroutine.

Prerequisites

The *target* parameter is used to continue run task subroutines in the multitasking mode. This parameter and multitasking are only available with mainframe firmware revision 3.0 or greater.

Syntax

CONT [*target*]



R.CON

Parameters

target Number of the run task containing the subroutine to be continued. The range for *target* is 0 to 7.

If *target* is not specified, the subroutine that was paused without the *target* parameter is continued. A "stepped" subroutine is also continued if *target* is not specified.

Remarks

Executing the CONT Command

When the *target* parameter is specified, the CONT command can be executed from within a subroutine as well as from the front panel or over the HP-IB. When *target* is not specified, CONT cannot be executed within a subroutine.

Data Returned

None. However, data may be returned from commands stored in the subroutine.

Related Commands

CALL, STEP, PAUSE, SUB, SUBEND, RUN

Examples

Example: Continuing a Paused Subroutine

```
10 OUTPUT 709;"RST"                !Reset the HP 3852A
20 OUTPUT 709;"REAL FLAG"          !Declare REAL variable FLAG
30 OUTPUT 709;"SUB TOGLE"          !Create subroutine TOGLE
40 OUTPUT 709;"VWRITE FLAG,0"      !Set FLAG = 0
50 OUTPUT 709;"PAUSE"              !Pause subroutine
```

```
60 OUTPUT 709;"VWRITE FLAG,1"       !Set FLAG = 1
70 OUTPUT 709;"SUBEND"              !End subroutine
            .
            .
210 OUTPUT 709;"CALL TOGLE"         !Subroutine pauses with FLAG = 0
220 OUTPUT 709;"VREAD FLAG"         !Read FLAG value
230 ENTER 709;Flag                  !Returns 0
240 PRINT Flag                      !Display 0
250 OUTPUT 709;"CONT"               !Subroutine continues to end
260 OUTPUT 709;"VREAD FLAG"         !Read FLAG value
270 ENTER 709;Flag                  !Returns 1
280 PRINT Flag                      !Display 1
290 END
```

Example: Continuing a Stepped Subroutine

```
10 OUTPUT 709;"RST"                 !Reset the HP 3852A
20 OUTPUT 709;"SUB BEEPER"          !Create subroutine BEEPER
30 OUTPUT 709;"BEEP"                !BEEP once
40 OUTPUT 709;"SUBEND"              !End subroutine
            .
            .
200 OUTPUT 709;"STEP BEEPER"        !Step to start of sub (before BEEP)
210 OUTPUT 709;"CONT"               !BEEP and finish sub
```

Example: Continuing a Subroutine Targeted to Pause

This example shows how the CONT command is used to continue a run task subroutine that was paused by the PAUSE *target* command.

In the program, the HP 3852A is placed in the multitasking mode and two subroutines are downloaded and directed to run tasks. Subroutine A (run task 1) counts from 0 to 199 and subroutine B (run task 2) repeatedly displays A B C.

As the program executes, subroutine B directs the PAUSE command (line 280) to run task 1 which pauses subroutine A. As subroutine B executes to completion, the CONT command in line 340 is directed to run task 1 which enables subroutine A to continue.

```
10    !Set a 1 second time-slice and set the number of run tasks and
20    !queue size the system will allow. Place the HP 3852A in the
30    !multitasking mode. The reset caused by ENABLE MULTI will load
40    !the TSLICE and NTASKS setting into the operating system.
50    !
60    OUTPUT 709;"TSLICE 1"
70    OUTPUT 709;"NTASKS 2,2"
80    OUTPUT 709;"ENABLE MULTI"
90    !
100   !Write and download the subroutines to be directed to run tasks.
110   !Subroutine A when active, counts from 0 to 199.
120   !
130   OUTPUT 709;"SUB A"
140   OUTPUT 709;"  INTEGER I"
```

```
150   OUTPUT 709;"  FOR I=0 TO 199"
160   OUTPUT 709;"    DISP I"
170   OUTPUT 709;"  NEXT I"
180   OUTPUT 709;"SUBEND"
190   !
200   !Download subroutine B. When active, the subroutine directs the
210   !PAUSE command to run task 1, thus pausing subroutine A.
220   !Subroutine B then displays the sequence A B C 300 times. When
230   !finished, the subroutine directs the CONT command to run task
240   !1 which enables subroutine A to continue.
250   !
260   OUTPUT 709;"SUB B"
270   OUTPUT 709;"  INTEGER J"
280   OUTPUT 709;"  PAUSE 1"
290   OUTPUT 709;"  FOR J=0 TO 299"
300   OUTPUT 709;"    DISP 'A'"
310   OUTPUT 709;"    DISP 'B'"
320   OUTPUT 709;"    DISP 'C'"
330   OUTPUT 709;"  NEXT J"
340   OUTPUT 709;"  CONT 1"
350   OUTPUT 709;"SUBEND"
360   !
370   !Direct the subroutines to run tasks. Set each run task to execute
380   !one time.
390   !
400   OUTPUT 709;"RUN 1 A"
410   OUTPUT 709;"RUN 2 B"
420   END
```

When this program runs, subroutine A executes first and reaches a count of about 70 before the system swaps to run task 2. As subroutine B executes, it directs the PAUSE command to run task 1 which pauses subroutine A. When the system swaps back to subroutine A, it sees that it is paused. At this point, the system swaps back to run task 2 and subroutine B executes to completion. Prior to completing, however, subroutine B directs the CONT command to run task 1 allowing subroutine A to continue.

CONV

- **Mainframe**

Description

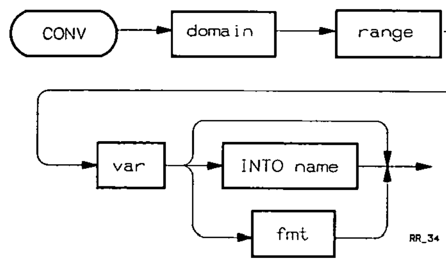
Linear interpolation of measurement data using user-defined look-up tables.

Prerequisites

The domain, range, var, and destination (if used) arrays must be separate arrays. Also, the domain and range arrays must be Real arrays with the same maximum index.

Syntax

CONV *domain range var* [**INTO** *name*] or [*fmt*]



Parameters

domain Real array of x-coordinates on the graph of the measurements to be interpolated. This array must have the same maximum index as the *range* array and its values must be entered in increasing order.

range Real array of y-coordinates on the graph of the measurements to be interpolated. These values correspond to the x values with the same array indices. The *range* array must have the same maximum index as the *domain* array.

var Array containing measurements which will be interpolated according to the x to y mapping defined by the *domain* and *range* arrays. The *var* array can be a Real, Integer, or Packed array.

INTO *name* See Glossary.

fmt See Glossary. Default *fmt* is RASC.

Remarks

Domain/Range vs. Array Indexing

Values are entered into the *domain* and *range* arrays (with the VWRITE command) as input-output pairs matched by array index. For example, in the following table, the value of 2 in element 0 of the *domain* array corresponds with the value of 4 in element 0 of the *range* array. The value of 3 in element 1 of the *domain* array corresponds with the value of 9 in the range array, and so on.

```
array element (index):     0   1   2   3   4   5  ...
----------------------------------------------

domain (x) value:          2   3   4   5   6   7  ...

range (y) value:           4   9   16  25  36  49 ...
```

The measurements in the *var* array are linearly interpolated by solving for y in the equation:

```
(y - y1)    (x - x1)
-------- = --------
(y2 - y1)  (x2 - x1)
```

where x is a measurement in the *var* array falling on, or within, a set of matched pairs, x1 is the x value of the lower matched pair, y1 is the y value of the lower matched pair, x2 is the x value of the upper matched pair, and y2 is the y value of the upper matched pair.

As an example, in the previous table, the set of ordered pairs (x,y) is (2,4), (3,9), (4,16)... If a measurement in the *var* array is 3.5, it will be interpolated to 12.5 based on the equation which is solved for y, and the domain and range values specified.

```
          (x - x1)                          (3.5 - 3)
y = y1 + -------- (y2 - y1)   =>  y = 9 + --------- (16 - 9) =  12.5
          (x2 - x1)                         (4 - 3)
```

Rules for Ranges and Domains

The *domain* (x) values must be in order of increasing magnitude (e.g. x(0) < x(1) < x(2)). For best accuracy, the *range* (y) values should be spaced equal distances from each other.

The *domain* and *range* arrays must also have the same maximum index. If the array sizes differ, an error message is generated before the CONV command executes.

Potential Errors

If the *domain* (x) values are not in order of increasing magnitude, no error is generated and CONV will execute. However, the interpolated measurements as a result of CONV will probably be wrong.

If a value in the *var* array does not fall within the bounds of the *domain* array, an error is generated and the resulting value is 1.0E+38 (overload). Following the error, the interpolation will continue as normal until all readings are converted.

Potential Deadlock Problem

With INBUF OFF, the controller and the HP 3852A may deadlock if multiple commands (or terminators) are sent in a single command line and a command generates enough data to fill the output buffer (CONV can fill the output buffer). The best way to avoid a deadlock is to send a single command per command line and read the results as soon as possible after a data-generating command is sent.

Data Returned

Results of *domain* (x) to *range* (y) interpolation. After all measurements are converted, the index pointers of the *domain*, *range*, and *var* arrays are reset to zero.

Related Commands

REAL, DIM, INTEGER, SCALE, VWRITE

Examples

Example: Interpolating Voltage Measurements Using CONV

The following program scans a total of five 20 channel multiplexers in slots 2 through 6 (channels 200 - 619). The 100 measurements are stored in the array VAR. The measurements are then interpolated using the 8-pair lookup table stored in arrays DOM and RNG, transferred to the computer, and displayed.

```
10  DIM Conv(0:99)                                !dimension controller array
20  OUTPUT 709;"RST"                              !Reset the HP 3852A
30  OUTPUT 709;"REAL DOM(7),RNG(7),VAR(99)"       !dimension mainframe arrays
40  OUTPUT 709;"VWRITE DOM -3,-2,-1,0,1,2,3,4"
50                                                !Define domain (x-coord)
60  OUTPUT 709;"VWRITE RNG -8,-4,-2,0,2,4,6,8"
70                                                !Define range (y-coord)
80  OUTPUT 709;"CONFMEAS DCV,200-619,USE 700,INTO VAR"
90                                                !Measure channels
100  OUTPUT 709;"CONV DOM,RNG,VAR"                !Interpolate readings
110  ENTER 709;Conv(*)                            !Enter data into controller
120  PRINT USING"K,/";Conv(*)                     !Display data
130  END
```

A typical output if the first five readings are 2.5, 1.5, -.5, 4, and -2.5 volts is:

```
 5.00000
 3.00000
-1.00000
 8.00000
-6.00000
```

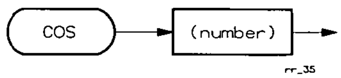
• **Trigonometric Operation**

Description

Numeric expression evaluated as a command parameter. Returns the cosine of its argument.

Syntax

COS (*number*)



Parameters

number Numeric expression in radians. Range = ±2.98156824429204E+8 radians.

Remarks

None

Examples

Example: Using COS Function

This program computes the cosine of 30 degrees (0.5236 radians) and displays the result (.8660248) on the controller CRT.

```
10 OUTPUT 709;"RST"                  !Reset the HP 3852A
20 OUTPUT 709;"VREAD COS (.5236)"    !Compute and read COS of 30 degrees
30 ENTER 709;A                       !Enter value
40 PRINT "Cosine = ";A               !Display value
50 END
```

A typical return for this program is:

```
Cosine = .8660248
```

CREATE ASCII

• **HP 44788A HP-IB Controller**

Description

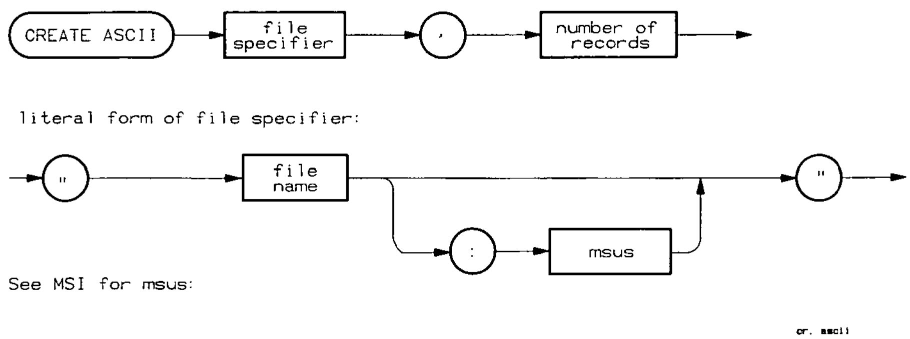
This statement creates an ASCII file on the mass storage media.

Prerequisites

Requires firmware revision 3.5 or greater.

Syntax

CREATE ASCII *file specifier, number of records*



Parameters

file specifier The name that the newly created file will have.

number of records Defines the length of the file.

Remarks

CREATE ASCII File and Directory Entries

CREATE ASCII creates a new ASCII file and directory entry on the mass storage media. CREATE ASCII does not open the file. Opening of files is done by the ASSIGN statement. The physical records of an ASCII file have a fixed length of 256 bytes, while logical records have variable lengths which are automatically determined when the OUTPUT statement is used. In the event of an error, no directory entry is made and the file is not created.

Data Returned

None.

Related Commands

None.

Examples

Example: Create ASCII file

The following entries both create ASCII file TEXT of 100 records. The first statement is entered from the HP 3852A front panel, while the second statement is entered from the system controller.

```
CREATE ASCII "TEXT",100                    !Create ASCII file TEXT
100 OUTPUT 709;"CREATE ASCII 'TEXT',100"   !Create ASCII file TEXT
```

• **HP 44788A HP-IB Controller**

Description

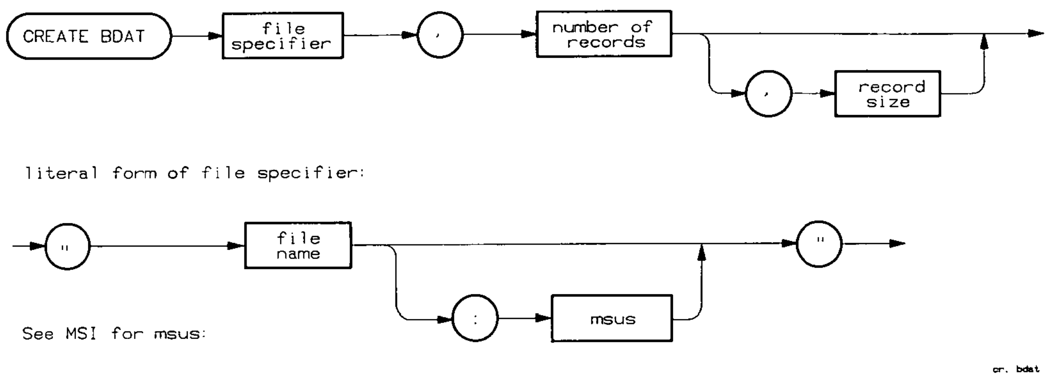
This statement creates a BDAT (binary untyped data file) on the mass storage media.

Prerequisites

Requires firmware revision 3.5 or greater.

Syntax

CREATE BDAT *file specifier, number of records* [*,record size*]



Parameters

file specifier The name that the newly created file will have.

number of records Defines the length of the file.

record size Defines the length of a record. Default size = 256 bytes/record.

Remarks

CREATE BDAT File and Directory Entries

CREATE BDAT creates a new BDAT file and directory entry on the mass storage media. CREATE BDAT does not open the file. Opening of files is done by the ASSIGN statement. In the event of an error, no directory entry is made and the file is not created. A sector is created at the beginning of the file for system use. This sector cannot be accessed by BASIC programs.

Data Returned

None.

Related Commands

None.

Examples

Example: Create BDAT File

The following statements each creates BDAT file MYDATA with 100 records of 128 bytes each. The first statement is entered from the HP 3852A front panel, while the second statement is entered from a system controller.

```
CREATE BDAT "MYDATA",100,128                    !Create BDAT file MYDATA
100 OUTPUT 709;"CREATE BDAT 'MYDATA',100,128"   !Create BDAT file MYDATA
```

• **Mainframe**

Description

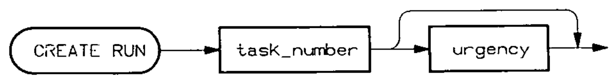
Sets the urgency (priority) of a run task environment before a subroutine is directed to that task by the RUN command. This enables the subroutine to start execution sooner once it has been directed to the task.

Prerequisites

Requires firmware revision 3.5 or greater and the HP 3852A must be in the multitasking mode.

Syntax

CREATE RUN *task_number* [*urgency*]



Parameters

task_number Run task whose urgency is set. The range for *task_number* is 0 to 7. Only one run task can be specified per CREATE RUN command.

urgency Urgency assigned to the specified run task. The range for *urgency* is 1 to 253, where 1 is the highest priority and 253 is the lowest priority. The default *urgency* is 85.

Remarks

CREATE RUN Does Not Direct Subroutines to Run Tasks

The purpose of CREATE RUN is to enable subroutines to start execution faster once they have been directed to a run task by the RUN command. CREATE RUN does not direct subroutines to run tasks nor does it initiate subroutine execution. CREATE RUN loads the specified (or default) urgency into the operating system so that it does not have to be done by the subsequent RUN command.

Queued Subroutines Execute at the Urgency Created for the Run Task

Queued subroutines awaiting execution in the run task will start execution faster when pulled from the queue and will execute at the priority set by CREATE RUN.

Executing CREATE RUN When the Task is Already Created

If CREATE RUN specifies a run task which is already created, only the urgency of the task changes. Thus, the command would have the same effect as the URGENCY command.

Data Returned

None

Related Commands

RUN, URGENCY

Examples

Example: Using CREATE RUN to Set Run Task Urgencies

The following programs demonstrate two methods for setting a run task to a specific urgency. In the first program, CREATE RUN is used and in the second program, the URGENCY command is used.

```
10    OUTPUT 709;"TSLICE 0.065"
20    OUTPUT 709;"NTASKS 1,1"
30    OUTPUT 709;"ENABLE MULTI"
40    OUTPUT 709;"CREATE RUN 0 50"
50    OUTPUT 709;"REAL T,T1"
60    OUTPUT 709;"SET TIME 0"
70    !
80    OUTPUT 709;"SUB A"
90    OUTPUT 709;"  TIME INTO T1"
100   OUTPUT 709;"  REAL VRDGS(9)"
110   OUTPUT 709;"  CONFMEAS DCV 600-609 USE 700 INTO VRDGS"
120   OUTPUT 709;"  VREAD T1-T"
130   OUTPUT 709;"SUBEND"
140   !
150   OUTPUT 709;"SUB B"
160   OUTPUT 709;"  TIME INTO T"
170   OUTPUT 709;"  RUN 0 A"
180   OUTPUT 709;"SUBEND"
190   OUTPUT 709;"CALL B"
200   END
```

In this program, CREATE RUN (line 40) sets the urgency of run task 0 to 50 as part of the system configuration phase. Subroutine A (lines 80-130) makes 10 DC voltage measurements when directed to run task 0 by the RUN command (line 170).

```
10    OUTPUT 709;"TSLICE 0.065"
20    OUTPUT 709;"NTASKS 1,1"
30    OUTPUT 709;"ENABLE MULTI"
40    OUTPUT 709;"REAL T,T1"
50    OUTPUT 709;"SET TIME 0"
60    !
70    OUTPUT 709;"SUB A"
80    OUTPUT 709;"  TIME INTO T1"
90    OUTPUT 709;"  REAL VRDGS(9)"
100   OUTPUT 709;"  CONFMEAS DCV 600-609 USE 700 INTO VRDGS"
110   OUTPUT 709;"  VREAD T1-T"
120   OUTPUT 709;"SUBEND"
130   !
140   OUTPUT 709;"SUB B"
150   OUTPUT 709;"  TIME INTO T"
160   OUTPUT 709;"  RUN 0 A"
170   OUTPUT 709;"  URGENCY 0 50"
180   OUTPUT 709;"SUBEND"
```

```
190    OUTPUT 709;"CALL B"
200    END
```

The second program is similar to the first, however; the urgency of run task 0 is set by the URGENCY command in line 170. Subroutine A again makes 10 DC voltage measurements when directed to run task 0 by the RUN command (line 160).

The TIME command executed in lines 160 and 90 of the first program and lines 150 and 80 of the second program measures the time from when subroutine A is directed to the run task to the start of its execution. Note that subroutine A will start earlier in the first program since the urgency is set by CREATE RUN in the configuration phase, and URGENCY does not have to follow the RUN command (Error Code 84) as shown in the second program.

Should the default priority (85) be used thus not requiring the URGENCY command, program execution still begins faster when CREATE RUN is used. Execution of the above programs will demonstrate the speed advantage of CREATE RUN.

DELAY

• HP 44701A Integrating Voltmeter

Description

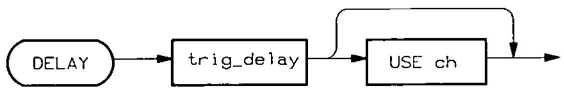
Sets the delay interval between measurement cycles. For one reading per trigger, the delay is inserted between the trigger event and the reading. For multiple readings per trigger, the delay is between the trigger event and the first measurement cycle and between the end of each measurement cycle and the start of the next cycle (between successive readings).

Prerequisites

None

Syntax

DELAY *trig_delay* [USE *ch*]



rr_36

Parameters

trig_delay Delay time in seconds between the trigger event and start of the measurement. *trig_delay* can be a number between 0 and 4294.967295 seconds. *trig_delay* = *AUTO* restores default delays used at power up or following a reset. *AUTO* is only available with mainframe firmware revisions 2.2 and greater.

USE *ch* Voltmeter slot number. See Glossary.

Remarks

Default Delay Times

The HP 44701A Integrating Voltmeter automatically sets a default delay time for each measurement function, range, and resolution.

The default delay time is updated automatically whenever the function or range changes. However, when a delay time value is specified, the value does not change until another delay value is specified; until the HP 44701A is reset (RST or RST *slot*); until power is cycled; or until the CONF command is executed.

For AC voltage measurements, default delay time is 1.5 seconds, regardless of range or resolution. Default delay times for DC voltage and 4-wire ohms measurements are listed in the following table.

HP 44701A Default Delay Times

Measurement Function	Range	Default delay in µseconds NPLC: 0.0005	0.005	0.1, 1, and 16
DCV	30 mV	380	496	600
	300 mV	200	260	320
	3V	200	260	320
	30V	36	46	57
	300V	11	14	17
OHMF	30 ohm	320	390	460
	300 ohm	200	240	290
	3k ohm	200	240	290
	30k ohm	200	240	290
	300k ohm	720	880	1000
	3M ohm	7200	8800	10000

Delays Less Than Default Values

You can specify delays less than the default values. However, the resulting settling time allowed may not produce accurate measurements. Also, because of hardware constraints, specified delays between 1 µsec and 58 µsec result in an actual delay of approximately 35 µsec.

Delays With OCOMP ON

With OCOMP ON, a delay occurs before the measurement with the current source on, and before the measurement with the current source off. (OCOMP is valid for 2-wire and 4-wire ohms measurements on the 30Ω through 30kΩ ranges. Offset compensation is not done on the 300kΩ and 3MΩ ranges.)

Data Returned

None

Related Commands

NRDGS, OCOMP, TRIG

Examples

Example: Set Trigger Delay (Multiple Readings)

This program sets a 1 second delay between the trigger event and the first measurement cycle and between successive measurement cycles (readings).

```
10 DIM Volt(0:9)               !Dimension computer array
20 OUTPUT 709;"RST"            !Reset the HP 3852A and HP 44701A
30 OUTPUT 709;"USE 100"        !Use voltmeter in mainframe slot 1
40 OUTPUT 709;"CONF DCV"       !Configure for DC voltage measurements
50 OUTPUT 709;"DELAY 1"        !Set 1 second delay
60 OUTPUT 709;"NRDGS 10"       !Set 10 readings per trigger
70 OUTPUT 709;"MEAS DCV, 1"    !Measure the voltage on channel 1 (slot 0)
80 ENTER 709;Volt(*)           !Enter readings
90 PRINT USING "K,/";Volt(*)   !Display readings
100 END
```

• **HP 44702A/B High-Speed Voltmeter (System Mode Only)**

Description

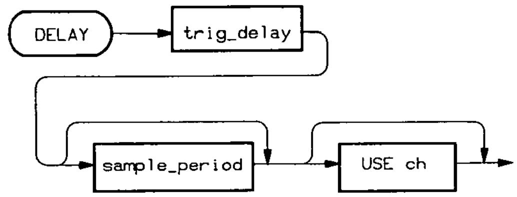
Sets the delay between the trigger and the start of the first measurement and the period of successive measurements on the same channel.

Prerequisites

None

Syntax

DELAY *trig_delay* [*sample_period*] [USE *ch*]



RR_37

Parameters

trig_delay — The delay time in seconds between the trigger and the start of the first measurement. *trig_delay* must be a number between 0 and 16.38375 msec. Power-on value for *trig_delay* = 0 µsec.

sample_period — Time (in seconds) of each successive measurement on the current measurement channel. *sample_period* must be a number between 0 and 1073.74182375 seconds. Power-on value for *sample_period* = 10 µsec. Default value = not changed from previous setting.

USE *ch* — Voltmeter slot number. See Glossary.

Remarks

Data Returned

None

Related Commands

NRDGS, TRIG, SPER

Examples

Example: Setting the Trigger Delay and Sample Period

The following program takes 30 voltage measurements and stores them in the mainframe. The program sets a 5 msec delay between the trigger and the first measurement on each channel, and a 1 second measurement period (*sample_period*) for the remaining nine measurements on each channel.

```
10 OUTPUT 709;"RST"                   !Reset the HP 3852A and HP 44702A/B
20 OUTPUT 709;"REAL RDGSSTOR(29)"     !Declare array in mainframe memory
30 OUTPUT 709;"USE 500"               !Use voltmeter in slot 5 of mainframe
40 OUTPUT 709;"SCANMODE OFF"          !Sets System Mode
50 OUTPUT 709;"CONF DCV"              !Configure for DC voltage measurements
60 OUTPUT 709;"NRDGS 10"              !Set 10 readings per trigger per channel
70 OUTPUT 709;"DELAY .005,1"          !Sets 5 ms trig delay, 1 sec sample period
80 OUTPUT 709;"MEAS DCV, 400-402, INTO RDGSSTOR"
90                                    !Measure and store DC voltage readings
100 END
```

• **HP 44714A 3-Channel Stepper Motor Controller/Pulse Output**

Description

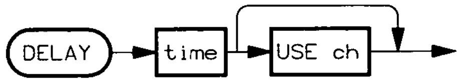
The DELAY command causes a specified time delay before a trigger command is executed for a MOVE or SUSTAIN command. A pulse train does not commence until the delay elapses and the trigger executes. All triggers can be delayed including an auto trigger.

Prerequisites

Requires mainframe firmware revision 3.0 or greater.

Syntax

DELAY *time* [USE *ch*]



Parameters

time — The *time* parameter sets the amount of delay between the trigger and the generation of pulses. The delay can be set from 0.000 to 65.536 seconds in one millisecond increments. At power-on, *time* = 0.000.

USE *ch* — Channel on which the trigger is to be delayed. Channel range is ES00 to ES02.

Remarks

Power-On State

The amount of delay at power-on is zero (*time* = 0.000).

Data Returned

None

Related Commands

TRIG, MOVE, SUSTAIN

Examples

Example: Send Pulses Continuously at 10 KHz; Delay the Trigger by 10 mSec

```
10 OUTPUT 709;"USE 201"                !Use channel 1 in slot 2
10 OUTPUT 709;"TRIG HOLD"              !Put trigger on hold
20 OUTPUT 709;"SUSTAIN 10000 NOWAIT"   !Velocity = 10 kHz
30 OUTPUT 709;"DELAY 0.010"            !Delay trigger 10 mSec
40 OUTPUT 709;"TRIG SGL"               !Trigger single (delayed)
50 END
```

DELSUB

- Mainframe

Description

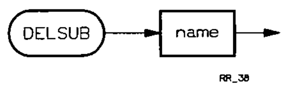
Recovers mainframe memory by deleting the contents of the HP 3852A subroutine stored in memory.

Prerequisites

None

Syntax

DELSUB *name*



Parameters

name Name of the subroutine whose contents will be deleted.

Remarks

Subroutine Name Not Deleted

Deleting a subroutine recovers the storage used for the commands in the subroutine. Therefore, the subroutine can't be called (with CALL). However, the deleted subroutine name will still appear in CAT (with size of zero) and can't be used as a variable or array name. Use SCRATCH to remove definition of all user-assigned names.

Commands Which Can't be Stored in Subroutines

SUB, DELSUB, SCRATCH, STEP, and CONT cannot be stored in subroutines.

Data Returned

None

Related Commands

CALL, STEP, CAT, SCRATCH, SUB

Examples

Example: Create and Delete Subroutine BEEPER

This program creates subroutine BEEPER and calls the subroutine, then deletes the subroutine and calls it again. The second call generates an error since BEEPER was deleted.

```
10 OUTPUT 709;"RST"                 !Reset the HP 3852A
20 OUTPUT 709;"SUB BEEPER"          !Start of sub BEEPER
30 OUTPUT 709;"BEEP"                !BEEP once
40 OUTPUT 709;"SUBEND"              !End of sub BEEPER
50 OUTPUT 709;"CALL BEEPER"         !Call BEEPER once
60 OUTPUT 709;"DELSUB BEEPER"       !Delete sub BEEPER
70 OUTPUT 709;"CALL BEEPER"         !Generates error - sub deleted
                 .
                 .
```

DELVAR

- **Mainframe**

Description

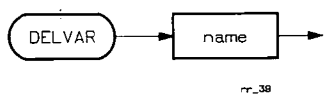
Recovers mainframe memory by releasing the memory allocated to the specified array.

Prerequisites

None

Syntax

DELVAR *name*



Parameters

name Name of the array whose contents will be deleted.

Remarks

Array Name Not Deleted

Deleting an array recovers the storage used by the array. However, the deleted array name will still appear in CAT (with size of zero) and cannot be used as a variable or subroutine name. Use SCRATCH to remove definition of all user-assigned names.

Array Type Cannot Be Redefined

An existing array or variable can be redeclared, but cannot be redefined to another type. For example, you could execute REAL J(2), use DELVAR to return the storage used by J, then execute REAL J(10). However, you cannot change the type of array J to other than REAL (i.e. INTEGER, PACKED).

Data Returned

None

Related Commands

CALL, STEP, CAT, SCRATCH, SUB

Examples

Example: Use of the DELVAR Command

This program declares a REAL array with a maximum index of 99. The DELVAR command is used to release the memory allocated to the array. The SIZE? command is included to show that following DELVAR, there is no memory associated with the specified array.

```
10 OUTPUT 709;"RST"             !Reset the HP 3852A
20 OUTPUT 709;"REAL J(99)"      !Declare array with 100 elements
30 OUTPUT 709;"DELVAR J"        !Release the memory allocated to the array
40 OUTPUT 709;"SIZE? J"         !Return the number of storage elements
50 END
```

When this program executes, 0 is returned to the mainframe display indicating there are no storage elements associated with array J.

DIM

• **Mainframe**

Description

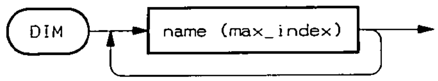
Declares a REAL (RL64) array in mainframe memory. DIM declares arrays only.

Prerequisites

None

Syntax

DIM *name (max_index)* [*name (max_index)* . . .]



Parameters

name (max_index)

Array name and size. *name* specifies the name of a REAL array and *max_index* specifies the maximum index of elements in the array. Since indices begin at 0, the number of elements in the array = *max_index* + 1. For example, DIM A(9) defines a REAL array of 10 elements, with elements 0 through 9.

name must be a string <= eight characters. It cannot begin with a digit and must contain only letters, digits, '_' and '?'. Parentheses are required around *max_index* number.

Remarks

Redeclaring Arrays

Executing DIM sets up a REAL array and fills the elements with zeroes. A REAL array if redeclared, must remain a REAL array. The redeclared array will be initialized to zeroes in all elements. (If the array is redeclared to the same size, it will just be zeroed.)

Arrays are Global

All arrays are global. Therefore, arrays dimensioned outside a subroutine are accessible in the subroutine and vice-versa.

DIM Statement Stored in Subroutine

If a DIM statement is stored in a subroutine, the array name is defined immediately but data storage for the array is not allocated until the subroutine is executed. (The name cannot be used for a subroutine, another array type, or variable.)

Recovering Storage Space With DELVAR

Send the DELVAR command to recover storage space allocated by the DIM command. The array name and type will remain but the memory allocated to the array is released. That is, the array becomes a REAL array with zero space.

Data Returned

None

Related Commands

REAL, DELVAR, SCRATCH

Examples

Example: Declaring a REAL Array With the DIM Command

The following program line declares a REAL array A of 11 elements with a starting index of 0.

```
100 OUTPUT 709;"DIM A(10)"   !Declares REAL array A of 11 elements
```

DISABLE

• **Mainframe**

Description

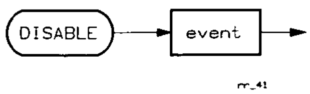
Disables event recognition and servicing.

Prerequisites

None

Syntax

DISABLE *event*



Parameters

event Interrupt or exception (alarm, limit) which causes an interrupt that is handled by the HP 3852A or by a controller. The following table describes the five *events*.

event	Description
INTR SYS	Disables the HP 3852A from recognizing an interrupt from an accessory. At power-on, INTR SYS is enabled. Front panel CLEAR key, the CLR command, or HP-IB device clear sets DISABLE INTR SYS.
INTR [USE ch]	Disables the accessory channel specified by USE from interrupting. At power-on, all accessory channels and slots are disabled.
LMT	Disables real-time limit exception testing. At power-on, LMT is disabled.
ALRM	Disables the HP 3852A real-time alarm from interrupting the HP 3852A or the controller. At power-on, ALRM is disabled. Front panel CLEAR key, the CLR command, or HP-IB device clear also sets DISABLE ALRM.
LOGCHAN	Disables measurement channel logging. At power-on, LOGCHAN is disabled.

Remarks

Index Pointer Unchanged for DISABLE LOGCHAN

DISABLE LOGCHAN does not alter the present position of the *var* array index pointer (refer to LOGCHAN command). For example, if the index pointer is currently set to element 20, it stays at element 20 after DISABLE LOGCHAN is executed.

Data Returned

None

Related Commands

ENABLE, LMT, SET ALRM, CLR

Examples

Example: Disable Events

The following program lines show how to disable events from generating interrupts.

```
              .
              .
20 OUTPUT 709;"DISABLE INTR SYS"      !Disable HP 3852A from recognizing
              .                       !interrupts from accessory channel
              .
50 OUTPUT 709;"DISABLE INTR, USE 201" !Disable accessory channel 1
              .                       !(slot 2) from interrupting
              .
80 OUTPUT 709;"DISABLE LMT"           !Disable limit exception testing
              .                       !from generating interrupt
              .
110 OUTPUT 709;"DISABLE ALRM"         !Disables system alarm from
              .                       !interrupting
              .
140 OUTPUT 709;"DISABLE LOGCHAN"      !Disables measurement channel
              .                       !data logging
              .
```

DISABLE EOL SWAP

- **Mainframe**

Description

Disable end of command swapping. When DISABLE EOL SWAP is set, swapping does not occur on command completion. A swap will occur, however, when the last command in the active task finishes. Execution of commands in higher priority tasks is also held off until the active task completes.

Prerequisites

The HP 3852A must be in the multitasking mode before DISABLE EOL SWAP can be executed. The HP 3852A multitasking capability is only available with firmware revision 3.0 or greater.

Syntax

DISABLE EOL SWAP

R.DIS

Parameters

None

Remarks

DISABLE EOL SWAP Does Not Affect Nested Subroutines

If DISABLE EOL SWAP is set and a subroutine calls another subroutine, the called subroutine executes to completion. The system then returns to the original subroutine where it executes to completion or until swapping is re-enabled with ENABLE EOL SWAP.

Suspending Command Execution When DISABLE EOL SWAP is Set

If command execution within a task is suspended by the WAITFOR SIGNAL, PAUSE, or SUSPEND command after DISABLE EOL SWAP is set, the next scheduled task will become active and its commands will begin to execute. If during a subsequent task SIGNAL occurs or the suspended period expires, the suspended task will resume; however, it may or may not immediately follow completion of the active task. It is recommended that swapping be enabled before a WAITFOR SIGNAL, PAUSE, or SUSPEND command is executed.

Data Returned

None

Related Commands

ENABLE MULTI, DISABLE MULTI, ENABLE EOL SWAP, URGENCY, TSLICE, NTASKS, RUN

Examples

Example: Disabling End of Command Swapping

DISABLE EOL SWAP is often used with ENABLE EOL SWAP to prevent a swap from occurring between a set of commands that must execute together.

In the following program, three subroutines are downloaded and directed to run tasks. Since the run tasks have the same priority (set when the HP 3852A enters the multitasking mode), the run tasks will time-slice and swap between each other. One of the run task subroutines performs a series of DC voltage measurements using the voltmeter's rear terminals and multiplexer channels. When the system first swaps to this run task, end of command swapping is disabled so that all of the rear terminal measurements are made at once. End of command swapping is then re-enabled and measurements using the multiplexer channels are made while time-slicing with the other run tasks.

```
10    !Set a 65 ms time-slice and set the number of run tasks and the
20    !queue size the system will allow. Place the HP 3852A in the
30    !multitasking mode. The reset caused by ENABLE MULTI will load
40    !the TSLICE and NTASKS parameters into the operating system.
50    !
60    OUTPUT 709;"TSLICE .065"
70    OUTPUT 709;"NTASKS 3,3"
80    OUTPUT 709;"ENABLE MULTI"
90    !
100   !Write and download subroutines to be directed to run tasks.
110   !Subroutine A when active, displays the numbers 0 through 599.
120   !Subroutine B when active, repeatedly displays A B C.
130   !
140   OUTPUT 709;"SUB A"
150   OUTPUT 709;"  INTEGER I"
160   OUTPUT 709;"  FOR I=0 TO 599"
170   OUTPUT 709;"    DISP I"
180   OUTPUT 709;"  NEXT I"
190   OUTPUT 709;"SUBEND"
200   OUTPUT 709;"SUB B"
210   OUTPUT 709;"  INTEGER J"
220   OUTPUT 709;"  FOR J=0 TO 599"
230   OUTPUT 709;"    DISP 'A'"
240   OUTPUT 709;"    DISP 'B'"
250   OUTPUT 709;"    DISP 'C'"
260   OUTPUT 709;"  NEXT J"
270   OUTPUT 709;"SUBEND"
280   !
290   !Write and download a subroutine that will make a series
300   !of DC voltage measurements and store the readings in mainframe
310   !arrays. Use DISABLE EOL SWAP and ENABLE EOL SWAP to enclose a
320   !set of commands so that the first series of measurements
330   !complete before a swap is allowed to occur.
340   !
350   OUTPUT 709;"SUB DCMEAS"
360   OUTPUT 709;"  INTEGER K"
370   OUTPUT 709;"  REAL DCRDGS1 (29)"
380   OUTPUT 709;"  REAL DCRDGS (299)"
```

```
390   OUTPUT 709;"  USE 700"
400   OUTPUT 709;"    CONF DCV"
410   OUTPUT 709;"    RANGE 5"
420   OUTPUT 709;"    TERM EXT"
430   !
440   !Prevent a swap from occurring until all 30 measurements are
450   !taken and stored.
460   !
470   OUTPUT 709;"  DISABLE EOL SWAP"
480   OUTPUT 709;"  FOR K = 0 TO 29"
490   OUTPUT 709;"    TRIG SGL"
500   OUTPUT 709;"    CHREAD 700 INTO DCRDGS1"
510   OUTPUT 709;"  NEXT K"
520   !
530   !Enable swapping to occur. This will enable the remaining 300 DC
540   !voltage measurements to be taken while time-slicing with the
550   !other run task subroutines which display 1,2,3 ... and A B C.
560   !
570   OUTPUT 709;"  ENABLE EOL SWAP"
580   OUTPUT 709;"  CONF DCV"
590   OUTPUT 709;"  FOR K=0 TO 99"
600   OUTPUT 709;"    MEAS DCV 600-602 INTO DCRDGS"
610   OUTPUT 709;"  NEXT K"
620   OUTPUT 709;"SUBEND"
630   !
640   !Direct the subroutines to be run tasks. Set each run task to
650   !run one time.
660   !
670   OUTPUT 709;"RUN 1 A"
680   OUTPUT 709;"RUN 2 B"
690   OUTPUT 709;"RUN 3 DCMEAS"
700   END
```

By looking at the display as the program executes, you will notice that when the system swaps to run task 3, CHREAD is displayed for a relatively long period of time. This indicates that end of command swapping was disabled (line 470) while measurements were taken on the rear terminals of the voltmeter (lines 480/510). When end of command swapping is re-enabled (line 570), swapping occurs between each of the run tasks as is evidenced by the periodic displays of letters, numbers, and of the MEAS command.

- **HP 44701A Integrating Voltmeter**
- **HP 44702A/B High-Speed Voltmeter (Scanner or System Mode)**

Description

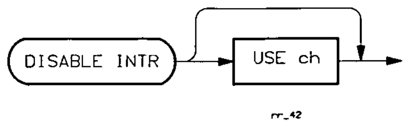
Channel interrupt disable. DISABLE INTR prevents the voltmeter from generating an interrupt when a reading becomes available.

Prerequisites

None

Syntax

DISABLE INTR [USE *ch*]



Parameters

USE *ch* Voltmeter slot number. See Glossary.

Remarks

Interrupts Automatically Cleared and Disabled

When an interrupt occurs on a channel which has been enabled to interrupt (with ENABLE INTR), the interrupt is disabled when it is serviced and is cleared when all measurements are read from the voltmeter. Note that CONF disables and clears interrupts.

Data Returned

None

Related Commands

ENABLE INTR, INTR?, USE, DISABLE INTR SYS, CONF

Examples

Example: Disabling Voltmeter Interrupt

In the following program segment, line 30 enables interrupt capability on a voltmeter in slot 1, while line 40 disables the interrupt capability.

```
10 OUTPUT 709;"RST"            !Reset the HP 3852A
20 OUTPUT 709;"USE 100"        !Use voltmeter in slot 1
30 OUTPUT 709;"ENABLE INTR"    !Enable interrupt for voltmeter
40 OUTPUT 709;"DISABLE INTR"   !Disable the interrupt
                .
                .
```

DISABLE INTR

• HP 44714A 3-Channel Stepper Motor Controller/Pulse Output

Description

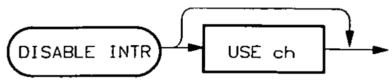
Disables the channel specified from providing an interrupt when a move is completed.

Prerequisites

Requires mainframe firmware revision 3.0 or greater.

Syntax

DISABLE INTR [USE *ch*]



Parameters

USE *ch* — Specifies the channel that is to be disabled from interrupting. Channel number range can be ES00 to ES02.

Remarks

DISABLE INTR Clears and Disables Interrupts

The DISABLE INTR command disables a channel from interrupting when a move has completed (when a pulse train stops). The command also clears an interrupt from the specified channel.

Data Returned

None

Related Commands

ENABLE INTR, DISABLE INTR SYS, INTR?, USE

Examples

Example: Disable Channel 2 From Interrupting

```
10 OUTPUT 709;"ENABLE INTR USE 202"
                  .
                  .
                  .
50 OUTPUT 709;"DISABLE INTR USE 202"
                  .
                  .
```

- **HP 44715A 5-Channel Counter/Totalizer**

Description

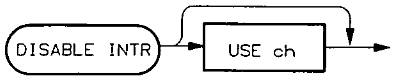
Disable counter channel from interrupting on counter overflow or when the reading is available.

Prerequisites

None

Syntax

DISABLE INTR [USE *ch*]



Parameters

USE *ch* — Address of channel. Channel number range (depends on HP 44715A hardware configuration) = ES00 through ES04.

Remarks

Interrupt Cleared/Disabled

If a channel is set for the TOTAL, TOTALM, UDC, UDCM, CD, or CDM functions, DISABLE INTR disables and clears the channel interrupt so that a subsequent ENABLE INTR will not generate an interrupt.

For the RAT, PER, PERD, functions or for the Frequency configuration, a DISABLE INTR command will clear but not disable a pending interrupt so a subsequent ENABLE INTR could generate an interrupt. The interrupt is cleared when the measurement is read.

If the function (range, edge, etc.) of a disabled channel is changed, the channel remains disabled but any pending interrupts are cleared. CONF disables and clears interrupts.

Data Returned

None

Related Commands

ENABLE INTR, INTR?, USE, DISABLE INTR SYS, CONF

Examples

Example: Disable Channel for Interrupt

These program lines disable channel 3 of a counter in slot 2 of the mainframe from interrupting on counter overflow. The channel counts positive edges and is set to the TOTAL function.

```
10 OUTPUT 709;"RST"            !Reset the HP 3852A and HP 44715A
20 OUTPUT 709;"USE 203"        !Use channel 3 in slot 2
30 OUTPUT 709;"CONF TOTAL"     !Set TOTAL function
40 OUTPUT 709;"ENABLE INTR"    !Enable interrupt on ch 3
                      .
                      .
90 OUTPUT 709;"DISABLE INTR"   !Disable ch 3 interrupt capability
                      .
                      .
```

DISABLE INTR

- **HP 44721A 16-Channel Digital Input**
- **HP 44722A 8-Channel Digital Input**

Description

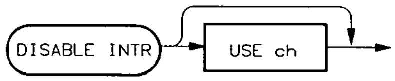
Disable digital input channels for event interrupt or for counter interrupt.

Prerequisites

None

Syntax

DISABLE INTR [USE *ch*]



Parameters

USE *ch*

Specifies type of interrupt to be disabled on channel. For *ch* = ES00-ES15 (ES00-ES07 for 8-channel), channel is disabled from interrupting on counter overflow (counter interrupt). For *ch* = ES16-ES31 (ES08-ES15 for 8-channel), channel is disabled from interrupting on edge specified by EDGE command (event interrupt).

Remarks

DISABLE INTR Clears and Disables Interrupts

The DISABLE INTR command disables a channel from interrupting on event or counter overflow, depending on the *ch* value set by the USE *ch* parameter. The command also clears an interrupt from the specified source.

Event and Counter Interrupts Disabled Separately

Since a physical channel has two associated logical channel numbers, a physical channel can be enabled for both event and counter interrupts. Therefore, to completely disable a physical channel from interrupting, both interrupt conditions must be disabled. That is, disabling a physical channel for event interrupts does not disable the channel for counter interrupts and vice-versa. CONF disables and clears both interrupt conditions.

Data Returned

None

Related Commands

ENABLE INTR, INTR?,USE, DISABLE INTR SYS

Examples

Example: Disable Channel for Interrupts

The following program lines show how to enable and disable interrupts on channel 6 of an HP 44721A 16-channel digital input in slot 2 of the mainframe. (For an HP 44722A 8-channel digital input, change USE 222 to USE 214 in lines 20 and 80.)

Note that if line 80 is not sent, channel 6 is disabled for counter interrupts, but is still enabled for event interrupts. If line 70 is not sent, channel 6 is disabled for event interrupts, but enabled for counter interrupts. Since both types of interrupts are enabled for the channel, both types must be disabled to completely disable the channel.

```
10 OUTPUT 709;"ENABLE INTR, USE 206"   !Enable ch 6 for counter interrupt
20 OUTPUT 709;"ENABLE INTR, USE 222"   !Enable ch 6 for event interrupt
                  .
                  .
70 OUTPUT 709;"DISABLE INTR, USE 206"  !Disable counter interrupt on ch 6
80 OUTPUT 709;"DISABLE INTR, USE 222"  !Disable event interrupt on ch 6
                  .
                  .
```

• **HP 44723A 16-Channel High-Speed Digital Sense/Control**

Description

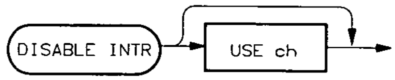
Disables interrupts for the specified channel. Types of interrupts which can be disabled are edge (channel range ES00-ES15 and ES93), pattern (ES90), input (ES91), and output (ES92).

Prerequisites

Requires mainframe firmware revision 3.0 or greater.

Syntax

DISABLE INTR [USE *ch*]



Parameters

USE *ch* Sets the applicable channel/function for interrupts to be disabled. The types of interrupts which can be enabled by ENABLE INTR and the action of DISABLE INTR for each type follow. At power-on or following a RST or RST *slot* command, all interrupts are disabled. All interrupts are also cleared except for output interrupts which are set.

USE ch	Type	Description
ES00-ES15	Edge	When enabled, edge interrupts occur when the edge programmed by EDGE is seen at the channel input. DISABLE INTR disables and clears edge interrupts.
ES90	Pattern	When enabled, pattern interrupts occur when user inputs match the pattern/mask condition set by PATTERN. DISABLE INTR disables and clears pattern interrupts.
ES91	Input	When enabled, input interrupts occur on a first rank input trigger and are cleared by a second rank input trigger. DISABLE INTR disables but does not clear input interrupts. Input interrupts are cleared only by a second rank input trigger, at power-on, or by a reset or slot reset command.
ES92	Output	When enabled, output interrupts occur on a second rank output trigger and are cleared by a write to the first rank output register (with CHWRITE, CHWRITEM, WRITE, or WRITEM).

		DISABLE INTR disables but does not clear output interrupts. Output interrupts are cleared only by a write to the first rank output register.
ES93	Edge	When enabled, edge interrupts occur when the edge programmed by EDGE is seen at <u>any</u> input channel. DISABLE INTR disables and clears edge interrupts.

Remarks

Data Returned

None

Related Commands

ENABLE INTR, EDGE, PATTERN

Examples

Example: Disabling Interrupts

The following program lines show how to disable edge, pattern, input and output interrupts.

```
        .
        .
100 OUTPUT 709;"DISABLE INTR USE 214"  !Disable edge intr for ch 214
110 OUTPUT 709;"DISABLE INTR USE 390"  !Disable pattern intr for slot 300
120 OUTPUT 709;"DISABLE INTR USE 91"   !Disable input intr for slot 0
130 OUTPUT 709;"DISABLE INTR USE 192"  !Disable output intr for slot 100
        .
        .
```

- **HP 44726A 2-Channel Arbitrary Waveform DAC**

Description

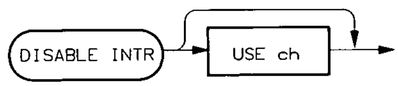
Channel interrupt disable. DISABLE INTR prevents the DAC accessory from generating an interrupt after NSCAN cycles of the waveform have occurred. Executing DISABLE INTR will also clear an interrupt that has not yet been serviced.

Prerequisites

Requires firmware revision 3.5 or greater when a DAC channel is specified.

Syntax

DISABLE INTR [USE *ch*]



Parameters

USE *ch* Channel disabled from interrupting after NSCAN cycles of the waveform have occurred. The range for *ch* is ES00 to ES01. The default USE *ch* is channel 0. At power-on or following a reset, channel interrupt capability is disabled.

Remarks

HP 44726A Interrupts Automatically Cleared and Disabled

When a channel has been enabled and an interrupt occurs, the interrupt is cleared and the channel set to DISABLE INTR when the interrupt is serviced.

Data Returned

None

Related Commands

ENABLE INTR

Examples

Example: Disabling HP 44726A Interrupts

The following program line prevents an interrupt from occurring on channel 1 following NSCAN cycles of any waveform selected. DISABLE INTR can be executed at any point in the program.

```
100   OUTPUT 709;"DISABLE INTR USE 1"
```

DISABLE MULTI

- **Mainframe**

Description

Disable HP 3852A multitasking capability. Execution of this command resets the HP 3852A and removes the instrument from the multitasking mode.

Prerequisites

The HP 3852A multitasking capability is only available with mainframe firmware revision 3.0 or greater.

Syntax

DISABLE MULTI

Parameters

None

Remarks

DISABLE MULTI Causes a System Reset

When DISABLE MULTI is executed, a system reset occurs. The reset returns each installed accessory to its power-on state and also erases all variables, arrays, and downloaded subroutines declared while the HP 3852A was in the multitasking mode. The reset clears the mainframe's input and output buffers, however, the size of the buffers and the size of the symbol table remains as last set by the INBUF, OUTBUF, and SYMSIZE commands or as set at power-on.

DISABLE MULTI Sets Specific Task Priorities

Regardless of the task priorities set while the HP 3852A was in the multitasking mode (URGENCY command), DISABLE MULTI sets the following:

```
Interrupt     (highest)
Front Panel
HP-IB         (lowest)
```

Note that you cannot set/change task priorities outside of the multitasking mode. Also, run tasks are not allowed.

Exiting the Multitasking Mode

Once in the multitasking mode, the mainframe remains in that mode until power is cycled or until DISABLE MULTI is executed.

Effect of DISABLE MULTI on the USE Channel

When DISABLE MULTI is executed, all subsequent settings of the USE channel are global. This means a USE channel set from the front panel, HP-IB, or within an interrupt-called subroutine will be used by commands from any of these origins unless changed by another USE command or by the USE channel parameter.

Commands Not Allowed when Multitasking is Disabled

The following commands can only be executed in the multitasking mode.

ABORT	SUSPEND
ENABLE EOL SWAP	RUN?
RUN	SIGNAL
WAITFOR SIGNAL	URGENCY

Data Returned

None

Related Commands

ENABLE MULTI, ENABLE EOL SWAP, DISABLE EOL SWAP, URGENCY, TSLICE, NTASKS, RUN

Examples

The DISABLE MULTI command can be executed at any time in order to remove the HP 3852A from the multitasking mode.

```
100   OUTPUT 709;"DISABLE MULTI"
```

DISABLE PROBE

• **Mainframe**

Description

Disables the operating system probe (PROBE command).

Prerequisites

Requires firmware revision 3.5 or greater.

Syntax

DISABLE PROBE

Parameters

None

Remarks

Disable the Probe Prior to Redeclaring the Probe Arrays

When redeclaring the probe arrays in order to clear the arrays and reset the index pointers, the probe should first be disabled. This ensures that once they are redeclared, probe data will not be placed in the arrays until instructed by the user.

Data Returned

None

Related Commands

PROBE

Examples

Example: Disabling the Probe

The following program segment is an example of when the probe might be disabled so that the probe arrays can be redeclared prior to a new trace.

```
10   OUTPUT 709;"INTEGER PRV_STAT(17),NEW_TASK(17)"
20   OUTPUT 709;"REAL PERIOD(17)"
      .
      .
110  OUTPUT 709;"PROBE PRV_STAT,NEW_TASK,PERIOD"
      .
      .
170  OUTPUT 709;"DISABLE PROBE"
180  OUTPUT 709;"INTEGER PRV_STAT(17),NEW_TASK(17)"
190  OUTPUT 709;"REAL PERIOD(17)"
200  OUTPUT 709;"PROBE PRV_STAT,NEW_TASK,PERIOD"
      .
      .
```

DISABLE/ENABLE DAC

- **HP 44726A 2-Channel Arbitrary Waveform DAC**

Description

DISABLE DAC sets the output of the DAC OUT BNC to 0V. ENABLE DAC allows the DAC to output a value other than 0V.

Prerequisites

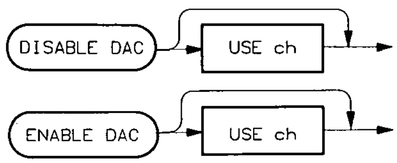
Requires firmware revision 3.5 or greater.

Syntax

DISABLE DAC [USE *ch*]

or

ENABLE DAC [USE *ch*]



Parameters

USE *ch* — Channel whose output is disabled and set to 0V, or the channel whose output is enabled. The default **USE** *ch* is channel 0. At power-on or following a reset, both channels are enabled.

Remarks

Waveforms Continue Running when the DAC is Disabled

If DISABLE DAC is executed while a waveform is at the output of the DAC, the output is set to 0V, however, data from the selected waveform is still sent to the input of the DAC. As a result, NSCAN cycles of the waveform are made and any programmed interrupts will still occur.

When the DAC output is re-enabled, the waveform resumes when the next trigger advancing the memory occurs. The point at which the waveform resumes depends on the number of trigger pulses that occurred while the DAC was disabled.

If APPLY DCV or APPLY PERC was executed prior to DISABLE DAC, the output is also set to 0V. When the output is re-enabled, it remains at 0V until APPLY DCV or APPLY PERC is executed again.

Examples

Example: Enabling and Disabling the Output of the DAC

The following program shows how DISABLE DAC and ENABLE DAC are used to control a signal supplied by the DAC.

In the example, a test signal is defined and stored on channel 0. Keys 1 and 9 on the HP 3852A front panel are assigned softkey definitions of ENABLE DAC and DISABLE DAC respectively. When the program runs, the test signal is applied to the circuit since the DAC is enabled. By pressing key 9, the user can disable the DAC output. Later, key 1 can be pressed which enables the DAC to re-apply the signal.

```
10    !Reset the mainframe and the HP 44726A. Assign softkey definitions
20    !to front panel keys 1 and 9. Softkey 1 allows the operator to
30    !enable the DAC in order to continue testing. Softkey 9 allows
40    !the operator to disable the DAC to suspend testing.
50    !
60    OUTPUT 709;"RST"
70    WAIT 1
80    OUTPUT 709;"EDIT KEY 1""ENABLE DAC"""
90    OUTPUT 709;"EDIT KEY 9""DISABLE DAC"""
100   !
110   !Define and store the test signal. The signal will be a 500 point
120   !triangle wave, 1 kHz, 5 Vp-p. Select the waveform from channel 0
130   !memory. Arm the channel so that when triggers are
140   !received, the data in that portion of memory is sent to the DAC
150   !which then generates the waveform.
160   !
170   OUTPUT 709;"WFWRITE RPV 4 5 PTS 500 TBASE (1/(1E3*500))"
180   OUTPUT 709;"APPLY WFV 0,4"
190   OUTPUT 709;"TRIG INT"
200   OUTPUT 709;"TARM AUTO"
210   END
```

• Mainframe

Description

DISABLE LABELS prevents command headings (labels) and data from appearing in the display as commands are executed.

ENABLE LABELS stores command headings (labels) internally which are then displayed along with the data as the commands execute.

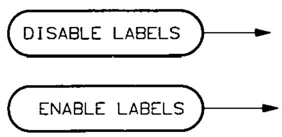
Prerequisites

Requires firmware revision 3.5 or greater.

Syntax

DISABLE LABELS

ENABLE LABELS



Parameters

None

Remarks

Using DISABLE LABELS When Downloading Subroutines

If DISABLE LABELS is executed prior to downloading a subroutine and ENABLE LABELS is executed prior to executing the subroutine, the labels of the subroutine commands are not displayed since the labels were disabled when the subroutine was compiled.

Using DISABLE LABELS in the Multitasking Mode

DISABLE LABELS is particularly useful for clearing the display while the HP 3852A is in the multitasking mode. This is often desirable since the swapping of tasks removes any correlation between the labels displayed and the function performed.

DISABLE LABELS Allows Messages to be Displayed.

Unlike DISP OFF, DISABLE LABELS will allow user-defined messages to be displayed.

CLR (Clear) and RST (Reset) set ENABLE LABELS

Anytime you clear (CLR) or reset (RST) the HP 3852A, ENABLE LABELS is set. This causes command headings to again be stored internally.

Data Returned

None

Related Commands

DISP

Examples

Example: Increasing Command Execution Speed

The following segment shows one method of programming the HP 3852A to execute commands at the fastest possible rate.

```
10    OUTPUT 709;"DISP OFF"
20    OUTPUT 709;"DISABLE LABELS"
30    OUTPUT 709;"SUB A"
40    OUTPUT 709;"  REAL DCRDGS(9)"
50    OUTPUT 709:"  CONFMEAS DCV 100-109 USE 700 INTO DCRDGS"
60    OUTPUT 709;"SUBEND"
70    OUTPUT 709;"ENABLE LABELS"
80    OUTPUT 709;"CALL A"
      .
      .
      .
```

With the display off, neither commands or data are displayed. With DISABLE LABELS, command headings are not stored internally. When commands are within subroutines, they are already compiled.

Example: Displaying Messages Only

The following example demonstrates how with DISABLE LABELS set, user-defined messages are still displayed.

```
10     OUTPUT 709;"DISABLE LABELS"
20     OUTPUT 709;"SUB ADJUST"
30     OUTPUT 709;"  REAL SET_PNT,T"
40     OUTPUT 709;"  SET_PNT = 26"
50     OUTPUT 709;"  USE 700"
60     OUTPUT 709;"  CONF THM10K"
70     OUTPUT 709;"  WHILE 1"
80     OUTPUT 709;"    MEAS THM10K 600 INTO T"
90     OUTPUT 709;"    IF T < SET_PNT THEN"
100    OUTPUT 709;"      DISP 'UP'"
110    OUTPUT 709;"    END IF"
120    OUTPUT 709;"    IF T > SET_PNT THEN"
130    OUTPUT 709;"      DISP 'DOWN'"
140    OUTPUT 709;"    END IF"
150    OUTPUT 709;"  END WHILE"
160    OUTPUT 709;"SUBEND"
170    OUTPUT 709;"ENABLE LABELS"
180    OUTPUT 709;"CALL ADJUST"
190    END
```

As this program executes, subroutine ADJUST is downloaded and temperature measurements are made continuously. When the temperature is below 26° C, "UP" is displayed on the HP 3852A - telling the operator to turn the temperature up. When the temperature exceeds 26°, 'DOWN' is displayed - telling the operator to turn the temperature down. Note that with DISABLE LABELS set, only UP or DOWN are displayed. Without DISABLE LABELS, the commands as they are executed also appear in the display.

- **Mainframe**

Description

Disable the mainframe display or display a user-defined string or message.

Prerequisites

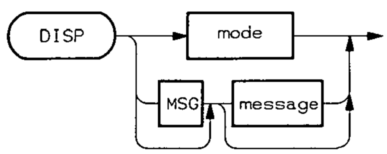
None

Syntax

DISP *mode*

or

[MSG] [*message*]



RR_43

Parameters

mode The *mode* parameter enables or disables the display mode or defines the message to be displayed.

mode	Definition
ON	Enables the front panel display. Commands and results appear on the front panel display, if MON ON or MON ALL is set (refer to MON command). At power-on, display is ON.
OFF	Disables the front panel display. Annunciators are suppressed and no results are displayed, although the keyboard continues to function normally.

MSG Indicates a message follows.

message Quoted string of characters (quotes are not displayed) or a numeric expression to be evaluated. When a string of characters is to be displayed, either double quotation marks (") or single quotation marks (') may be used. If no *message* is specified, the display is cleared.

Remarks

DISP OFF Improves System Performance

Using DISP OFF can greatly improve system performance by speeding up command execution. Thus, for example, DISP OFF is an excellent technique to use when doing high-speed scans.

Data Returned

Quoted string of characters or value of numeric expression to be evaluated is returned to the front panel ONLY.

Related Commands

DISABLE/ENABLE LABELS, MON

Examples

Example: Disabling the Display Mode

The following program line shows how to disable the display mode. With display mode disabled, all commands entered from the front panel are displayed, but results are not. If commands are entered over the HP-IB, neither the command or result is displayed.

```
10 OUTPUT 709;"DISP OFF"      !Disable the display mode.
```

Example: Displaying a Quoted String

The following program lines show four ways to display the same quoted string (message) on the front panel display. Note that the display mode must be enabled (DISP ON) to display the message.

```
10 OUTPUT 709;"DISP 'DISPLAY MY MESSAGE'"
                     or
10 OUTPUT 709;"DISP ""DISPLAY MY MESSAGE"""
                     or
10 OUTPUT 709;"DISP MSG 'DISPLAY MY MESSAGE'"
                     or
10 OUTPUT 709;"DISP MSG ""DISPLAY MY MESSAGE"""
```

Example: Displaying a Numeric Expression

The following program shows how to display a numeric expression on the front panel, assuming the display mode is ON. Note that the number is displayed in Real number (RASC) format.

```
10 OUTPUT 709;"DISP (4/2)"
20 END
```

The value displayed is:

```
2.000000E+00
```

DIV

- **Math Function**

Description

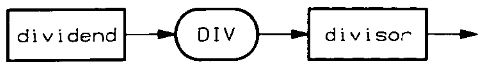
Returns the integer portion of a number obtained by the division of one number by another.

Prerequisites

Requires mainframe firmware revision 3.5 or greater.

Syntax

dividend **DIV** *divisor*



38521P R.DIV

Parameters

dividend Constant, variable, or numeric expression.

divisor Constant, variable, or numeric expression.

Remarks

Data Returned

Returned is the integer portion of a number obtained by dividing one number by another.

Related Commands

MOD

Examples

Example: Demonstrating the DIV Function

The following program gives an example of the data returned by the DIV function.

```
10     OUTPUT 709;"RST"
20     OUTPUT 709;"INTEGER A,B"
30     OUTPUT 709;"A=49"
40     OUTPUT 709;"B=6"
50     OUTPUT 709;"VREAD (A DIV B)"
60     ENTER 709;Rslt
70     PRINT Rslt
80     END
```

When the program executes, 8 is displayed on the controller.

DONE?

• **HP 44714A 3-Channel Stepper Motor Controller/Pulse Output**

Description

Queries the HP 44714A to determine if a move has completed and if so, under what conditions the move terminated. A code is returned indicating how the move completed (see the following table).

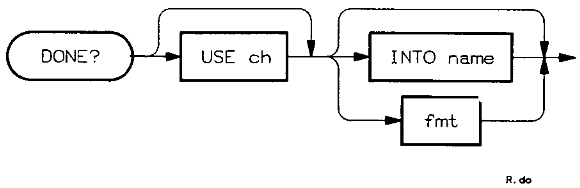
CODE	MEANING
-3	Waiting for a trigger
-2	Running
-1	SUSTAIN running/Ready for next command
0	Move completed successfully
1	Move aborted because of halt
2	Move aborted because of hard limit

Prerequisites

Requires mainframe firmware revision 3.0 or greater.

Syntax

DONE? [USE *ch*] [INTO *name*] or [*fmt*]



Parameters

USE *ch* Channel that is queried. Channel range is ES00 to ES02.

INTO *name* See Glossary.

fmt See Glossary. Default format is IASC.

Remarks

Data Returned

An integer number from -3 through +2

Related Commands

MOVE, SUSTAIN, TRIG, HALT, HARDLIM

Examples

Example: Determine If a Move Has Completed

```
10 OUTPUT 709;"DONE?, USE 201"
20 ENTER 709; VALUE
30 PRINT VALUE
40 END
```

• HP 44715A 5-Channel Counter/Totalizer

Description

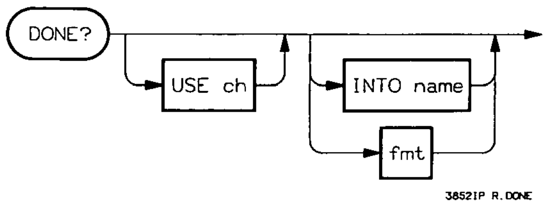
Determines if the counter measurement is complete. By monitoring the measurement, other HP 3852A operations can occur rather than sending a command to retrieve the data and having to wait until the data is available.

Prerequisites

Requires mainframe firmware revision 3.5 or greater.

Syntax

DONE? [USE *ch*] [INTO *name*] or [*fmt*]



Parameters

USE *ch* Counter channel on which the measurement is being performed.

INTO *name* See Glossary.

fmt See Glossary. Default format is IASC.

Remarks

DONE? Should Not be Used During Frequency Measurements

When measuring frequency, the counter's microprocessor is dedicated to the measurement. Thus, the accessory does not respond to DONE? until the measurement is complete.

Data Returned

DONE? returns the following information based on the status of the measurement.

1	Measurement is complete and data is available.
0	Measurement is not complete, counter is busy.
-1	Measurement has not been triggered.

Related Commands

None

Examples

Example: Using DONE? to Monitor the Status of Period Measurements

The following example shows how DONE? is used to determine when a period measurement is complete.

DONE? (cont)

The program downloads two subroutines. One subroutine will eventually count to 1000, while the other measures the period of an input signal. If DONE? indicates the measurement is finished (1 is returned), the CHREAD command is executed to retrieve the period measurement. If the measurement is not finished, the counting subroutine is called. This demonstrates that by using DONE?, the mainframe can monitor the measurement and perform other functions while waiting for the measurement to complete. Otherwise, executing CHREAD before the measurement is finished prevents the mainframe from performing other tasks until the measurement completes.

Note that the count and the period measured are alternately displayed on the HP 3852A.

```
10    !Clear and reset the HP 3852A. Declare a counting variable
20    !and download a "counting" subroutine.
30    !
40    CLEAR 709
50    OUTPUT 709;"RST"
60    OUTPUT 709;"INTEGER I"
70    OUTPUT 709;"SUB COUNT"
80    OUTPUT 709;"  DISP I"
90    OUTPUT 709;"  I=I+1"
100   OUTPUT 709;"SUBEND"
110   !
120   !Download a subroutine which measures the period of the input
130   !signal after five periods have occurred.
140   !
150   OUTPUT 709;"SUB EXECMEAS"
160   OUTPUT 709;"  INTEGER ISTATS"
170   OUTPUT 709;"  REAL RSLTS"
180   OUTPUT 709;"  USE 200"
190   OUTPUT 709;"    CONF PER"
200   OUTPUT 709;"    NPER  5"
210   OUTPUT 709;"    TRIG SGL"
220   !
230   !Use DONE? to determine if data is available (measurement has
240   !been made). If it is, read the period using CHREAD. If it
250   !isn't, call the counting subroutine. Execute the program
260   !until a count of 1000 is reached.
270   !
280   OUTPUT 709;"  WHILE I <= 1000"
290   OUTPUT 709;"    DONE? INTO ISTATS"
300   OUTPUT 709;"    IF ISTATS=1 THEN"
310   OUTPUT 709;"      CHREAD 200 INTO RSLTS"
320   OUTPUT 709;"      VREAD RSLTS"
330   OUTPUT 709;"      TRIG SGL"
340   OUTPUT 709;"      WAIT 1"
350   OUTPUT 709;"    ELSE"
360   OUTPUT 709;"      CALL COUNT"
370   OUTPUT 709;"    END IF"
380   OUTPUT 709;"  END WHILE"
390   OUTPUT 709;"SUBEND"
400   OUTPUT 709;"CALL EXECMEAS"
410   END
```

• **HP 44715A 5-Channel Counter/Totalizer**

Description

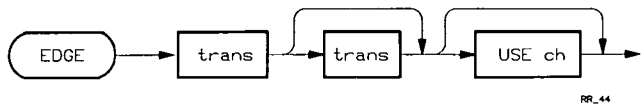
Set the edges (positive or negative) which the channel counts or sets the level (high or low) for which the gate is active. Channel is selected with USE *ch*.

Prerequisites

None

Syntax

EDGE *trans* [*trans*] [USE *ch*]



Parameters

trans

trans has four values: HL, LH, HI, or LO. The first *trans* applies to the A input, the second *trans* to the B input. The second *trans* parameter is illegal for 5-channel TOTAL or FREQ configuration; for channel 2 in 3-CH configuration; or for channels 1,2, and 3 in 4-CH configuration.

If the second *trans* parameter is not specified for a channel with a double-input channel, the value in the first *trans* parameter is used for both inputs. Power-on value = LH or HI.

trans	Description
LH	Low-to-high transitions (equivalent to HI). Useful for functions which count or measure frequency. For Counts/Direction (CD) or for Counts/Direction, Modulo NPER (CDM) when the B input is set for LH, A input counts are up-counts when the B input is high, down-counts when the B input is low.
HL	High-to-low transitions (equivalent to LO). Useful for functions which count or measure frequency. For Counts/Direction (CD) or for Counts/Direction, Modulo NPER (CDM) when the B input is set for HL, A input counts are down-counts when the B input is high, up-counts when the B input is low.
HI	High Level (equivalent to LH). Useful for functions which use inputs as gates or direction.
LO	Low Level (equivalent to HL). Useful for functions which use inputs as gates or direction.

USE *ch* — Channel address. See Glossary. Channel number range (depends on HP 44715A hardware configuration) = ES00 through ES04.

Remarks

Counts/Direction Function

For the CD and CDM functions, when the B input (direction) is set for EDGE HI or EDGE LH, the A input counts are up-counts when the B input is high, down-counts when the B input is low. When the B input is set for EDGE LO or EDGE HL, the A input counts are down-counts when the B input is high, up-counts when the B input is low.

Gating Total Counts

For the TOTAL (Gated Total Counts) and TOTALM (Gated Total Counts, Modulo NPER) functions, the EDGE *trans* parameter for the B input sets the gating level. For EDGE LH,HI or EDGE HL,HI the gate is enabled (and the A input counted) ONLY when the B input is high. For EDGE LH,LO or EDGE HL,LO the gate is enabled (and the A input counted) ONLY when the B input is low.

Data Returned

None

Related Commands

None

Examples

Example: Set Channel for Counts/Direction

This program segment sets channel 3 of a counter in slot 5 of the mainframe to the Counts/Direction (CD) function. The A input is counted on high-to-low transitions. The count is up-count when the B input is low, down-count when the B input is high.

```
10 OUTPUT 709;"RST"            !Reset the HP 3852A and HP 44715A
20 OUTPUT 709;"USE 503"        !Use ch 3 in slot 5 for following cmds
30 OUTPUT 709;"FUNC CD"        !Set counts/direction function
40 OUTPUT 709;"EDGE HL, LO"    !Count high-to-low transitions of A
50                             !input.  Count up when B input is low,
60                             !count down when B input is high.
         .
         .
         .
```

- **HP 44721A 16-Channel Digital Input**
- **HP 44722A 8-Channel Digital Input**

Description

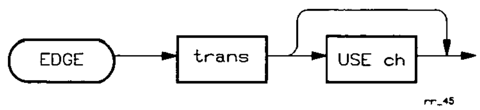
Sets the edge(s) (positive, negative, or both) which the specified channel will count or sets the edge transition which will cause an interrupt.

Prerequisites

EDGE BOTH requires mainframe firmware revision 3.0 or greater plus an HP 44721A with serial number 2711A01765 or greater or an HP 44722A with serial number 2711A00178 or greater. Also, USE *ch* = ES90 requires mainframe firmware revision 3.0 or greater.

Syntax

EDGE *trans* [USE *ch*]



Parameters

trans Sets the edge(s) to be counted or the transition which will cause an interrupt. Power-on and reset *trans* = OFF.

trans	Description
OFF	Neither edge.
LH	Count or interrupt on low-to-high edge.
HL	Count or interrupt on high-to-low edge.
BOTH*	Count or interrupt on either edge.

* = Valid only for mainframe firmware revision 3.0 or greater.

USE *ch* Sets the channel(s) on which edge(s) are to be counted or detected.

USE ch	Description
ES00-ES15 [1]	Count specified edge(s) on channel specified by USE ch. When enabled, channel specified by USE ch interrupts on counter overflow (counter interrupt).
ES16-ES31 [2]	Detect specified edge(s) on channel specified by USE ch. When enabled, channel specified by USE ch interrupts when the specified edge occurs (event interrupt).

ES90	[3]	When enabled, event interrupt is generated when the specified edge(s) occur on <u>any</u> channel in range ES16-ES31 (ES08-ES15 for the HP 44722A).

[1] = ES00-ES07 for the HP 44722A.
[2] = ES08-ES15 for the HP 44722A.
[3] = Valid only for mainframe firmware revision 3.0 and greater.

Remarks

EDGE Sets Edge for Both Event and Counter Interrupts

The EDGE command sets the edge(s) (positive, negative, or both) for BOTH the counting function and interrupt transition on a channel, regardless of which channel (counter or event) is specified. Thus, for example, for the same physical channel you cannot set positive edges to be counted and a negative edge transition to generate an interrupt. (See the ENABLE INTR command description.)

Data Returned

None

Related Commands

ENABLE INTR

Examples

Example: Set Edge to Count Input

This program enables channel 206 of a HP 44724A in slot 2 of the mainframe to count positive (LH) edges and to display the count after 10 seconds.

```
10  OUTPUT 709;"USE 206"       !Set channel 206 to count edges
20  OUTPUT 709;"CNTSET 0"      !Ensure ch counter set to 0
30  OUTPUT 709;"EDGE LH"       !Count ch 206 positive edges
40  WAIT 10                    !Wait 10 seconds
50  OUTPUT 709;"EDGE OFF"      !Stop counting
60  OUTPUT 709;"CHREAD 206"    !Read ch 206 counts
70  ENTER 709;A                !Enter ch 206 counts
80  PRINT "Counts = ";A        !Display ch 206 counts
90  END
```

A typical display for 5 counts on channel 206 (5 positive edges) is:

```
Counts = 5
```

- **HP 44723A 16-Channel High-Speed Digital Sense/Control**

Description

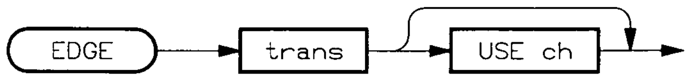
Sets the edge(s) (positive, negative, or both) of the specified input channel which will generate an edge interrupt when enabled.

Prerequisites

Requires mainframe firmware revision 3.0 or greater.

Syntax

EDGE *trans* [USE *ch*]



r.ed

Parameters

trans Sets the edge transition that will cause an interrupt when enabled. Power-on and reset *trans* = OFF.

trans	Description
OFF	Neither edge.
LH	Interrupt on low-to-high edge.
HL	Interrupt on high-to-low edge.
BOTH	Interrupt on either edge.

USE *ch* For *ch* = ES00-ES15, when enabled by ENABLE INTR, an edge interrupt occurs when the edge(s) specified by *trans* is seen at the input channel specified by *ch*. For *ch* = ES93, when enabled by ENABLE INTR, an edge interrupt occurs when the edge(s) specified by *trans* is seen at any input channel.

Remarks

Edge Interrupt Cleared/Disabled

Edge interrupts are generated from user input data. EDGE clears any pending edge interrupts on the channel. Edge interrupts are cleared and disabled when serviced by the mainframe. Also, DISABLE INTR clears and disables edge interrupts on the channel. Note that ENABLE INTR is also required for an edge interrupt to occur.

Data Returned

None

Related Commands

ENABLE INTR

Examples

Example: Enable Edge Interrupt

This program enables channel 206 of an HP 44723A in slot 2 of the mainframe to generate an edge interrupt on either a low-to-high (LH) or a high-to-low (HL) input edge. The program loops until an LH or HL transition on channel 206 generates an SRQ (the transition causes an edge interrupt which generates the SRQ). The interrupt is serviced by controller subroutine Results. When the program completes, the time of the interrupt is displayed.

When the interrupt occurs, the HP-IB SRQ line is set TRUE and an SRQ is sent to the controller. Also, the INTR bit (bit 9) and the service request bit (bit 6) in the status register are set. Because the interrupt is handled by the controller, both bits must be cleared (STA? clears bit 9, SPOLL clears bit 6) before the controller can respond to the next interrupt that occurs.

```
10   ON INTR 7 GOTO Results          !Call sub on interrupt
20   ENABLE INTR 7;2                 !Enable controller intr on SRQ
30   OUTPUT 709;"USE 206"            !Use ch 206
40   OUTPUT 709;"RQS ON;RQS INTR"    !Enable interrupt on SRQ
50   OUTPUT 709;"STA?"               !Clear FPS,LCL,INTR,LMT,ALRM bits
60   OUTPUT 709;"CLROUT"             !Clear STA? data from output buffer
70   OUTPUT 709;"EDGE BOTH"          !Interrupt on either edge transition
80   OUTPUT 709;"ENABLE INTR"        !Enable accessory to interrupt
90   OUTPUT 709;"ENABLE INTR SYS"    !Enable mainframe to sense interrupt
100  GOTO 100                        !Loop until SRQ occurs
110  Results:  !                     !Start controller subroutine
120  OUTPUT 709;"TIME"               !Query time of day
130  ENTER 709;T                     !Enter time of day
140  PRINT "Ch 206 intr @ ";TIME$(T) !Display interrupt time/message
150  A=SPOLL(709)                    !Read/clear SRQ bit
160  STOP                            !End controller subroutine
170  END
```

When the interrupt occurs (an HL or LH edge on channel 206), the controller queries the time of day and enters the time. A typical return is:

```
Ch 206 intr @ 02:46:50
```

• Mainframe

Description

Assigns softkey definitions to front panel keys 0 through 9.

Prerequisites

Requires firmware revision 3.5 or greater.

Syntax

EDIT KEY *key string*

Parameters

key Number of the key (0 through 9) which is defined.

string Softkey definition assigned. The maximum length of the string is 27 characters, not including the quotation marks which enclose the string.

Remarks

Accessing and Executing "Softkey" Commands

The command or message represented by a softkey is accessed when the numeric key is the first key pressed. If any key is pressed before the numeric key, the value of the numeric key is entered. Note that commands or messages represented by softkeys execute, or are displayed immediately. Softkeys are not entered (ENT key).

Softkeys are Allowed in Remote

Any command programmed as a softkey will execute with the instrument in either the Remote or Local operating state. Executing a softkey does not place the HP 3852A in the local state. Softkeys cannot be accessed when the keyboard is locked (LOCK).

Softkey Commands May Change Data Destinations

Commands programmed as softkeys are considered front panel commands when executed. Thus, in the power-on or multitasking mode, executing softkey commands may change the destination of the data returned. For example, if in the power-on mode a subroutine called over the HP-IB is subsequently paused (PAUSE command), continuing (CONT command) the routine using a softkey would cause the data generated by the routine to be returned to the front panel rather than the output buffer.

Similarly, if in the multitasking mode a subroutine executing in a task other than a run task is paused, and then continued with a softkey, the subroutine resumes execution in the **front panel task** with any data being returned to the display. Also, the USE channel is now the default USE channel or the channel local to the front panel task.

Note that when subroutines execute within a run task, data destinations and USE channels do not change (see Examples).

Changing/Deleting Softkey Definitions

Softkey definitions are changed using EDIT KEY and entering a new string for the softkey specified. All softkey definitions are erased anytime the instrument is reset or power is cycled. The SCRATCH KEY command enables you to erase selected softkeys.

Data Returned

None

Related Commands

SCRATCH KEY

Examples

Example: Using Softkeys to Change Program Parameters

In certain applications, the HP 3852A may be located a considerable distance from the controller. The following program demonstrates how softkeys can be used to change program parameters and control program execution in such situations.

The program places the HP 3852A in the multitasking mode, assigns softkey definitions, and downloads two subroutines which will be directed to run tasks. The first subroutine continuously measures the temperature on a preset channel. When the temperature exceeds the specified limit, the date and time are recorded, and the operator is prompted to select another channel and continue the subroutine using softkeys. The second subroutine which aborts the first and releases the controller, is directed to a run task also using a softkey.

```
10    !Select a time-slice period of 65 ms, set the number of run tasks
20    !and queue size, and place the HP 3852A in the multitasking mode.
30    !
40    OUTPUT 709;"TSLICE 0.065"
50    OUTPUT 709;"NTASKS 2,2"
60    OUTPUT 709;"ENABLE MULTI"
70    !
80    !Clear the display except for the user prompt, declare the
90    !temperature limit and measurement channel variables, and define
100   !the softkeys.
110   !
120   OUTPUT 709;"DISABLE LABELS"
130   OUTPUT 709;"INTEGER T,CHAN"
140   OUTPUT 709;"CHAN=2800"
150   OUTPUT 709;"EDIT KEY 0 ""CHAN=2800"""
160   OUTPUT 709;"EDIT KEY 1 ""CHAN=2801"""
170   OUTPUT 709;"EDIT KEY 2 ""CHAN=2802"""
180   OUTPUT 709;"EDIT KEY 3 ""CONT 0"""
190   OUTPUT 709;"EDIT KEY 9 ""RUN 1 STOP"""
200   !
```

```
210   !Monitor the temperature on the selected channel. Report the
220   !date and time when the temperature limit is exceeded. Signal
230   !the user to select the next channel and continue the program
240   !(using the softkeys).
250   !
260   OUTPUT 709;"SUB TESTTEMP"
270   OUTPUT 709;"  T=24"
280   OUTPUT 709;"  REAL A"
290   OUTPUT 709;"  USE 2900"
300   OUTPUT 709;"  CONF THM10K"
310   OUTPUT 709;"  WHILE 1"
320   OUTPUT 709;"    MEAS THM10K CHAN INTO A"
330   OUTPUT 709;"    IF A > T THEN"
340   OUTPUT 709;"      TIMEDATE"
350   OUTPUT 709;"      DISP 'NEXT CHANNEL'"
360   OUTPUT 709;"      PAUSE"
370   OUTPUT 709;"    END IF"
380   OUTPUT 709;"  END WHILE"
390   OUTPUT 709;"SUBEND"
400   !
410   !Direct the subroutine to run task 0.
420   !
430   OUTPUT 709;"RUN 0 TESTTEMP"
440   !
450   !Abort the measurements when softkey "9" is pressed. Send a
460   !service request message to free the controller.
470   !
480   OUTPUT 709;"SUB STOP"
490   OUTPUT 709;"  ABORT 0"
500   OUTPUT 709;"  SRQ"
510   OUTPUT 709;"SUBEND"
520   !
530   !Enter and record the date and time when the limit is exceeded.
540   !
550   OUTPUT 709;"RQS ON"
560   OUTPUT 709;"RQS DAV,FPS"
570   OUTPUT 709;"STA? INTO T"
580   !
590   PRINT "LIMIT EXCEEDED ON:"
600   PRINT
610   ON INTR 7 GOSUB Srq
620   ENABLE INTR 7;2
630 Loop:  GOTO Loop
640   !
650 Srq: IF BIT(SPOLL(709),0) THEN
660        ENTER 709;L
670        PRINT DATE$(L),TIME$(L)
680        PRINT
690        ENABLE INTR 7;2
700      ELSE
710        STOP
720      END IF
730   RETURN
740   END
```

As the program executes, the temperature is monitored on channel 0. When the temperature exceeds 24 degrees C, the HP 3852A's Julian date and time are read, the user is prompted to select another channel, and the subroutine is paused. If, for example, the user presses softkeys "1" and "3", 2801 becomes the value of the CHAN variable used in line 320 and the subroutine is continued. Note that since the subroutine is executing within a run task, the destination of the date and time is still the output buffer even though the program is continued from the front panel.

The operator can end the testing by pressing softkey "9". This directs subroutine STOP to run task 1 which when active, aborts the measurements and releases the controller. A partial output based on this program is shown below:

```
LIMIT EXCEEDED ON:

14 SEP 1987          14:52:07

14 SEP 1987          15:05:12

14 SEP 1987          15:33:18

  .
  .
  .
```

• Mainframe

Description

See **IF...END IF** Command

ENABLE

- **Mainframe**

Description

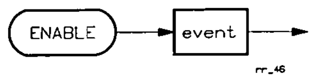
Enables event recognition and servicing.

Prerequisites

None

Syntax

ENABLE *event*



Parameters

event Interrupt or exception (alarm, limit) which causes an interrupt that is responded to by the HP 3852A or by a controller. The five *events are:*

event	Description
INTR SYS	Enables the HP 3852A to recognize an interrupt from a plug-in accessory which can then be responded to by the mainframe or controller. At power-on, INTR SYS is enabled. Front panel CLEAR key, the CLR command, or HP-IB device clear sets DISABLE INTR SYS.
INTR [USE ch]	Enables the accessory channel specified by USE to interrupt. At power-on, all accessory channels and slots are disabled.
LMT	Enables real-time limit exception testing. At power-on, LMT is disabled.
ALRM	Enables the HP 3852A real-time alarm to interrupt the HP 3852A or the controller. At power-on, ALRM is disabled. Front panel CLEAR key, the CLR command, or HP-IB device clear also sets DISABLE ALRM.
LOGCHAN	Enables measurement channel logging. At power-on, LOGCHAN is disabled.

Remarks

Index Pointer Unchanged for ENABLE LOGCHAN

ENABLE LOGCHAN does not alter the present position of the *var* array index pointer (refer to LOGCHAN command). For example, if the index pointer is currently set to element 20, it stays at element 20 after ENABLE LOGCHAN is executed.

Data Returned

None

Related Commands

DISABLE, LMT, SET ALRM, CLR

Examples

Example: Enable Events

The following program lines show how to enable *events* to generate interrupts.

```
              .
              .
 20 OUTPUT 709;"ENABLE INTR SYS"       !Enable HP 3852A to recognize
              .                        !interrupt from accessory channel
              .
 50 OUTPUT 709;"ENABLE INTR,USE 201"   !Enable accessory channel 1
              .                        !in slot 2 to generate interrupt
              .
 80 OUTPUT 709;"ENABLE LMT"            !Enable real-time limit testing
              .
              .
110 OUTPUT 709;"ENABLE ALRM"           !Enable system alarm to
              .                        !interrupt
              .
140 OUTPUT 709;"ENABLE LOGCHAN"        !Enable measurement channel
              .                        !data logging
              .
```

ENABLE EOL SWAP

• **Mainframe**

Description

Enable end of command swapping. When ENABLE EOL SWAP is set, the system swaps to the next scheduled task of equal priority upon completion of the command that is executing when the time-slice period expires. A swap will occur on completion of the last command in a task which may happen before the time-slice period expires.

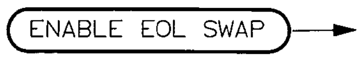
ENABLE EOL SWAP also allows commands within higher priority tasks to execute immediately.

Prerequisites

The HP 3852A must be in the multitasking mode before ENABLE EOL SWAP can be executed. The HP 3852A multitasking capability is only available with firmware revision 3.0 or greater.

Syntax

ENABLE EOL SWAP



R.ENA

Parameters

None

Remarks

Swapping Occurs Between All Tasks of Equal Priority

Generally, in a multitasking system, swapping occurs between the run tasks. Note that any tasks (front panel, HP-IB, interrupt, run tasks) with the same priority (urgency) will swap between each other. Thus, for example, it is possible for the mainframe to swap from a subroutine executing within a run task to a subroutine executing in response to an interrupt.

ENABLE EOL SWAP is set by the ENABLE MULTI Command

Execution of the ENABLE MULTI command also sets ENABLE EOL SWAP. Thus, ENABLE EOL SWAP often does not have to be specified directly when setting up your multitasking system. Generally, you would enter ENABLE EOL SWAP after DISABLE EOL SWAP was used to prevent a swap from occurring between a set of commands that need to execute while the task is currently active. A typical situation would be when a single voltmeter accessory is used for a variety of measurements and configuration of the voltmeter for each set of measurements is critical. Disabling (and later enabling) swapping would prevent a swap from occurring between the configuration phase and measurement phase which, otherwise, could change the configuration and invalidate the measurements.

ENABLE EOL SWAP is Global

ENABLE EOL SWAP is a global setting. This means that when ENABLE EOL SWAP is set, end of command swapping occurs between all tasks of equal priority (urgency), and commands within higher priority tasks (on command completion) are allowed to execute immediately. Tasks cannot be enabled/disabled to swap on an individual basis.

End of Command Swap Time

The time required for the HP 3852A to swap to the next task after command completion is approximately 550 usec.

Data Returned

None

Related Commands

ENABLE MULTI, DISABLE MULTI, DISABLE EOL SWAP, URGENCY, TSLICE, NTASKS, RUN

Examples

Example: Re-enabling End of Command Swapping

This example program shows how ENABLE EOL SWAP is used following DISABLE EOL SWAP to re-enable swapping between tasks.

In the program, the HP 3852A is placed in the multitasking mode and three subroutines are downloaded and directed to run tasks. Since the run tasks have the same priority (set when the HP 3852A enters the multitasking mode), the run tasks will time-slice and swap between each other. One of the run task subroutines performs a series of DC voltage measurements using the voltmeter's rear terminals and multiplexer channels. When the system first swaps to this run task, end of command swapping is disabled so that all of the rear terminal measurements are made at once. End of command swapping is then re-enabled using ENABLE EOL SWAP and measurements involving the multiplexer channels are made while time-slicing with the other run tasks.

```
10    !Set a 65 ms time-slice and set the number of run tasks and
20    !queue size the system will allow. Place the HP 3852A in the
30    !multitasking mode. The reset caused by ENABLE MULTI will load
40    !the TSLICE and NTASKS parameters into the operating system.
50    !
60    OUTPUT 709;"TSLICE .065"
70    OUTPUT 709;"NTASKS 3,3"
80    OUTPUT 709;"ENABLE MULTI"
90    !
100   !Write and download subroutines to be directed to run tasks.
110   !Subroutine A when active, displays the numbers 0 through 599.
120   !Subroutine B when active, repeatedly displays A B C.
130   !
140   OUTPUT 709;"SUB A"
150   OUTPUT 709;"  INTEGER I"
160   OUTPUT 709;"  FOR I=0 TO 599"
```

```
170   OUTPUT 709;"    DISP I"
180   OUTPUT 709;"  NEXT I"
190   OUTPUT 709;"SUBEND"
200   OUTPUT 709;"SUB B"
210   OUTPUT 709;"  INTEGER J"
220   OUTPUT 709;"  FOR J=0 TO 599"
230   OUTPUT 709;"    DISP 'A'"
240   OUTPUT 709;"    DISP 'B'"
250   OUTPUT 709;"    DISP 'C'"
260   OUTPUT 709;"  NEXT J"
270   OUTPUT 709;"SUBEND"
280   !
290   !Write and download a subroutine that will make a series
300   !of DC voltage measurements and store the readings in mainframe
310   !arrays. Use DISABLE EOL SWAP and ENABLE EOL SWAP to enclose a
320   !set of commands so that the first series of measurements
330   !complete before a swap is allowed to occur.
340   !
350   OUTPUT 709;"SUB DCMEAS"
360   OUTPUT 709;"  INTEGER K"
370   OUTPUT 709;"  REAL DCRDGS1 (29)"
380   OUTPUT 709;"  REAL DCRDGS (299)"
390   OUTPUT 709;"  USE 700"
400   OUTPUT 709;"    CONF DCV"
410   OUTPUT 709;"    RANGE 5"
420   OUTPUT 709;"    TERM EXT"
430   !
440   !Prevent a swap from occurring until all 30 measurements are
450   !taken and stored.
460   !
470   OUTPUT 709;"  DISABLE EOL SWAP"
480   OUTPUT 709;"  FOR K = 0 TO 29"
490   OUTPUT 709;"    TRIG SGL"
500   OUTPUT 709;"    CHREAD 700 INTO DCRDGS1"
510   OUTPUT 709;"  NEXT K"
520   !
530   !Enable swapping to occur. This will enable the remaining 300 DC
540   !voltage measurements to be taken while time-slicing with the
550   !other run task subroutines which display 1,2,3 ... and A B C.
560   !
570   OUTPUT 709;"  ENABLE EOL SWAP"
580   OUTPUT 709;"  CONF DCV"
590   OUTPUT 709;"  FOR K=0 TO 99"
600   OUTPUT 709;"    MEAS DCV 600-602 INTO DCRDGS"
610   OUTPUT 709;"  NEXT K"
620   OUTPUT 709;"SUBEND"
630   !
```

```
640   !Direct the subroutines to run tasks. Set each run task to
650   !run one time.
660   !
670   OUTPUT 709;"RUN 1 A"
680   OUTPUT 709;"RUN 2 B"
690   OUTPUT 709;"RUN 3 DCMEAS"
700   END
```

If you look at the display as this program executes, you will notice that when the system swaps to run task 3, CHREAD is displayed for a relatively long period of time. This indicates that end of command swapping was temporarily disabled (line 470) while the measurements were taken on the rear terminals of the voltmeter (lines 480/510). When end of command swapping is re-enabled (line 570), swapping occurs between each of the run tasks as is evidenced by the periodic displays of letters, numbers, and of the MEAS command.

ENABLE INTR

- **HP 44701A Integrating Voltmeter**
- **HP 44702A/B High-Speed Voltmeter (Scanner or System Mode)**

Description

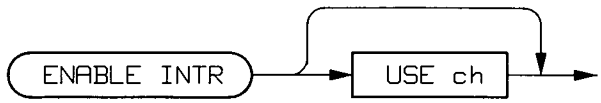
Channel interrupt enable. ENABLE INTR allows the voltmeter to generate an interrupt when a reading becomes available.

Prerequisites

System interrupt capability must also be enabled (with the ENABLE INTR SYS command) before an interrupt can be generated. (At power-on, system interrupt is enabled. However, a device clear disables system interrupt.)

Syntax

ENABLE INTR [USE *ch*]



rr_47

Parameters

USE *ch* Voltmeter slot number. See Glossary.

Remarks

Interrupts Automatically Cleared and Disabled

When an interrupt occurs on a channel which has been enabled to interrupt, the interrupt is disabled when it is serviced and is cleared when data is read from the voltmeter.

Interrupts Handled by the HP 3852A

To respond to an interrupt from an HP 44701A or HP 44702A/B voltmeter with the HP 3852A mainframe, the following conditions must be set:

- Specify the voltmeter that is to interrupt with the USE command.
- Enable the HP 3852A to respond to the interrupt by setting up a subroutine and sending the ON INTR CALL *name* command, where *name* is the name of the HP 3852A subroutine.
- Configure the voltmeter to interrupt on the desired condition.
- Enable the voltmeter to interrupt with the ENABLE INTR command.
- Enable the HP 3852A to sense the interrupt from the voltmeter with the ENABLE INTR SYS command.

Interrupts Handled by the Controller

To respond to an interrupt from an HP 44701A or HP 44702A/B voltmeter with a controller, the following conditions must be set:

- Provide and enable interrupt routines within the controller.
- Clear Status Register bit 9 (INTR) with the STA? command.
- Enable HP 3852A SRQ capability (set HP-IB SRQ line TRUE when an interrupt signal occurs) with the RQS ON command.
- Unmask the INTR bit in the Status Register with RQS INTR or RQS 512.
- Specify the voltmeter that is to interrupt with the USE command.
- Configure the voltmeter to interrupt on the desired condition.
- Enable the voltmeter to interrupt with the ENABLE INTR command.
- Enable the HP 3852A to sense the interrupt from the voltmeter with the ENABLE INTR SYS command.
- Clear the service request bit (bit 6 in the status register) using STB? or the HP Series 200/300 SPOLL command so that the controller can respond to multiple interrupts on the channel.

Data Returned

None

Related Commands

DISABLE INTR, INTR?, USE, ENABLE

Examples

Example: Enable Voltmeter Interrupt, Handle with HP 3852A

This program enables an HP 44701A voltmeter in slot 1 of the mainframe to interrupt when a voltage reading is available. The measurement is made using the input terminals on the rear panel of the voltmeter. The interrupt is handled by the HP 3852A. When the subroutine is called, DATA READY is displayed for three seconds then the measurement is displayed.

```
10 OUTPUT 709;"RST"                   !Reset the HP 3852A and HP 44701A
20 OUTPUT 709;"USE 100"               !Use voltmeter in slot 1
30 OUTPUT 709;"SUB Data"              !Define subroutine Data
40 OUTPUT 709;"DISP 'DATA READY'"     !Display msg on front panel
50 OUTPUT 709;"WAIT 3"                !Display message for 3 seconds
60 OUTPUT 709;"CHREAD 100"            !Display measurement on mainframe
70 OUTPUT 709;"SUBEND"                !End subroutine
80 OUTPUT 709;"ON INTR CALL Data"     !Call Data on data available
90 OUTPUT 709;"CONF DCV"              !Configure for DC volts
100 OUTPUT 709;"TERM EXT"             !Set ext. terms as the only input
110 OUTPUT 709;"TRG EXT"              !System trigger source is SYSTEM
```

```
120                                 !TRIGGER IN BNC
130 OUTPUT 709;"ENABLE INTR"        !Enable voltmeter to interrupt
140 OUTPUT 709;"ENABLE INTR SYS"    !Sense the backplane interrupt
150 OUTPUT 709;"TRIG SYS"           !Vm trigger source is system trig
160 END
```

When this program runs, the subroutine "Data" is downloaded into the HP 3852A. The voltmeter is configured for the measurement and enabled to interrupt, then the mainframe is enabled to sense the interrupt. Once the reading is taken, the interrupt occurs.

The voltmeter's interrupt capability is disabled when the interrupt is serviced. The interrupt is cleared when the measurement is read from the voltmeter (CHREAD).

Example: Enable Voltmeter Interrupt, Handle with Controller

This program enables an HP 44702A voltmeter in slot 5 of the mainframe to interrupt when a scan of the specified channels completes. When the interrupt occurs, the HP-IB SRQ line is set TRUE and the interrupt is sent to the controller. The interrupt is handled by the controller subroutine Results. When the program completes, "SCAN IS COMPLETE" is displayed on the controller CRT then the readings are transferred to the controller.

```
 10 ON INTR 7 GOSUB Results          !Call sub on interrupt
 20 ENABLE INTR 7;2                  !Enable controller intr capability
 30 DIM Rdgs(0:9)                    !Dimension controller array
 40 OUTPUT 709;"RST"                 !Reset the HP 3852A and HP 44702A/B
 50 OUTPUT 709;"RQS ON"              !Set RQS mode on
 60 OUTPUT 709;"RQS INTR"            !Unmask the INTR bit (bit 9)
 70 OUTPUT 709;"STA?"                !Ensure status register bit 9 is cleared
 80 OUTPUT 709;"CLROUT"              !Clear the output buffer
 90 OUTPUT 709;"USE 500"             !Use the voltmeter in slot 5
100 OUTPUT 709;"SCANMODE ON"         !Turn scanner mode on
110 OUTPUT 709;"CONF DCV"            !Configure for DC voltage measurements
120 OUTPUT 709;"CLWRITE SENSE 400-409"
130                                  !Specify ch list and ribbon cableconn.
140 OUTPUT 709;"RDGSMODE END"        !Interrupt when scan completes
150 OUTPUT 709;"ENABLE INTR"         !Enable voltmeter to interrupt
160 OUTPUT 709;"ENABLE INTR SYS"     !Enable mainframe to sense interrupt
170 OUTPUT 709;"SCTRIG EXTO"         !Externally trigger voltmeter
180 GOTO 180                         !Loop until interrupt occurs
190 Results:  !                      !Start controller subroutine
200 PRINT "SCAN IS COMPLETE"         !Display message
210 A=SPOLL(709)                     !Clear HP 3852 service request bit
220 OUTPUT 709;"XRDGS 500"           !Transfer readings to output buffer
230 ENTER 709;Rdgs(*)                !Enter readings into controller
240 END
```

When the interrupt occurs, the INTR bit (bit 9) and the service request bit (bit 6) in the status register are set. Because the interrupt is handled by the controller, both bits must be cleared (STA? clears bit 9, SPOLL clears bit 6) before the controller can respond to the next channel interrupt that occurs.

• HP 44714A 3-Channel Stepper Motor Controller/Pulse Output

Description

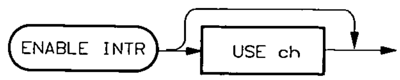
Enables the interrupt that occurs for the channel specified when a move (pulse train) is completed, when a limit is reached and pulses stop abruptly, or when a halt is executed and pulses stop abruptly.

Prerequisites

Requires mainframe firmware revision 3.0 or greater.

Syntax

ENABLE INTR [USE *ch*]



Parameters

USE *ch* — Channel which is enabled to interrupt. Channel number range can be ES00 through ES02.

Remarks

Interrupts Generated

A channel that has the interrupt enabled generates an interrupt after pulses initiated by a MOVE or SUSTAIN command stop (by reaching a limit, by a halt command, or by completing a move).

Interrupts Automatically Cleared

An interrupt is enabled and is automatically cleared when the ENABLE INTR command is executed.

Handling Interrupts With the HP 3852A

For the HP 3852A to respond to an interrupt from an HP 44714A, the following conditions must be set:

• Enable the HP 3852A to handle the interrupt by setting up a subroutine and using the ON INTR CALL *name* command (*name* is the name of the interrupt servicing subroutine).

• Specify the HP 44714A channel that is to interrupt with the USE command.

• Enable the HP 44714A channel with the ENABLE INTR command.

• Enable the HP 3852A to sense the interrupt from the HP 44714A with the ENABLE INTR SYS command.

Handling Interrupts With a Controller

To respond to an interrupt from an HP 44714A using a controller, the following conditions must be set:

- Provide and enable intrrupt routines within the controller.
- Clear Status Register bit 9 (INTR) with the STA? command.
- Enable HP 3852A SRQ capability (set HP-IB SRQ line TRUE when an interrupt signal occurs) with the RQS ON command.
- Unmask the INTR bit in the Status Register with RQS INTR or RQS 512.
- Specify the HP 44714A channel to interrupt with the USE command.
- Enable the HP 44714A to interrupt with the ENABLE INTR command.
- Enable the HP 3852A to sense the interrupt from the HP 44714A with the ENABLE INTR SYS command.
- Clear the service request bit (bit 6 in the status register) using STB? or the HP series 200/300 SPOLL command so that the controller can respond to multiple interrupts on the channel.

Servicing Multiple Interrupts

If more than one channel generates an interrupt, the HP 3852A services the lowest-numbered channel first, then the next lowest-numbered channel, etc.

Data Returned

None

Related Commands

DISABLE INTR, INTR?, USE, ENABLE INTR SYS

Examples

Example: Enable Channel Interrupt, Handle with HP 3852A

```
 10 OUTPUT 709;"RST"                          !Reset HP 3852 and HP 44714A
 20 OUTPUT 709;"USE 502"                      !Use channel 2 slot 5
 30 OUTPUT 709;"STANDBY AUTO LO"              !Set standby output
 40 OUTPUT 709;"PULSE SS HI HI"               !Set pulse mode
 50 OUTPUT 709;"PROFILE FREQ 100 900 1E3 25E-6" !Set motion profile
 60 OUTPUT 709;"SUB Comp"                     !Define subroutine Comp
 70 OUTPUT 709;"DISP 'PULSES COMPLETE'"       !Display msg (front panel)
 80 OUTPUT 709;"SUBEND"                       !End subroutine
 90 OUTPUT 709;"ON INTR CALL Comp"            !Call Comp on interrupt
100 OUTPUT 709;"ENABLE INTR"                  !Enable ch to interrupt
110 OUTPUT 709;"ENABLE INTR SYS"              !Sense backplane interrupt
120 OUTPUT 709;"MOVE 8E3 NOWAIT"              !Move 8000 then interrupt
130 END
```

Channel interrupt capability is disabled when the interrupt is serviced. The interrupt is cleared when ENABLE INTR is executed and is active after MOVE completes.

Example: Enable Channel Interrupt, Handled by Controller

```
This program enables channel 2 of an HP 44714A in slot 5 of the
mainframe to interrupt the controller each time a pulse train is generated
and completes.

When the interrupt occurs, the HP-IB SRQ line is set TRUE and the interrupt
is sent to the controller.  When the program completes, "Pulses Complete" is
displayed on the controller CRT.

 10 ON INTR 7 GOTO Finished                    !Define event initiated branch
 20 ENABLE INTR 7;2                            !Enable controller intr capability
 30 OUTPUT 709;"RST"                           !Reset HP 3852A and HP 44714A
 40 OUTPUT 709;"STA?"                          !Clear INTR bit in status register
 50 OUTPUT 709;"RQS ON"                        !Set RQS mode ON
 60 OUTPUT 709;"RQS INTR"                      !Unmask INTR bit in status reg
 70 OUTPUT 709;"USE 502"                               !Use channel 2 in slot 5
 80 OUTPUT 709;"STANDBY AUTO LO"                       !Set the standby output
 90 OUTPUT 709;"PULSE SD HI HI"                        !Set the pulse mode
100 OUTPUT 709;"PROFILE FREQ 100 800 1E3 25E-6"        !Set a motion profile
110 OUTPUT 709;"ENABLE INTR"                           !Enable channel to interrupt
120 OUTPUT 709;"ENABLE INTR SYS"                   !Sense the backplane interrupt
130 OUTPUT 709;"MOVE 8E3 NOWAIT"                   !Move 8000 then interrupt
140 GOTO 140                                       !Loop until interrupt occurs
150 Finished: !                                    !Line label for branch
160 PRINT "Pulses completed on ch 502"             !Display message
170 A=SPOLL(709)                                   !Clear service request bit
180 END
```

When the HP 44714A channel completes a pulse train, the INTR bit (bit 9) and the service request bit (bit 6) in the status register are set. Because the interrupt is handled by the controller, both bits must be cleared (STA? clears bit 9, SPOLL clears bit 6) before the controller can respond to the next channel interrupt that occurs.

ENABLE INTR

- **HP 44715A 5-Channel Counter/Totalizer**

Description

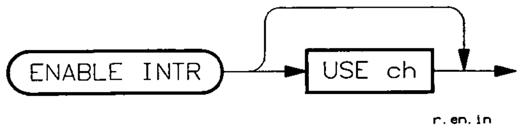
Enable specified counter channel to interrupt on an overflow or when a reading is available.

Prerequisites

None

Syntax

ENABLE INTR [USE *ch*]



Parameters

USE *ch* Channel which is enabled to interrupt. Channel number range (depends on HP 44715A hardware configuration) = ES00 through ES04.

Remarks

Interrupts by Channel Function

Channels set for TOTAL or TOTALM functions will interrupt on counter overflow. Channels set for RAT, PER, PERD, or Frequency function will interrupt when the reading is available. (Interrupts are not available for the UDC, UDCM, CD, or CDM functions.)

Interrupts Automatically Cleared and Disabled

If a channel is set for the TOTAL or TOTALM function, ENABLE INTR enables the channel for interrupts and clears any previous interrupt.

If a channel is set for the RAT, PER, or PERD functions or for Frequency configuration, the ENABLE INTR command will not clear a pending interrupt; therefore, executing ENABLE INTR could cause an interrupt to occur immediately. For these functions, use the CHREAD, CHREADZ, or XRDGS command to clear a pending interrupt.

If the function (range, edge, etc.) of an enabled channel is changed, the channel remains enabled but any pending interrupt is cleared.

Interrupts Handled by the HP 3852A

To respond to an interrupt from an HP 44715A counter using the HP 3852A, the following conditions must be set:

- Specify the counter channel that is to interrupt with the USE command.

- Enable the HP 3852A to handle the interrupt by setting up a subroutine and using the ON INTR CALL *name* command, where *name* is the name of the subroutine.
- Configure the counter to interrupt on the desired condition.
- Enable the counter to interrupt (on overflow or reading available) with the ENABLE INTR command.
- Enable the HP 3852A to sense the interrupt from the counter with the ENABLE INTR SYS command.

Interrupts Handled by the Controller

To respond to an interrupt from an HP 44715A counter using a controller, the following conditions must be set:

- Provide and enable interrupt routines within the controller.
- Clear Status Register bit 9 (INTR) with the STA? command.
- Enable HP 3852A SRQ capability (set HP-IB SRQ line TRUE when an interrupt signal occurs) with the RQS ON command.
- Unmask the INTR bit in the Status Register with RQS INTR or RQS 512.
- Specify the counter channel that is to interrupt with the USE command.
- Configure the counter to interrupt on the desired condition.
- Enable the counter to interrupt (on overflow or reading available) with the ENABLE INTR command.
- Enable the HP 3852A to sense the interrupt from the counter with the ENABLE INTR SYS command.
- Clear the service request bit (bit 6 in the status register) using STB? or the HP Series 200/300 SPOLL command so that the controller can respond to multiple interrupts on the channel.

Data Returned

None

Related Commands

DISABLE INTR, INTR?, USE, ENABLE INTR SYS

Examples

Example: Enable Channel Interrupt, Handle with HP 3852A

This program enables channel 3 of an HP 44715A counter in slot 2 of the mainframe to interrupt when the channel counter overflows (from -1 to 0). The counter is preset to overflow after 10 counts.

The interrupt is handled by the HP 3852A. When the interrupt occurs, the program calls subroutine Data to respond to the interrupt. When the program completes, SUBEND is displayed in the left display window and CTR OVERFLOW in the right display window.

```
10 OUTPUT 709;"RST"                        !Reset HP 3852A & HP 44715A
20 OUTPUT 709;"USE 203"                    !Use channel 3 in slot 2
30 OUTPUT 709;"SUB Data"                   !Define subroutine Data
40 OUTPUT 709;"DISP 'CTR OVERFLOW'"        !Display msg on front panel
50 OUTPUT 709;"SUBEND"                     !End subroutine
60 OUTPUT 709;"ON INTR CALL Data"          !Call Data on ctr overflow
70 OUTPUT 709;"EDGE LH"                    !Count positive edges
80 OUTPUT 709;"FUNC TOTAL"                 !Set ch for TOTAL function
90 OUTPUT 709;"CNTSET -10"                 !Overflow after 10 pos edges
100 OUTPUT 709;"ENABLE INTR"               !Enable channel to intr
110 OUTPUT 709;"ENABLE INTR SYS"           !Sense the backplane interrupt
120 OUTPUT 709;"TRIG SGL"                  !Trigger once
130 END
```

Channel interrupt capability is disabled when the interrupt is serviced. Also, because of the the counter function (TOTAL), the interrupt is also cleared during servicing.

Example: Enable Channel Interrupt, Handled by Controller

This program enables channel 3 of an HP 44715A counter in slot 2 of the mainframe to interrupt the controller when the counter channel overflows (from -1 to 0). The counter is preset to overflow after 10 counts.

When the interrupt occurs, the HP-IB SRQ line is set TRUE and the interrupt is sent to the controller. When the program completes, "Overflow on ch 203" is displayed on the controller CRT.

```
10 ON INTR 7 GOTO Results                  !Define event initiated branch
20 ENABLE INTR 7;2                         !Enable controller intr capability
30 OUTPUT 709;"RST"                        !Reset HP 3852A and HP 44715A
40 OUTPUT 709;"STA?"                       !Clear INTR bit in status register
50 OUTPUT 709;"RQS ON"                     !Set RQS mode ON
60 OUTPUT 709;"RQS INTR"                   !Unmask INTR bit in status register
70 OUTPUT 709;"USE 203"                    !Use channel 3 in slot 2
80 OUTPUT 709;"EDGE LH"                    !Count positive edges
90 OUTPUT 709;"FUNC TOTAL"                 !Set channel for TOTAL function
100 OUTPUT 709;"CNTSET -10"                !Overflow after 10 pos edges
110 OUTPUT 709;"ENABLE INTR"               !Enable channel to interrupt
120 OUTPUT 709;"ENABLE INTR SYS"           !Sense the backplane interrupt
130 OUTPUT 709;"TRIG SGL"                  !Trigger once
140 GOTO 140                               !Loop until interrupt occurs
150 Results: !                             !Line label for branch
160 PRINT "Overflow on ch 203"             !Display message
170 A=SPOLL(709)                           !Clear HP 3852 service request bit
180 END
```

When the counter overflows, the INTR bit (bit 9) and the service request bit (bit 6) in the status register are set. Because the interrupt is handled by the controller, both bits must be cleared (STA? clears bit 9, SPOLL clears bit 6) before the controller can respond to the next channel interrupt that occurs.

- **HP 44721A 16-Channel Digital Input**
- **HP 44722A 8-Channel Digital Input**

Description

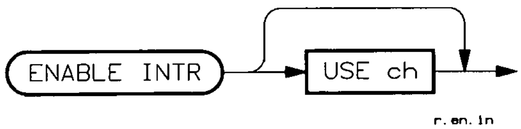
Enables the specified digital input channel(s) to interrupt on counter overflow or when the specified edge transition occurs.

Prerequisites

ENABLE INTR USE ES90 or ES91 requires mainframe firmware revision 3.0 or greater.

Syntax

ENABLE INTR [USE *ch*]



Parameters

USE *ch* Specifies channel(s) to be enabled for counter or event interrupts.

USE ch		Description
ES00-ES15	[1]	Enables channel specified by USE ch to interrupt on counter overflow (counter interrupt).
ES16-ES31	[2]	Enables channel specified by USE ch to interrupt when the edge specified by EDGE occurs (event interrupt).
ES90	[3]	Enables event interrupt to be generated when the edge(s) specified by EDGE occurs on <u>any</u> channel in range ES16-ES31 (ES08-ES15 for the HP 44722A).
ES91	[3]	Enables counter interrupt to be generated when counter overflow occurs on <u>any</u> channel in range ES00-ES15 (ES00-ES07 for the HP 44722A).

[1] = ES00-ES07 for the HP 44722A.
[2] = ES08-ES15 for the HP 44722A.
[3] = Valid only for mainframe firmware revision 3.0 or greater.

Remarks

Interrupts Automatically Cleared and Disabled

For an enabled channel, when an interrupt (event or counter) occurs, the interrupt is automatically cleared and disabled when the interrupt is serviced.

Channel Can be Enabled for Both Event and Counter Interrupts

A digital input channel can be simultaneously enabled for both event interrupts and counter interrupts. For example, if EDGE LH is set for channel 203, ENABLE INTR, USE 203 enables channel 203 to interrupt when the channel counter overflows (from -1 to 0). Or, ENABLE INTR, USE 219 (ENABLE INTR, USE 211 for 8-channel) enables channel 203 to interrupt on the first positive edge.

EDGE Command Sets Edges for Both Event and Counter Interrupts

The EDGE command sets the edges (positive or negative) for BOTH event and counter interrupts on a channel. Thus, for example, you can't set positive edges on a channel for counter interrupts and negative edges for event interrupts. (See the EDGE command description).

Servicing Multiple Interrupts

If an enabled channel generates an event interrupt and a counter interrupt, the HP 3852A services the counter interrupt first, then the event interrupt. If more than one channel generates an interrupt, the HP 3852A services the lowest-numbered channel first, then the next lowest-numbered channel, and so on. The digital input keeps track of the interrupts which have been serviced.

Interrupts Handled by the HP 3852A

To respond to an interrupt from an HP 44721A or HP 44722A digital input with the HP 3852A mainframe, the following conditions must be set:

- Specify the digital input channel to interrupt with the USE command.
- Enable the HP 3852A to handle the interrupt by setting up a subroutine and sending the ON INTR CALL *name* command, where *name* is the name of the HP 3852A subroutine.
- Configure the digital input to interrupt on the desired condition.
- Enable the digital input channel to interrupt (on edge sensed or on counter overflow) with the ENABLE INTR command.
- Enable the HP 3852A to sense the interrupt from the digital input with the ENABLE INTR SYS command.

Interrupts Handled by the Controller

To respond to an interrupt from an HP 44721A or HP 44722A digital input with the controller, the following conditions must be set:

- Provide and enable interrupt routines within the controller.
- Clear Status Register bit 9 (INTR) with the STA? command.
- Enable HP 3852A SRQ capability (set HP-IB SRQ line TRUE when an inter-

rupt signal occurs) with the RQS ON command.

- Unmask the INTR bit in the Status Register with RQS INTR or RQS 512.
- Specify the digital input channel to interrupt with the USE command.
- Configure the digital input to interrupt on the desired condition.
- Enable the digital input channel to interrupt (on edge detected or on counter overflow) with the ENABLE INTR command.
- Enable the HP 3852A to sense the interrupt from the digital input with the ENABLE INTR SYS command.
- Clear the service request bit (bit 6 in the status register) using STB? or the HP Series 200/300 SPOLL command so that the controller can respond to multiple interrupts on the channel.

Data Returned

None

Related Commands

DISABLE INTR, INTR?, USE, ENABLE INTR SYS

Examples

Example: Enable Event Interrupt, Handle with HP 3852A

This program enables channel 3 of an HP 44721A 16-channel digital input in slot 2 of the mainframe to interrupt when the first positive edge is sensed. The interrupt is handled by the HP 3852A.

Channel interrupt capability is disabled when the interrupt is serviced. For the digital input accessories, servicing also clears the interrupt.

When the interrupt occurs, the program calls subroutine Data to respond to the interrupt. When the program completes, SUBEND is displayed in the left display window and INTR-CH 203 is displayed in the right display window.

```
 10 OUTPUT 709;"RST"                    !Reset the HP 3852A and HP 44721A
 20 OUTPUT 709;"USE 219"                !Use ch 3 in slot 2 (event intr)
 30 OUTPUT 709;"SUB Dta"                !Define subroutine Dta
 40 OUTPUT 709;"DISP 'INTR-CH 203'"     !Display msg on front panel
 50 OUTPUT 709;"SUBEND"                 !End subroutine
 60 OUTPUT 709;"ON INTR CALL Dta"       !Call Dta on first pos edge
 70 OUTPUT 709;"EDGE LH"                !Detect Positive Edge
 80 OUTPUT 709;"ENABLE INTR"            !Enable channel to intr
 90 OUTPUT 709;"ENABLE INTR SYS"        !Sense the backplane interrupt
100 END
```

Example: Enable Counter Interrupt, Handle with Controller

This program enables channel 3 of an HP 44721A 16-channel digital input in slot 2 of the mainframe to interrupt when the channel counter overflows (from -1 to 0). The digital input channel is preset to overflow after 10 counts.

When the interrupt occurs, the HP-IB SRQ line is set TRUE and the interrupt is sent to the controller. When the program completes, "Overflow on ch 203" is displayed on the controller CRT.

When the digital input channel counter overflows, the INTR bit (bit 9) and the service request bit (bit 6) in the status register are set. Because the interrupt is handled by the controller, both bits must be cleared (STA? clears bit 9, SPOLL clears bit 6) before the controller can respond to the next channel interrupt that occurs.

```
 10 ON INTR 7 GOTO Results           !Define event-initiated branch
 20 ENABLE INTR 7;2                  !Enable interrupt capability
 30 OUTPUT 709;"RST"                 !Reset the HP 3852A and HP 44721A
 40 OUTPUT 709;"STA?"                !Clear INTR bit in status register
 50 OUTPUT 709;"RQS ON"              !Set RQS mode ON
 60 OUTPUT 709;"RQS INTR"            !Unmask INTR bit in status register
 70 OUTPUT 709;"USE 203"             !Use channel 3 in slot 2
 80 OUTPUT 709;"EDGE LH"             !Count positive edges
 90 OUTPUT 709;"CNTSET -10"          !Overflow after 10 pos edges
100 OUTPUT 709;"ENABLE INTR"         !Enable channel to interrupt
110 OUTPUT 709;"ENABLE INTR SYS"     !Sense the backplane interrupt
120 GOTO 120                         !Loop until intr occurs
130 Results: !                       !Line label for branch
140 PRINT "Overflow on ch 203"       !Display message
150 A=SPOLL(709)                     !Clear HP 3852 service request bit
160 END
```

ENABLE INTR

• **HP 44723A 16-Channel High-Speed Digital Sense/Control**

Description

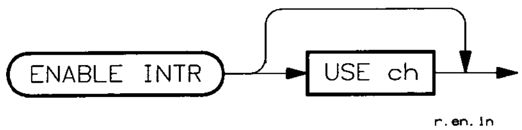
Enables interrupts for the specified channel. Types of interrupts enabled are edge (channel range ES00-ES15 and ES93), pattern (ES90), input (ES91), and output (ES92).

Prerequisites

Requires mainframe firmware revision 3.0 or greater.

Syntax

ENABLE INTR [USE *ch*]



Parameters

USE *ch* Sets the applicable channel/function for interrupts to be enabled. At power-on or following a RST or RST *slot* command, all interrupts are disabled. All interrupts are also cleared except output interrupts which are set.

USE ch	Type	Description
ES00-ES15	Edge	When enabled, edge interrupts occur when the edge programmed by EDGE is seen at the channel input.
ES90	Pattern	When enabled, pattern interrupts occur when user inputs match the pattern/ mask condition set by PATTERN.
ES91	Input	When enabled, input interrupts occur on a first rank input trigger and are cleared by a second rank input trigger.
ES92	Output	When enabled, output interrupts occur on a second rank output trigger and are cleared by a write to the first rank output register (with CHWRITE, CHWRITEM, WRITE, or WRITEM).
ES93	Edge	When enabled, edge interrupts occur when the edge programmed by EDGE is seen at any input channel.

Remarks

Edge Interrupts

Edge interrupts are generated from user input data. Any pending edge interrupt on the channel is cleared by ENABLE INTR. Edge interrupts are cleared and disabled when serviced by the mainframe.

Pattern Interrupts

Enabling pattern interrupts will generate an immediate interrupt if the pattern/mask criteria set by PATTERN is currently met. Pattern interrupts are cleared and disabled when serviced by the mainframe.

Input Interrupts

Input interrupts (ES91) cannot be enabled if SRTRIG EXT or SRTRIG SYS is set for the second rank input register, and vice-versa. Enabling input interrupts results in an immediate interrupt if a first rank input trigger has been received without a corresponding second rank input trigger.

Output Interrupts

Enabling output interrupts (ES92) results in an immediate output trigger: (a) after power-on; (b) when a RST or RST *slot* command is executed; or (c) if the last data written to the first rank output register (with CHWRITE, CHWRITEM, WRITE, or WRITEM) has been transferred to the second rank output register by a second rank output trigger.

Data Returned

None

Related Commands

DISABLE INTR, EDGE, PATTERN

Examples

Example: Enable Input Interrupt

This program enables an HP 44723A in slot 2 of the mainframe to generate an input interrupt when an external (HL) trigger is input to the first rank input trigger port on the terminal module. The program loops until the trigger occurs and an SRQ is generated (the trigger causes an input interrupt which generates an SRQ). The interrupt is serviced by controller subroutine Results. When the program completes, the time of the interrupt is displayed.

When the interrupt occurs, the HP-IB SRQ line is set TRUE and an SRQ is sent to the controller. Also, the INTR bit (bit 9) and the service request bit (bit 6) in the status register are set. Because the interrupt is handled by the controller, both bits must be cleared (STA? clears bit 9, SPOLL clears bit 6) before the controller can respond to the next interrupt that occurs. Also, since SRTRIG INT is set, READ (line 140) is used to clear the input interrupt.

```
10   ON INTR 7 GOTO Results            !Call sub on interrupt
20   ENABLE INTR 7;2                   !Enable controller intr on SRQ
30   OUTPUT 709;"RQS ON;RQS INTR"      !Enable interrupt on SRQ
40   OUTPUT 709;"STA?"                 !Clear FPS,LCL,INTR,LMT,ALRM bits
50   OUTPUT 709;"CLROUT"               !Clear STA? data from output buffer
60   OUTPUT 709;"TRIG EXT USE 200"     !Intr on ext input trigger
70   OUTPUT 709;"ENABLE INTR USE 291"  !Enable input interrupt
80   OUTPUT 709;"ENABLE INTR SYS"      !Enable mainframe to sense interrupt
90   GOTO 90                           !Loop until interrupt occurs
100  Results: !                        !Start controller subroutine
110  OUTPUT 709;"TIME"                 !Query time of day
120  ENTER 709;T                       !Enter time of day
130  PRINT "Slot 2 intr @ ";TIME$(T)   !Display interrupt time/message
140  OUTPUT 709;"READ 200"             !Clear input intr (SRTRIG INT set)
150  A=SPOLL(709)                      !Read/clear SRQ bit
160  STOP                              !End controller subroutine
170  END
```

When the input interrupt occurs (an HL trigger to the first rank input trigger port), the controller queries the time of day and enters the time. A typical return is:

```
Slot 2 intr @ 02:46:50
```

Example: Enable Output Interrupt

This program enables an HP 44723A in slot 2 of the mainframe to generate an output interrupt when an external (HL) trigger is input to the second rank output trigger port on the terminal module. The program loops until an external trigger generates an SRQ (the trigger causes an output interrupt which generates the SRQ). The interrupt is serviced by controller subroutine Results. When the program completes, the time of the interrupt is displayed. When the interrupt occurs, "65" is written to the first rank output register.

When the interrupt occurs, the HP-IB SRQ line is set TRUE and an SRQ is sent to the controller. Also, the INTR bit (bit 9) and the service request bit (bit 6) in the status register are set. Because the interrupt is handled by the controller, both bits must be cleared (STA? clears bit 9, SPOLL clears bit 6) before the controller can respond to the next interrupt that occurs. Note that a WRITE command (line 70) is used to prevent an immediate output interrupt.

```
10   ON INTR 7 GOTO Results            !Call sub on interrupt
20   ENABLE INTR 7;2                   !Enable controller intr on SRQ
30   OUTPUT 709;"RQS ON;RQS INTR"      !Enable interrupt on SRQ
40   OUTPUT 709;"STA?"                 !Clear FPS,LCL,INTR,LMT,ALRM bits
50   OUTPUT 709;"CLROUT"               !Clear STA? data from output buffer
60   OUTPUT 709;"SRTRIG EXT USE 216"   !Intr on ext input trigger
70   OUTPUT 709;"WRITE 200,0"          !Write to first rank output register
80   OUTPUT 709;"ENABLE INTR USE 292"  !Enable output interrupt
90   OUTPUT 709;"ENABLE INTR SYS"      !Enable mainframe to sense interrupt
100  GOTO 100                          !Loop until interrupt occurs
110  Results: !                        !Start controller subroutine
120  OUTPUT 709;"TIME"                 !Query time of day
```

```
130  ENTER 709;T                      !Enter time of day
140  PRINT "Slot 2 intr @ ";TIME$(T)  !Display interrupt time/message
150  OUTPUT 709;"WRITE 200,65"        !Write new data to first rank out reg
160  A=SPOLL(709)                     !Read/clear SRQ bit
170  STOP                             !End controller subroutine
180  END
```

When the output interrupt occurs (an HL trigger to the second rank output trigger port), the controller queries the time of day and enters the time. A typical return is:

```
Slot 2 intr @ 02:46:50
```

• HP 44726A 2-Channel Arbitrary Waveform DAC

Description

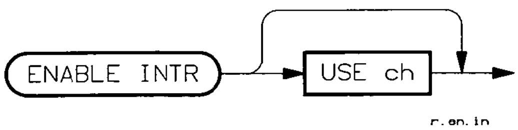
Channel interrupt enable. ENABLE INTR allows the DAC accessory to generate an interrupt after NSCAN cycles of the waveform have occurred. Executing ENABLE INTR will also clear an interrupt which has not been serviced.

Prerequisites

System interrupt capability must also be enabled (ENABLE INTR SYS) before an interrupt will be recognized. This command when used with the HP 44726A requires firmware revision 3.5 or greater.

Syntax

ENABLE INTR [USE *ch*]



Parameters

USE *ch* Channel which will interrupt after NSCAN cycles have occurred. The range for *ch* is ES00 to ES01. The default USE *ch* is channel 0.

Remarks

HP 44726A Interrupts Automatically Cleared and Disabled

When a channel has been enabled and an interrupt occurs, the interrupt is cleared and the channel set to DISABLE INTR when the interrupt is serviced.

Interrupts Handled by the HP 3852A

To respond to an interrupt from the HP 44726A with the HP 3852A mainframe, the following conditions must be set:

- Enable the HP 3852A to respond to the interrupt by setting up a subroutine and sending the ON INTR CALL *name* command, where *name* is the name of the HP 3852A subroutine.
- Define and store the appropriate waveform and set the number of cycles to occur before the interrupt is generated.
- Enable the HP 44726A to interrupt with the ENABLE INTR command.
- Enable the HP 3852A to sense the interrupt from the arbitrary waveform DAC with the ENABLE INTR SYS command.

Interrupts Handled by the Controller

To respond to an interrupt from the HP 44726A with a controller, the following conditions must be set:

- Provide and enable interrupt routines within the controller.
- Clear Status Register bit 9 (INTR) with the STA? command.
- Enable HP 3852A SRQ capability (set HP-IB SRQ line TRUE when an interrupt signal occurs) with the RQS ON command.
- Unmask the INTR bit in the Status Register with RQS INTR or RQS 512.
- Define and store the appropriate waveform and set the number of cycles to occur before the interrupt is generated.
- Enable the HP 44726A to interrupt with the ENABLE INTR command.
- Enable the HP 3852A to sense the interrupt from the arbitrary waveform DAC with the ENABLE INTR SYS command.
- Clear the service request bit (bit 6 in the status register) using STB? or the HP Series 200/300 SPOLL command so that the controller can respond to multiple interrupts on the channel.

Data Returned

None

Related Commands

DISABLE INTR, NSCAN

Examples

Example 1: Enable HP 44726A Interrupt, Handle with HP 3852A

This program enables the HP 44726A in slot 0 of the mainframe to interrupt after 10 cycles of an arbitrary waveform have completed. The waveform used is a signal that has been digitized by the HP 44702A. When the cycles complete and the interrupt occurs, CYCLES COMPLETE is displayed on the mainframe.

```
10    !Reset the mainframe and the HP 44726A. Download, and enable a
20    !subroutine to be called when NSCAN cycles of the waveform have
30    !completed.
40    !
50    OUTPUT 709;"RST"
60    WAIT 1
70    OUTPUT 709;"SUB WV_FORM"
80    OUTPUT 709;"  DISP 'CYCLES COMPLETE'"
90    OUTPUT 709;"SUBEND"
100   OUTPUT 709;"ON INTR CALL WV_FORM"
110   !
120   !Configure the HP 44702A voltmeter to take 10,000 readings 10 us
130   !apart. Transfer those readings to an array which will be
```

```
140  !specified as the amplitude array in the WFWRITE ARB command.
150  !
160  OUTPUT 709;"REAL AMPLT(9999)"
170  OUTPUT 709;"USE 500"
180  OUTPUT 709;"  CONF DCV"
190  OUTPUT 709;"  TERM EXT"
200  OUTPUT 709;"  NRDGS 10000"
210  OUTPUT 709;"  DELAY 0,10E-6"
220  OUTPUT 709;"  TRIG SGL"
230  OUTPUT 709;"  XRDGS 500 INTO AMPLT"
240  !
250  !Define and store the arbitrary waveform. Specify the array loaded
260  !by the HP 44702A as the amplitude array, and set the time base
270  !equal to the sample rate of the voltmeter. Select waveform 4 in
280  !channel 0 memory. Output 10 cycles of the waveform.
290  !
300  OUTPUT 709;"USE 0"
310  OUTPUT 709;"  WFWRITE ARB 4 AMPLT TBASE 10E-6"
320  OUTPUT 709;"  APPLY WFV 0,4"
330  OUTPUT 709;"  NSCAN 10"
340  !
350  !Enable the HP 44726A to interrupt once the cycles have completed.
360  !Enable the mainframe to sense the interrupt from the accessory.
370  !Arm the channel so that when triggers are received, the
380  !waveform is output from the DAC.
390  !
400  OUTPUT 709;"  ENABLE INTR"
410  OUTPUT 709;"  ENABLE INTR SYS"
420  OUTPUT 709;"  TRIG INT"
430  OUTPUT 709;"  TARM AUTO"
440  END
```

Example 2: Enable HP 44726A Interrupt, Handle with Controller

This program enables the HP 44726A in slot 0 of the mainframe to interrupt after 1,000 cycles of a special function sine wave have completed.

When the interrupt occurs, the HP-IB SRQ line is set TRUE and the interrupt is sent to the controller. The controller's interrupt routine then displays CYCLES COMPLETE.

```
10   !Enable the controller to respond to an interrupt. Reset the
20   !mainframe and the HP 44726A. Turn the RQS mode on and unmask the
30   !backplane (accessory) interrupt bit (INTR) in the status register.
40   !Ensure that the INTR bit is cleared and remove the data returned
50   !by STA? from the output buffer.
60   !
70   ON INTR 7 GOSUB Results
80   ENABLE INTR 7;2
90   OUTPUT 709;"RST"
100  WAIT 1
110  OUTPUT 709;"RQS ON"
120  OUTPUT 709;"RQS INTR"
130  OUTPUT 709;"STA?"
```

```
140   OUTPUT 709;"CLROUT"
150   !
160   !Define and store a special function sine wave.  The sine wave will
170   !be a 100 point waveform, 5 Vp-p, 5 kHz, with a 3V DC offset. Select
180   !the waveform in channel 0 memory. Turn on the anti-aliasing filter
190   !to smooth the waveform. Output 1,000 cycles of the waveform.
200   !
210   OUTPUT 709;"WFWRITE ACV 7 5 OFFSET 3 PTS 100 TBASE (1/(5E3*100))"
220   OUTPUT 709;"APPLY WFV 0,7"
230   OUTPUT 709;"FILTER ON"
240   OUTPUT 709;"NSCAN 1000"
250   !
260   !Enable the HP 44726A to interrupt once the cycles have completed.
270   !Enable the mainframe to sense the interrupt from the accessory.
280   !Arm the channel so that when triggers are received, the
290   !waveform is output from the DAC.
300   !
310   OUTPUT 709;"ENABLE INTR"
320   OUTPUT 709;"ENABLE INTR SYS"
330   OUTPUT 709;"TRIG INT"
340   OUTPUT 709;"TARM AUTO"
350   GOTO 350
360   !
370   !Display the message on the controller when the interrupt occurs.
380   !Clear the service request bit in order for the controller to
390   !respond to the next interrupt that occurs.
400   !
410 Results:   !
420   PRINT "CYCLES COMPLETE"
430   A=SPOLL(709)
440   END
```

ENABLE MULTI

- **Mainframe**

Description

Enable HP 3852A multitasking capability. Execution of this command resets the HP 3852A and places the instrument in the multitasking mode.

Prerequisites

The HP 3852A multitasking capability is only available with mainframe firmware revision 3.0 or greater.

Syntax

ENABLE MULTI

Parameters

None

Remarks

ENABLE MULTI Causes a System Reset

When ENABLE MULTI is executed, a system reset occurs. The reset returns each installed accessory to its power-on state and also erases any previously declared variables, arrays, and downloaded subroutines. A reset clears the mainframe's input and output buffers; however, the size of the buffers and the size of the symbol table remains as last set by the INBUF, OUTBUF, and SYMSIZE commands or as set at power-on.

Since a reset is required to load the TSLICE and NTASKS settings into the mainframe's operating system, do not declare variables, arrays, or write and download subroutines until the multitasking mode has been set, and the time-slice period, number of run tasks, and queue size have been loaded.

Note that a system reset takes approximately 600 ms to complete. Although not shown with each command reference entry, it is recommended that a WAIT statement be included following ENABLE MULTI. This will prevent errors from occurring in subsequent commands due to system configurations changed during the reset.

Task Definition

When the HP 3852A is in the multitasking mode, commands and subroutines execute within environments called tasks. Specifically, the HP 3852A tasks are:

```
Front Panel
HP-IB
Interrupt
Run Tasks
```

For example, a command entered from the front panel executes within the front panel task. Similarly, a command sent over the HP-IB executes within the HP-IB task. The Interrupt task executes a subroutine called in response to an interrupt that has occurred (e.g. ON event CALL name). Run Tasks execute subroutines that have been directed to a particular run task number. Note that a subroutine called by the CALL command executes in the task where the CALL command was executed.

Task Priorities Set by ENABLE MULTI

When ENABLE MULTI is executed, the following task priorities (urgencies) are established:

```
Front Panel  (25)      HIGHEST - executed first
HP-IB        (45)
Interrupt    (65)
Run Tasks    (85)      LOWEST - executed last
```

Exiting the Multitasking Mode

Once the mainframe has been placed in the multitasking mode, it remains in that mode until power is cycled or until the DISABLE MULTI command is executed.

Effect of the Multitasking Mode on Variables, Arrays, and the USE Channel

As previously mentioned, when ENABLE MULTI is executed, a reset of the mainframe occurs which erases any previously declared variables, arrays, and downloaded subroutines. When arrays and variables are declared in the multitasking mode, they are global. This means an array or variable declared from a specific task (i.e. front panel, HP-IB, interrupt, run task) can be accessed by commands from any other task.

In the multitasking mode, the USE channel specified by the USE command is local to the task. This means a USE channel specified from a particular task applies only to the commands in that task. All other tasks which do not specify a USE channel when required will use the default USE channel set when the mainframe is reset. This may not be desirable for all applications.

Multitasking Commands set by ENABLE MULTI

When executed, the ENABLE MULTI command also sets ENABLE EOL SWAP and URGENCY. Refer to these commands for information on their function or if different settings are required.

Data Returned

None

Related Commands

DISABLE MULTI, ENABLE EOL SWAP, DISABLE EOL SWAP, URGENCY, TSLICE, NTASKS, RUN

ENABLE MULTI (cont)

Examples

Example: Setting the Multitasking Mode

This example shows how the ENABLE MULTI command is used to set the multitasking mode. Other commands are included to show the context in which ENABLE MULTI is used. For this example, two subroutines are set up and directed to run tasks. One subroutine counts and displays the numbers from 0 through 9 and the other subroutine repeatedly displays the sequence A B C. The HP 3852A's multitasking capability is demonstrated as the mainframe will alternately count and display A B C.

```
10    !Set a 1.5 second time-slice and set the number of run tasks and
20    !queue size the system will allow. Place the HP 3852A in the
30    !multitasking mode. The reset caused by ENABLE MULTI will load
40    !the TSLICE and NTASKS parameters into the operating system.
50    !
60    OUTPUT 709;"TSLICE 1.5"
70    OUTPUT 709;"NTASKS 2,2"
80    OUTPUT 709;"ENABLE MULTI"
90    WAIT 1
100   !
110   !Write and download the subroutines to be directed to run tasks.
120   !Subroutine A when active, displays the numbers 0 through 9.
130   !Subroutine B when active, repeatedly displays A B C.
140   !
150   OUTPUT 709;"SUB A"
160   OUTPUT 709;"  INTEGER I"
170   OUTPUT 709;"  FOR I=0 TO 9"
180   OUTPUT 709;"    DISP I"
190   OUTPUT 709;"    WAIT .5"
200   OUTPUT 709;"  NEXT I"
210   OUTPUT 709;"SUBEND"
220   OUTPUT 709;"SUB B"
230   OUTPUT 709;"  INTEGER J"
240   OUTPUT 709;"  FOR J=0 TO 9"
250   OUTPUT 709;"    DISP 'A'"
260   OUTPUT 709;"    WAIT .5"
270   OUTPUT 709;"    DISP 'B'"
280   OUTPUT 709;"    WAIT .5"
290   OUTPUT 709;"    DISP 'C'"
300   OUTPUT 709;"    WAIT .5"
310   OUTPUT 709;"  NEXT J"
320   OUTPUT 709;"SUBEND"
330   !
340   !Direct the subroutines to run tasks. Set run task 1 (subroutine
350   !A) to execute three times and run task 2 (subroutine B) to
360   !execute one time.
370   !
380   OUTPUT 709;"RUN 1 A 3"
390   OUTPUT 709;"RUN 2 B"
400   END
```

When this program runs, line 60 specifies the time-slice period and line 70 specifies the number of run tasks and the queue size the system will allow. Line

80 places the HP 3852A in the multitasking mode and loads the TSLICE and NTASKS values into the operating system. Note that specifying the exact number of run tasks and matching queue size prevents mainframe memory (overhead) from being allocated for unused (non-existent) run tasks.

After the subroutines are downloaded (lines 150 through 320), the RUN command is used to direct subroutines A and B to run tasks 1 and 2 and specify the number of times they are to run.

As the run tasks execute, the following is displayed in the sequence shown:

```
0           (run task 1 active - run task 2 scheduled)
1
2

A           (run task 2 active - run task 1 scheduled)
B
C

3           (run task 1 active - run task 2 scheduled)
4
5

A           (run task 2 active - run task 1 scheduled)
B
C
.
.
.
```

The display sequence shown demonstrates an important point about HP 3852A multitasking. Although the HP 3852A is executing subroutines within multiple tasks, only one task is active at a time. As shown above, when run task 1 is active, 0 1 2 is displayed. When run task 2 becomes active, A B C is displayed. When run task 1 becomes active again, 3 4 5 is displayed. Notice the counting sequence of run task 1 started from where it left off. The subroutine was not executed simultaneously with the subroutine of run task 2, thus, it did not continue to count in the "background".

If executed as shown, this program runs to completion in about 35 seconds.

ENABLE/DISABLE INTR BNC

- Mainframe

Description

Connects the backplane interrupt line to the CHANNEL CLOSED BNC. This enables backplane (accessory) interrupts to function as a trigger source for other accessories.

Prerequisites

Requires firmware revision 3.5 or greater and the 03852-66523 controller module.

Syntax

ENABLE INTR BNC

or

DISABLE INTR BNC

Parameters

None

Remarks

Handling Backplane Interrupts

To use a backplane interrupt as a trigger, the accessory must be configured to interrupt on the appropriate condition and the channel's interrupt capability must be enabled (ENABLE INTR USE ch). If ENABLE INTR SYS is also set, the mainframe will service the interrupt which can then be handled by the mainframe or by the controller.

Pulse Duration

When channel interrupt capability has been enabled (ENABLE INTR USE ch) and ENABLE INTR BNC is set, the CHANNEL CLOSED BNC is pulled low the instant an interrupt occurs. The output remains low until the channel is disabled (DISABLE INTR USE ch) or until the interrupt is cleared. If multiple interrupts have occurred, the output remains low until all interrupts have been disabled or cleared.

Routing the Interrupt

With an interrupt as the trigger source, connecting the CHANNEL CLOSED BNC to the ports listed below will initiate activities suspended by the following conditions.

```
CHANNEL CLOSED BNC to:

        • EVENT IN BNC - initiates functions suspended by
         WAITFOR EVENT.

        • SYSTEM TRIGGER IN BNC - initiates functions suspended
        due to an accessory set to TRIG SYS and the system trigger set
        to TRG EXT.

        • PACER TRIGGER IN - triggers the system pacer which was
        idle due to PTRIG EXT.

        • an accessory's (e.g. HP 44702A/B, HP 44726A) external
        trigger port will activate the accessory whose trigger source
        is set accordingly.
```

The advantage of using ENABLE INTR BNC is when the interrupt occurs, a function suspended pending the interrupt can begin within microseconds of the interrupt. On the other hand, using the interrupt to call a subroutine would start the function within milliseconds of the interrupt.

Channel Closed Pulses do not Occur When ENABLE INTR BNC is Set

When ENABLE INTR BNC is set, a pulse occurs on the CHANNEL CLOSED BNC only when a backplane interrupt occurs. At power-on, following a reset, or following the DISABLE INTR BNC command, pulses occur on the BNC whenever an accessory channel is closed.

Data Returned

None

Related Commands

ENABLE INTR USE ch, ENABLE INTR SYS

Examples

Example: Triggering From an Interrupt

The following example shows how a digital input accessory is enabled to interrupt and how the interrupt is used to trigger a voltmeter accessory.

In the program, an HP 44721A digital input is configured to interrupt on a low to high edge transition. The HP 44702A voltmeter is set up to take 100 DC voltage measurements when it receives a trigger through its EXT 0 BNC port. ENABLE INTR BNC is executed so that when the interrupt occurs, a pulse will appear on the CHANNEL CLOSED BNC. By connecting a BNC cable between the CHANNEL CLOSED BNC and the voltmeter's EXT 0 port, the interrupt pulse will be routed to the EXT 0 port, at which point the voltmeter is triggered.

ENABLE/DISABLE INTR BNC (cont)

A subroutine is downloaded and then called when the interrupt occurs to demonstrate that interrupts used as trigger sources can still be serviced by the mainframe.

```
10    !Declare a controller array to store the readings taken when the
20    !HP 44702A voltmeter is triggered by the interrupt. Reset the
30    !HP 3852A and then download a subroutine to be called when the
40    !interrupt occurs. Enable channel 8 of the digital input in slot
50    !3 to interrupt on a low to high edge transition.
60    !
70    REAL V_data(0:99)
80    OUTPUT 709;"RST"
90    OUTPUT 709;"SUB A"
100   OUTPUT 709;"  DISP'VOLTMETER TRIGGERED'"
110   OUTPUT 709;"SUBEND"
120   OUTPUT 709;"ON INTR USE 324 CALL A"
130   OUTPUT 709;"EDGE LH USE 324"
140   !
150   !Set up the voltmeter to make 100 DC voltage measurements. Trigger
160   !the voltmeter through its rear panel EXTO BNC. Set the slope
170   !such that the voltmeter triggers on a high to low transition.
180   !
190   OUTPUT 709;"USE 400"
200   OUTPUT 709;"  FUNC DCV"
210   OUTPUT 709;"  TERM EXT"
220   OUTPUT 709;"  NRDGS 100"
230   OUTPUT 709;"  SLOPE HL"
240   OUTPUT 709;"  TRIG EXTO"
250   !
260   !Connect the backplane interrupt line to the CHANNEL CLOSED BNC.
270   !(A BNC cable is connected between the CHANNEL CLOSED BNC and the
280   !voltmeter's EXT 0 port). Enable the digital input channel to
290   !interrupt on the specified edge. Enable the mainframe to sense
300   !(and service) the interrupt so that the subroutine is called.
310   !
320   OUTPUT 709;"ENABLE INTR BNC"
330   OUTPUT 709;"ENABLE INTR USE 324"
340   OUTPUT 709;"ENABLE INTR SYS"
350   !
360   !After the interrupt occurs and triggers the voltmeter, transfer
370   !the readings to the output buffer. Enter and display the readings
380   !on the controller.
390   !
400   OUTPUT 709;"XRDGS 400"
410   ENTER 709;V_data(*)
420   PRINT V_data(*)
430   END
```

Once the program executes and an interrupt occurs, VOLTMETER TRIGGERED is displayed on the HP 3852A and the readings taken are displayed on the controller. One line of readings appears as shown below:

```
4.775    4.775    4.775    4.775    4.775    4.775    4.775    4.775
```

• Mainframe

Description

Changes the width of the pacer pulse from 0.5 us to 5.0 us.

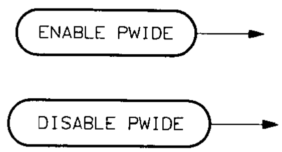
Prerequisites

Requires firmware revision 3.5 or greater and the 03852-66523 controller module.

Syntax

ENABLE PWIDE

DISABLE PWIDE



Parameters

None

Remarks

ENABLE/DISABLE PWIDE Can Execute During Any Pulse Output

Executing ENABLE PWIDE or DISABLE PWIDE will change the width of the pacer pulse if the pacer is currently outputting a fixed number or continuous train of pulses.

Executing DISABLE PWIDE

DISABLE PWIDE changes the width of the pacer pulse from 5.0 us to 0.5 us.

Data Returned

None

Related Commands

PACER, PTRIG

Example

Example: Configuring the System Pacer

The following program segment shows the sequence of commands used to set up and activate the pacer.

```
100  !Program the pacer to continuously output a pacer pulse every
110  !100 us. Set the pulse width to 5 us. Issue a single trigger to
120  !activate the pacer.
130  !
140  OUTPUT 709;"PACER 100E-6"
150  OUTPUT 709;"ENABLE PWIDE"
160  OUTPUT 709;"PTRIG SGL"
```

END

• **Mainframe**

Description

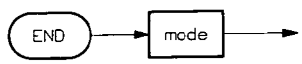
Suppress or assert the End-Or-Identify (EOI) signal concurrent with the last byte of data returned by an individual command.

Prerequisites

The END command is only available with mainframe firmware revisions 2.2 and greater.

Syntax

END *mode*



Parameters

mode Suppresses or asserts EOI concurrent with the last byte of data returned by each command. At power-on, *mode* is ON.

mode	Description
OFF	Suppresses EOI.
ON	EOI is asserted with the last byte of data returned by the command.

Remarks

Setting OUTBUF ON When Using END

The purpose of the END command is to suppress the EOI signal thus enabling you to enter multiple data sets from the HP 3852A HP-IB output buffer into the controller with a single ENTER statement. Multiple data sets are available with OUTBUF ON since data returned by the HP 3852A is appended to the data currently in the output buffer. With OUTBUF OFF, data is overwritten.

Data Returned

None

Related Commands

OUTBUF

Examples

Example: Entering Multiple Data Sets with END ON

The following program illustrates one method of how data can be entered into the controller when OUTBUF ON is set and when EOI is asserted with the last byte of each data set sent to the output buffer (END ON).

```
 10 DIM Dcrdgs(0:4)                    !Dimension array for first data set
 20 DIM Dcrdgs1(0:4)                   !Dimension array for second data set
 30 OUTPUT 709;"RST"                   !Reset the HP 3852A
 40 OUTPUT 709;"OUTBUF ON"             !Enable output data buffering
 50 OUTPUT 709;"END ON"                !Assert EOI
 60 OUTPUT 709;"USE 700"               !Use the voltmeter in slot 7
 70 OUTPUT 709;"CONFMEAS DCV 400-404"  !Generate first data set
 80 OUTPUT 709;"RANGE 10"              !Specify maximum signal amplitude
 90 OUTPUT 709;"MEAS DCV, 405-409"     !Generate second data set
100 ENTER 709;Dcrdgs(*)                !Enter first data set
110 ENTER 709;Dcrdgs1(*)               !Enter second data set
120 END
```

In this program, two enter statements (lines 100 and 110) are required to enter the data from the HP 3852A output buffer into the controller. The two statements are necessary since EOI was not suppressed and is asserted with the last byte of each data set in the buffer. EOI immediately terminates each ENTER statement.

Example: Entering Multiple Data Sets With END OFF

The following program suppresses the EOI signal enabling data to be retrieved from the HP 3852A HP-IB output buffer with a single ENTER statement.

```
 10 DIM Dcrdgs1(0:9)                   !Dimension array for both data sets
 20 OUTPUT 709;"RST"                   !Reset the HP 3852A
 30 OUTPUT 709;"OUTBUF ON"             !Enable output data buffering
 40 OUTPUT 709;"END OFF"               !Suppress EOI
 50 OUTPUT 709;"USE 700"               !Use the voltmeter in slot 7
 60 OUTPUT 709;"CONFMEAS DCV 400-404"  !Generate first data set
 70 OUTPUT 709;"RANGE 10"              !Specify maximum signal amplitude
 80 OUTPUT 709;"MEAS DCV, 405-409"     !Generate second data set
 90 ENTER 709;Dcrdgs(*)                !Enter both data sets
100 END
```

END IF

• **Mainframe**

Description

See **IF...END IF** Command

- Mainframe

Description

See **WHILE ...END WHILE** Command

ENTER

• HP 44788A HP-IB Controller

Description

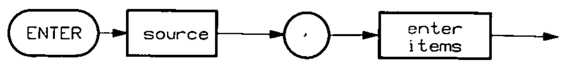
This statement is used to input data from a device or file and assign the values entered to variables.

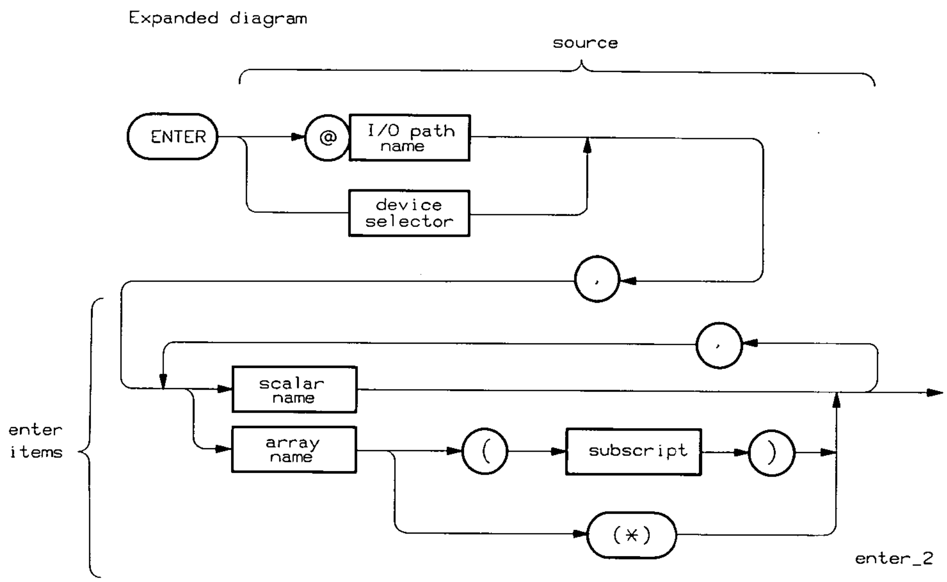
Prerequisites

Requires firmware revision 3.5 or greater.

Syntax

ENTER *@I/O path name* or *device selector*, *enter_items*





Parameters

@I/O path name The name of the bidirectional path assigned to a device or devices that data is to be entered from.

device selector The HP-IB select code (ESnn) for the device that data is to be entered from. The local HP-IB address is always ESnn, where E = extender # (mainframe =0), S = slot, and nn = device address.

enter_items The variables that the items received will be entered into.

Remarks

The Number Builder

If the data being received is ASCII and the associated variable is numeric, a number builder is used to create a numeric quantity from the ASCII representation. The number builder ignores all leading non-numeric characters and blanks, and terminates on the first non-numeric character. (Numeric characters are 0 through 9, +, -, decimal point, e, and E, in a meaningful numeric order).

If the number cannot be converted to the type of the associated variable, an error is generated. If more digits are received than can be stored in a variable of type REAL, the rightmost digits are lost, but any exponent will be built correctly. Overflow occurs only if the exponent overflows.

Entering Arrays

Entire arrays may be entered by using the asterisk specifier. Each element in an array is treated as an item by the ENTER statement, as if the elements were listed separately.

Files as Source

If an I/O path has been assigned to a file, the file may be read with ENTER statements. The file must be an ASCII or BDAT file, The FORMAT attribute in the ASSIGN statement is used only if the file is a BDAT file. Data read from an ASCII file is always in ASCII format. Data read from a BDAT file is considered to be in internal format if FORMAT is OFF, and is read as ASCII characters if FORMAT is ON.

Serial access is used for both ASCII and BDAT files. The file pointer is set to the beginning of the file when the file is opened by an ASSIGN. The file pointer always points to the next byte available for ENTER operations.

Devices as Source

An I/O path name or a device selector may be used to ENTER from a device. If a device selector is used, the default system attributes are used (see ASSIGN). If an I/O path name is used, the ASSIGN statement determines the attributes used.

If FORMAT ON is the current attribute, the items are read as ASCII. If FORMAT OFF is the current attribute, items are read from the device in the HP 3852A's internal format. Two bytes are read for each INTEGER, eight bytes for each REAL.

Data Returned

None

Related Commands

None

Examples

Example: Typical ENTER Statements

The following statements enter data to devices. The first statement is entered from the HP 3852A front panel, while the second statement is entered from a system controller.

```
ENTER 405,Number,NUM2                    !Enter Number,NUM2 to device 405
110 OUTPUT 709;"ENTER @File, Array(*)"   !Enter Array over I/O path @File
```

• **Mainframe**

Description

Reads the error code stored in the HP 3852A error buffer.

Prerequisites

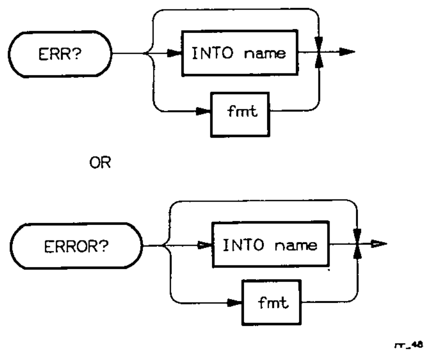
None

Syntax

ERR? [**INTO** *name*] or [*fmt*]

or

ERROR? [**INTO** *name*] or [*fmt*]



Parameters

INTO *name* See Glossary.

fmt See Glossary. Default format is IASC.

Remarks

The ERR? Command Clears the Error Buffer

The codes and messages for up to four errors are stored in the HP 3852A error buffer. ERR? reads the error code and erases the code (and message) from the error buffer one error at a time. Errors that occur after the buffer fills will be displayed as usual, but will not be stored in the error buffer.

The ERR bit in the status register and the ERR annunciator are set when the buffer contains an error message. When all errors in the buffer have been read, the error buffer, the ERR bit in the status register, and the ERR annunciator are cleared. Reading an error code (or message) makes room for a new error code and message.

RST Clears the Error Buffer

The RST (or RESET) command clears the error buffer and the ERR bit in the status register.

Data Returned

One integer is returned corresponding to an error code in the error buffer. Errors are read in first-in-first-out fashion. That is, the first error to occur is the first error to be read. If no error codes or messages are present in the error buffer, 0 is returned. (Refer to the "Error Messages" section of this manual for a list of HP 3852A error codes and messages.)

Related Commands

ERRSTR?

Examples

Example: Reading Error Codes

This program shows how to use the ERR? command to read error codes from the error buffer.

```
10 INTEGER Err(0:3)
20 FOR I =0 to 3
30 OUTPUT 709;"ERR?"
40 ENTER 709;Err(I)
50 NEXT I
60 PRINT USING "K,/";Err(*)
70 END
```

A typical display if errors 72, 2, 74, and 71 have occurred (in that order) follows.

```
72
 2
74
71
```

• Mainframe

Description

See **ERR?** Command

ERRSTR?

- **Mainframe**

Description

Reads the error code and the error message stored in the HP 3852A error buffer.

Prerequisites

None

Syntax

ERRSTR?

Parameters

None

Remarks

The ERRSTR? Command Clears the Error Buffer

The codes and messages for up to four errors can be stored in the HP 3852A error buffer. ERRSTR? reads the error code and message and erases the code and message from the error buffer one error at a time. Errors that occur after the buffer fills will be displayed as usual, but will not be stored in the error buffer.

The ERR bit in the status register and the ERR annunciator are set when the buffer contains an error message. When all errors in the buffer have been read, the error buffer, the ERR bit in the status register, and the ERR annunciator are cleared. Reading an error code or message makes room for a new error code and message.

RST Clears the Error Buffer

The RST (or RESET) command clears the error buffer and the ERR bit in the status register.

Data Returned

One 60-character string is returned corresponding to an error code and error message in the error buffer. Errors are read in first-in-first-out fashion. That is, the first error to occur is the first error to be read. If no error codes and messages are present in the error buffer, "0   NO ERROR   " is returned. (Refer to the "Error Messages" section of this manual for a list of HP 3852A error codes and messages.)

Related Commands

ERR?

Examples

Example: Reading Error Codes and Messages

This program shows how to use the ERRSTR? command to read error codes and messages from the error buffer.

```
10 DIM Errstr$(0:3)[60]
20 FOR I =0 to 3
30 OUTPUT 709;"ERRSTR?"
40 ENTER 709;Errstr$(I)
50 NEXT I
60 PRINT USING "K,/";Errstr$(*)
70 END
```

A typical display if errors 72, 2, 74, and 71 have occurred (in that order) follows. For each error shown, the data which was entered in error is displayed at the end of the error message (this is not the case for all error messages).

```
"   72: THIS KEYWORD NOT EXPECTED - TOTAL              "
"    2: SYMBOL TOO LONG - CONFFMEAS                    "
"   74: COMMAND END NOT EXPECTED - OPEN +;             "
"   71: UNDEFINED WORD - COND                          "
```

EXOR

• **Logical Operator**

Description

Returns a 0 or a 1 based on the logical exclusive-or of the arguments.

Prerequisites

Requires mainframe firmware revision 3.5 or greater.

Syntax

number **EXOR** *number*

38521P R.NUM

Parameters

number Constant, variable, or numeric expression. If one argument specified is a non-zero value, a 1 is returned. If both arguments are 0, or both are non-zero, a 0 is returned.

Remarks

EXOR Truth Table

The EXOR function tests whether an argument specified has a value of zero. The truth table for this function is shown below:

A	B	A EXOR B
z	z	0
z	nz	1
nz	z	1
nz	nz	0

```
 z: value = 0
nz: value is non-zero
```

Data Returned

See Parameters.

Related Commands

None

Examples

Example: Demonstrating the EXOR Function

The following program demonstrates the EXOR function.

```
10    OUTPUT 709;"RST"
20    OUTPUT 709;"INTEGER A,B,C,D"
30    OUTPUT 709;"A=1"
40    OUTPUT 709;"B=1"
50    OUTPUT 709;"C=0"
60    OUTPUT 709;"D=1"
70    OUTPUT 709;"SUB FNCT"
80    OUTPUT 709;"  IF (A EXOR B) AND (C OR D) THEN"
90    OUTPUT 709;"    DISP'EXPRESSION IS TRUE'"
100   OUTPUT 709;"  ELSE"
110   OUTPUT 709;"    DISP'EXPRESSION IS FALSE'"
120   OUTPUT 709;"  END IF"
130   OUTPUT 709;"SUBEND"
140   OUTPUT 709;"CALL FNCT"
150   END
```

When the program executes, variables A, B, C, and D are assigned the values shown in lines 30 - 60. Given these values, A exclusive-or'ed with B returns a 0, and C or'ed with D returns a 1. The 0 AND'ed with the 1 evaluates to 0 and

```
EXPRESSION IS FALSE
```

is displayed on the HP 3852A.

EXP

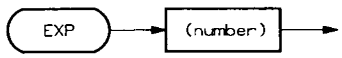
- **Math Function**

Description

Numeric expression evaluated as a command parameter. Raises the base e to the power specified by its argument.

Syntax

EXP (*number*)



RR_58

Parameters

number Number or numeric expression which must evaluate within -708.396418532264 to +709.7827128933838.

Remarks

None

Examples

Example: Using EXP Function

This program raises e to the 10th power and returns the result (22026.47) to the controller CRT.

```
10 OUTPUT 709;"RST"            !Reset the HP 3852A
20 OUTPUT 709;"VREAD EXP (10)" !Raise e to the 10th power, read value
30 ENTER 709;A                 !Enter value to controller
40 PRINT "Value = ";A          !Display value
50 END
```

For this program, a typical display is:

```
Value = 22026.47
```

EXTEND?

• **Mainframe**

Description

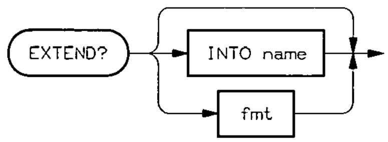
Identifies the HP 3853A Extenders connected to the HP 3852A. The command returns seven numbers representing the number(s) of the extender(s) connected to the HP 3852A.

Prerequisites

None

Syntax

EXTEND? [**INTO** *name*] or [*fmt*]



Parameters

INTO *name* — See Glossary.

fmt — See Glossary. Default format is IASC.

Remarks

Entering EXTEND? from the HP 3852A Front Panel

When entering EXTEND? from the front panel, use FASTDISP OFF to see each value.

Data Returned

EXTEND? returns seven numbers ranging from 0-7 which represent the extenders connected to the HP 3852A. For those extender numbers not used, zeros are returned.

Related Commands

None

Examples

Example: Identifying Extenders Connected

```
10  INTEGER Extender(0:6)
20  OUTPUT 709;"RST"
30  OUTPUT 709;"EXTEND?"
40  ENTER 709;Extender(*)
50  PRINT Extender(*)
60  END
```

A typical return if two HP 3853A Extenders with extender numbers of 2 and 5 are connected to the mainframe is:

```
0  2  0  0  5  0  0
```

FASTDISP

- **Mainframe**

Description

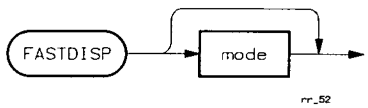
Enable/disable the fast display mode. With fast display mode enabled, results are displayed as quickly as they are generated. With fast display mode disabled, results are displayed slowly enough to be seen.

Prerequisites

None

Syntax

FASTDISP [*mode*]



Parameters

mode Enables or disables the fast display mode. Power-on and default *mode* = ON.

mode	Definition
ON	Fast display mode enabled.
OFF	Fast display mode disabled.

Remarks

FASTDISP ON/OFF Considerations

FASTDISP affects the display only if the display is on (DISP ON). FASTDISP OFF is useful to see each result in commands which return multiple results, such as EXTEND? or MEAS. However, FASTDISP OFF (especially with DISP ON and MON ON) greatly slows down command processing time.

Data Returned

None

Related Commands

None

Examples

This program line shows how to disable the fast display mode.

```
10 OUTPUT 709;"FASTDISP OFF"    !Disable the fast display mode.
```

• **Mainframe**

Description

Fast output mode. With FASTOUT OFF, data is output at maximum rate available (about 120 kbytes/sec) with standard HP-IB configuration. With FASTOUT ON, data is output at maximum rate (about 140 kbytes/sec) for a specific HP-IB configuration.

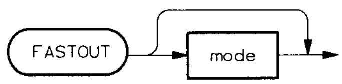
Prerequisites

For maximum output rate with FASTOUT ON, set the following HP-IB bus conditions:

- Power to all HP-IB Devices.
- < 50 pF capacitive loading at each device.
- ≤ 15 metres of HP-IB cable.
- At least one device load per metre of cable.

Syntax

FASTOUT [*mode*]



rr_53

Parameters

mode Sets maximum output data rate. Power-on = FASTOUT OFF; default = FASTOUT ON. With FASTOUT OFF, maximum data output rate = 120 kbytes/sec for standard HP-IB configuration. With FASTOUT ON, maximum data output rate = 140 kbytes/sec for HP-IB configuration shown in "Prerequisites".

Remarks

Data Returned

None

Related Commands

None

Examples

Example: Set Fast Output Mode

This program line sets the HP 3852A for fast output mode (140 kbytes/sec) if the HP-IB system is properly configured.

```
10 OUTPUT 709;"FASTOUT ON"
```

• **HP 44726A 2-Channel Arbitrary Waveform DAC**

Description

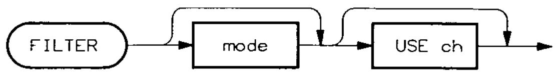
Switches the channel's anti-aliasing filter into the signal path of special function and arbitrary waveforms. Filtering increases the waveform's settling time from 12 us to approximately 30 us.

Prerequisites

Requires firmware revision 3.5 or greater.

Syntax

FILTER [*mode*] [USE *ch*]



Parameters

mode Controls the use of the filter.

ON - switches the filter into the signal path. The default *mode* is ON.

OFF - removes the filter from the signal path. The power-on *mode* is OFF.

USE *ch* Channel whose filter is switched into the signal path. The default USE *ch* is channel 0.

Remarks

When to Use the Filter

Due to the filter's cutoff frequency of 150 kHz, the filter is the most useful for waveforms with a time base between 3.0 us (333.3 kHz) and 1.25 us (800 kHz).

Waveform Settling Time

Waveform settling time as it applies to special function and arbitrary waveforms is the time it takes the DAC output to settle due to large changes between amplitude points. Settling time approaches 30 us when the filter is on and a 20V step between amplitude points exists.

Data Returned

None

Related Commands

None

Examples

Example: Using the Filter to Smooth a Waveform

The following program shows how the channel's filter is used to smooth a waveform.

In the example, a 100 kHz, 10 Vp-p, eight point sine wave is defined and stored. Since the time base required to produce the waveform described is 1.25 us, the filter's 150 kHz cut off frequency attenuates the aliasing components generated. By switching the filter inside and outside of the signal path, the effects of the filter on the sine wave defined can be seen.

```
10    !Reset the HP 3852A and the HP 44726A. Define and store the special
20    !function sine wave. The waveform will be an 8 point sine wave,
30    !10 Vp-p, with a frequency of 100 kHz.
40    !
50    OUTPUT 709;"RST"
60    WAIT 1
70    OUTPUT 709;"WFWRITE ACV 0 10 PTS 8 TBASE(1/(100E3*8))"
80    !
90    !Switch channel 0's anti-aliasing filter into the signal path in
100   !order to smooth the waveform. Select the waveform from channel 0
110   !memory. Set the trigger source which will advance the output of the
120   !DAC. Arm the channel so that the trigger signals will be accepted.
130   !
140   OUTPUT 709;"FILTER ON"
150   OUTPUT 709;"APPLY WFV 0,0"
160   OUTPUT 709;"TRIG INT"
170   OUTPUT 709;"TARM AUTO"
180   END
```

FILTER

- **HP 44730A Track/Hold with Signal Conditioning**
- **HP 44732A 120 Ohm Dynamic Strain Gage FET Multiplexer**
- **HP 44733A 350 Ohm Dynamic Strain Gage FET Multiplexer**

Description

Adds a filter to the specified channel.

Prerequisites

Command requires mainframe firmware revision 3.5 or greater. The system must not be scanning with the HP 44702A/B voltmeter when FILTER is executed.

Syntax

FILTER *function* [USE *ch*]

Parameters

function Adds a 4-pole Bessel filter to the specified channel. Power-on/reset *function* = OFF.

function	Description
OFF	Filter is removed from channel input signal path.
ON	Filter is added to channel input signal path.

USE *ch* Specify channel to be used for the FILTER command. Channel number range = ES00 through ES03.

Remarks

Default Filter Characteristics

This table shows channel performance with the default (factory-installed) filter added to a channel (In) vs. channel performance with the filter out of the channel (Out). Note that filtering does not affect Common Mode Rejection Ratio (CMRR). Full-power bandwidth for all gain settings is 190 kHz. (RTI = Referred to Input.)

Gain	3 dB Bandwidth		RMS Noise RTI		CMRR @ 10 kHz	
	In	Out	In	Out	In	Out
1	10 kHz	1 MHz	51 µV	510 µV	30 dB	30 dB
10	10 kHz	1 MHz	6.4 µV	64 µV	50 dB	50 dB
100	10 kHz	300 kHz	2.0 µV	11 µV	70 dB	70 dB

Changing Filter Components

As factory-set, each channel contains a 4-pole Bessel (low-pass) filter with cutoff frequencies in the range of 10 Hz to 10 kHz. The filter characteristics can be changed by changing resistors on the HP 44732A or HP 44733A terminal module. Refer to the HP 44730A or HP 44732A/44733A Configuration and Programming Manual for details.

Data Returned

None.

Related Commands

None.

Examples

Example: Adding Channel Filter

This program uses FUNC to set channel 500 of an HP 44730A to amplification with gain of 10. Filtering is added to channel 500 with the FILTER command. Channel measurements are made with an HP 44701A voltmeter in slot 6 via the mainframe backplane.

Since ribbon cable operation is not used, the ribbon cable on the HP 44730A must be looped back to the connector on the component module. Note that CONFMEAS automatically closes the channel and the isolation relay (channel 590). If a CLOSE command is used, the isolation relay and the channel must both be closed (OUTPUT 709;"CLOSE 500,590").

```
10  OUTPUT 709;"USE 500"             !HP 44730A use ch is 500
20  OUTPUT 709;"FUNC AMPLIFY,10"     !Ch 500 to amplify, gain = 10
30  OUTPUT 709;"FILTER ON"           !Add filter to ch 500
40  OUTPUT 709;"AZERO ONCE"          !Autozero on ch 500
50  OUTPUT 709;"USE 600"             !Use 44701A voltmeter in slot 6
60  OUTPUT 709;"CONFMEAS DCV,500"    !Conf/meas DC volts on ch 500
70  ENTER 709;A                      !Enter reading
80  PRINT A                          !Display reading
90  END
```

For a 0.5 V, 5% input, a typical return is:

```
4.98
```

FOR...NEXT

- **Mainframe**

Description

Defines a loop that is repeated until the loop counter passes a specified value.

Prerequisites

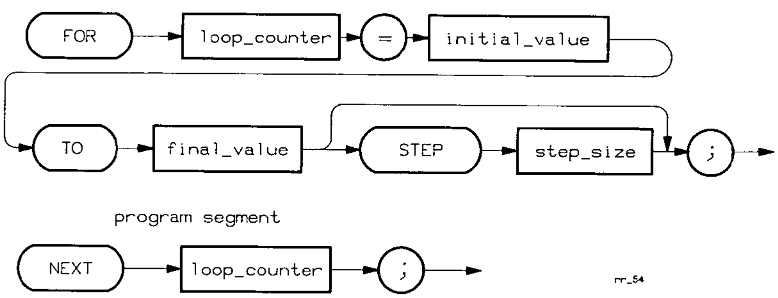
The *loop_counter* name must have been defined with a REAL or INTEGER command. The FOR...NEXT construct must be part of a subroutine (SUB) entry.

Syntax

FOR *loop_counter* = *initial_value* **TO** *final_value* [**STEP** *step_size*]

program segment

NEXT *loop_counter*



Parameters

loop_counter — Name of the numeric variable which acts as the loop counter.

initial_value — Numeric expression that is the beginning value of the loop counter.

final_value — Numeric expression that is the ending value of the loop counter.

step_size — Numeric expression which specifies the amount to increment or decrement the loop counter for each pass through the loop. Default = 1.

Remarks

Using FOR...NEXT Command

The FOR...NEXT command must be stored in a subroutine. The subroutine is then called (see CALL) to execute the command. All commands between the FOR and NEXT commands will be executed for each pass through the loop.

Since the FOR statement must be entered via HP-IB, we suggest that all non-trivial subroutines be stored in the controller so the subroutine may be edited in the system controller and sent to the HP 3852A again, if necessary.

FOR...NEXT Loop Operation

1. The *loop__counter* is set equal to the *initial__value* when the loop is entered. Each time the corresponding NEXT statement is reached, the *step__size* (which defaults to 1) is added to the *loop__counter* value, and the new value is tested against the *final__value*.

2. If the *final__value* has not been passed in the direction of the STEP, the loop is executed again, beginning with the command immediately following the FOR statement. If the *final__value* has been passed, program execution continues at the command following the NEXT statement.

3. The *loop__counter* value is also tested against the *final__value* when the values are assigned (when the loop is first entered). If the *loop__counter* value has passed the *final__value* in the direction of the STEP, the loop is not executed.

4. The *initial__value*, *final__value*, and *step__size* values are calculated when the loop is entered and are used while the loop is repeating. If a variable or expression is used for any of these values, its value may be changed after entering the loop without affecting how many times the loop is repeated. However, changing the value of the *loop__counter* can affect how many times the loop is repeated.

5. The *loop__counter* variable is allowed in expressions that determine the *initial__value*, *final__value*, or *step__size* values. The previous value of the *loop__counter* is changed until after the *initial__value*, *final__value*, and *step__size* values are calculated. If the *step__size* value evaluates to 0, the loop repeats indefinitely, but no error is given.

Data Returned

None. However, commands between FOR and NEXT may return data.

Related Commands

SUB, SUBEND, WHILE...END WHILE

Examples

Example: FOR...NEXT Loop - Step by -.1 Increments

This program sets up a FOR...NEXT loop with an *initial__value* = 4, a *final__value* = 0, and *step__size* = -.1. The FOR...NEXT loop is in subroutine FORSQR which is called with the CALL statement in line 110.

```
 10 OUTPUT 709;"RST"                   !Reset the HP 3852A
 20 OUTPUT 709;"REAL R"                !Define loop_counter = R
 30 OUTPUT 709;"SUB FORSQR"            !Start subroutine
 40 OUTPUT 709;"FOR R=4 TO 0 STEP -.1" !Set inital,final,step size
 50 OUTPUT 709;"DISP R"                !Display value of R
 60 OUTPUT 709;"WAIT 1"                !Wait 1 second
 70 OUTPUT 709;"DISP SQR(R)"           !Display square root of R
 80 OUTPUT 709;"WAIT 1"                !Wait 1 second
 90 OUTPUT 709;"NEXT R"                !Repeat loop if R >=0
100 OUTPUT 709;"SUBEND"                !End subroutine
110 OUTPUT 709;"CALL FORSQR"           !Call subroutine
120 END
```

FRACT

• **Math Function**

Description

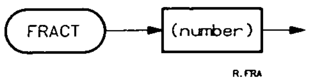
Math function. Returns the fractional part of the value of the argument. For all X, X = INT(X) + FRACT(X).

Prerequisites

The FRACT function is only available with mainframe firmware revision 3.0 or greater.

Syntax

FRACT (*number*)



Parameters

number Number or numeric expression.

Remarks

Data Returned

The data returned is a number greater than or equal to 0 but less than 1, which is the fractional part of the number or expression specified. The result is the same type (REAL or INTEGER) as the argument.

Related Commands

None

Examples

Example: Returning the Fractional Part of the Square Root of 2

This example demonstrates the use of the FRACT command by returning the fractional part of the square root of 2.

```
10   OUTPUT 709;"VREAD FRACT (SQR(2))"
20   ENTER 709;A
30   PRINT A
40   END
```

When this program executes, the following is displayed:

```
.4142136
```

- **HP 44701A Integrating Voltmeter**
- **HP 44702A/B High-Speed Voltmeter**

Description

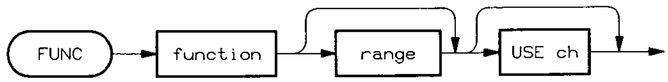
Select specified voltmeter function and measurement range.

Prerequisites

None

Syntax

FUNC *function* [*range*] [USE *ch*]



rr_55

Parameters

function

Measurement function for HP 44701A Integrating Voltmeter or for HP 44702A/B High-Speed Voltmeter. Power-on *function* = DCV.

HP 44701A Integrating Voltmeter

function	Definition
DCV	DC voltage measurements
ACV	AC voltage measurements
OHMF	4-wire ohms measurements

HP 44702A/B High-Speed Voltmeter

function	Definition
DCV	DC voltage measurements
OHMF10K	4-wire ohms up to 10k ohms
OHMF100K	4-wire ohms up to 100k ohms
OHMF1M	4-wire ohms up to 1M ohms

range

Selects a measurement range or the autorange mode. To select a measurement range, specify the maximum expected signal amplitude (or maximum expected resistance) to be measured. The voltmeter then selects the correct range.

To select autorange, use the word AUTO or the number 0 for the *range* parameter. In autorange mode, the voltmeter samples the input signal and selects the appropriate range before each measurement. Power-on and default *range* = AUTO.

HP 44701A Integrating Voltmeter

DC Voltage		AC Voltage		4-Wire Ohms	
range	Sets Range	range	Sets Range	range	Sets Range
0 or AUTO	Autorange	0 or AUTO	Autorange	0 or AUTO	Autorange
>0 - .03	30 mV	>0 -.2	200 mV	>0 - 30	30 Ω
>.03 - .3	300 mV	>.2 - 2	2V	>30 - 300	300 Ω
>.3 - 3	3V	>2 - 20	20V	>300 - 3E3	3 kΩ
>3 - 30	30V	>20 - 200	200V	>3E3 - 3E4	30 kΩ
>30 - 300	300V			>3E4 - 3E5	300 kΩ
				>3E5 - 3E6	3 MΩ

HP 44702A/B High-Speed Voltmeter

DC voltage		4-wire ohms up to 10 kΩ	
range	Selects Range	range	Selects Range
0 or AUTO	Autorange	0 or AUTO	Autorange
>0 - .040	40 mV	>0 - 40	40 Ω
>.04 - .32	320 mV	>40 - 320	320 Ω
>.32 - 2.56	2.56V	>320 - 2.56E3	2.56k Ω
>2.56 - 10.24	10.24V	>2.56E3 - 10.24E3	10.24k Ω

4-wire ohms up to 100 kΩ		4-wire ohms up to 1M Ω	
range	Selects Range	range	Selects Range
0 or AUTO	Autorange	0 or AUTO	Autorange
>0 - 400	400 Ω	>0 - 4E3	4k Ω
>400 - 3.2E3	3.2k Ω	>4E3 - 32E3	32k Ω
>3.2E3 - 25.6E3	25.6k Ω	>32E3 - 256E3	256k Ω
>25.6E3 - 102.4E3	102.4k Ω	>256E3 - 1.024E6	1.024M Ω

USE *ch* Voltmeter slot number. See Glossary.

Remarks

Data Returned

None.

Related Commands

ARANGE, RANGE, USE

Examples

Example: Selecting 4-wire Ohms Function

This program performs a 4-wire ohms measurement on channel 1 of the HP 44705A multiplexer accessory installed in slot 0. The FUNC command is used to set the function and range of the HP 44701A voltmeter in slot 1. For this program, the maximum expected resistance is 1 kΩ.

```
 10 OUTPUT 709;"RST"               !Reset the HP 3852A and HP 44701A
 20 OUTPUT 709;"USE 100"           !Use voltmeter in mainframe slot 1
 30 OUTPUT 709;"FUNC OHMF,1E3"     !Select 4-wire ohms, 1k ohm max input
 40 OUTPUT 709;"TERM BOTH"         !Set backplane as the vm input
 50 OUTPUT 709;"CLOSE 1,11,91,94"  !Close channels and tree relays
 60 OUTPUT 709;"TRIG SGL"          !Trigger once
 70 OUTPUT 709;"CHREAD 100"        !Retrieve reading from voltmeter
 80 OUTPUT 709;"OPEN 1,11,91,94"   !Open channels and tree relays
 90 ENTER 709;A                    !Enter reading into controller
100 PRINT A                        !Display reading
110 END
```

• HP 44715A 5-Channel Counter/Totalizer

Description

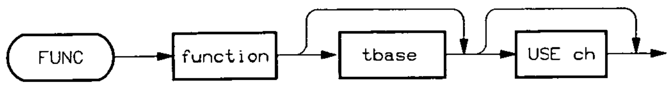
Set counter channel function.

Prerequisites

Double-input functions (all functions except Ungated Total Counts (TOTAL) and Ungated Total Counts, Modulo NPER (TOTALM)) can only be programmed on channels configured for double inputs. FUNC command does not apply to channels set for Frequency configuration.

Syntax

FUNC *function* [*tbase*] [**USE** *ch*]



Parameters

function

Set counter channel function. Sets channel to one of nine functions, as shown in the following table, where *function* = parameter of the FUNC command, D = double-input function, and S = single-input function. Refer to the NPER command for details on Modulo NPER functions. At power-on, *function* = TOTAL.

Function	function	Ch	Description	Application
Ungated Total Counts	TOTAL	S	Totalizes A input.	Total counts on single input.
Gated Total Counts	TOTAL	D	Totalizes A input, gated by B input.	Total counts on single input, gated by second input.
Ungated Total Counts, Mod NPER	TOTALM	S	Totalizes A input, count modulo NPER.	Total counts on single input, modulo NPER.
Gated Total Counts, Mod NPER	TOTALM	D	Totalizes A input, gated by B input, count modulo NPER.	Total counts on single input, gated by second input, count modulo NPER.

Function	function	Ch	Description	Application
Up/Down Counts	UDC	D	Count up on A input, count down on B input. Result is (A-B) counts.	Count difference between counts of two inputs.
Up/Down Counts, Mod NPER	UDCM	D	Count up on A input, count down on B input. Result is (A-B) counts, modulo NPER.	Count difference between counts of two inputs, modulo NPER.
Counts/ Direction	CD	D	Count A input up or down. B input controls direction.	Count relative number of up counts and down counts.
Counts/ Direction, Mod NPER	CDM	D	Count A input up or down. B input controls direction. Count modulo NPER.	Count relative number of up counts and down counts, modulo NPER.
Ratio	RAT	D	Count A input until NPER periods to B input occur.	Measure ratio of A input to B input counts.
Period	PER	D	Measure average of NPER periods of A input.	Measure average value of NPER periods of input
Delayed Period	PERD	D	Measure NPERth period of A input, as gated by B input.	Measure value of single period of input, delayed by NPER periods.

tbase Period of the HP 44715A internal clock which is counted during the time it takes the number of periods of the input signal supplied by the user to occur (NPER). *tbase* values are 1 us, 10 us, 100us, 1 ms, and 10 ms. The default setting for *tbase* is AUTO. *tbase* is valid for the PER and PERD functions only. If necessary, also see the TBASE command.

USE *ch* Specify channel to be used for FUNC command. Channel number range (depends on HP 44715A hardware configuration) = ES00 through ES04.

Remarks

Power-On Function Settings

When the card configuration jumper is set to FREQ position, all five channels simultaneously measure frequency and no other functions can be set. With the jumper in this position, the FUNC command may not be used.

For other settings of the card configuration jumper (TOTAL, 3-CH, or 4-CH), the power-on function is TOTAL for both single-input and double-input channels.

Data Returned

None

Related Commands

CHREAD, CHREADZ, CNTSET, DISABLE INTR, EDGE, ENABLE INTR, NPER, SPER, TBASE, TERM, TRIG, USE, XRDGS

Examples

Example: Set Channel Function

This program counts the number of positive edges that occur on a counter channel during a 10 second period. The reading is then stored in a mainframe variable.

```
10 OUTPUT 709;"INTEGER CNTS"           !Declare mainframe variable
20 OUTPUT 709;"USE 100"                !Use ch 0 in slot 1
30 OUTPUT 709;"EDGE LH"                !Count positive edges
40 OUTPUT 709;"FUNC TOTAL"             !Set totalizing function
50 OUTPUT 709;"TRIG SGL"               !Trigger counter to start counting
60 WAIT 10                             !Halt program execution for 10 sec
70 OUTPUT 709;"CHREAD 100 INTO CNTS"   !Store reading in mainframe memory
80 END
```

- **HP 44730A Track/Hold with Signal Conditioning**
- **HP 44732A 120 Ohm Dynamic Strain Gage FET Multiplexer**
- **HP 44733A 350 Ohm Dynamic Strain Gage FET Multiplexer**

Description

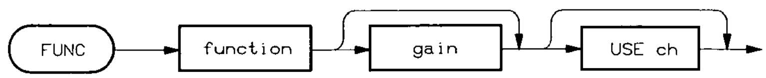
Sets the function on a specified channel.

Prerequisites

Command requires mainframe firmware revision 3.5 or greater. The system must not be scanning with the HP 44702A/B voltmeter when FUNC is executed.

Syntax

FUNC *function* [*gain*] [USE *ch*]



Parameters

function

Set channel specified by USE *ch* to one of the following functions. Power-on/reset *function* = AMPLIFY.

function	Description
AMPLIFY	Amplify the input signal with gain = 1, 10, or 100.
SAMPLE	Sample the input signal.
POSPEAK	Detect positive peak of the input.
NEGPEAK	Detect negative peak of the input.
CALHI	Set channel to measure input HIGH level (used with CAL).
CALLO	Set channel to measure input LOW level (used with CAL).

gain

Set channel gain to 1, 10, or 100. Range of *gain* is $\geq$0 to $\leq$100. Values are rounded <u>up</u> to the next allowable value (1, 10, or 100). Default *gain* = 1.

USE *ch*

Specify channel to be used for the FUNC command. Channel number range = ES00-ES03.

Remarks

Setting Function after CAL

After channel calibration (using FUNC CALHI and FUNC CALLO), the channel function must be reset to AMPLIFY, SAMPLE, POSPEAK, or NEGPEAK as required.

Gain Parameter Required for FUNC CALLO and FUNC CALHI

When using FUNC CALLO and FUNC CALHI for gain calibration, the *gain* parameter must be specified and the input voltage must be within a range such that the (CAL *true*)* (channel gain) is between 9.00 Vdc and 10.24 Vdc.

Data Returned

None.

Related Commands

CAL, GAIN

Examples

Example: Sampling Measurements

This program samples the input to channel 500 of an HP 44730A once 0.1 sec for 1 second (10 samples). The program returns the value at each sample point and computes the maximum, minimum, and mean values and standard deviation of the input during the 1-second sample period. Channel 500 of the HP 44730A is set for sampling mode with filtering off and channel gain of 10. The HP 44702A/B voltmeter is set for 10 prescans, no postscans, one reading per channel, and internal scan, measure, and stop triggers.

NOTE

To illustrate the programming concept, the sample rate shown is much slower than typical sample rates. Sample rates to 100 kHz (with SCDELAY 0,10E-6) are available.

```
10   !
20   !Define arrays and variables
30   !
40   DIM E(0:9)                          !Define controller array
50   OUTPUT 709;"REAL L,H,M,S"           !Define statistics variables
60   OUTPUT 709;"REAL Vlts(9)"           !Define HP 3852A REAL array
70   !
80   !Set up HP 44730A Accessory
90   !
100  OUTPUT 709;"USE 500"                !HP 44730A use ch is 500
110  OUTPUT 709;"FUNC SAMPLE,10"         !Set sampling w/gain of 10
120  OUTPUT 709;"TRIG RIBBON"            !Trigger from ribbon cable
130  !
140  !Set up HP 44702A/B Voltmeter
150  !
160  OUTPUT 709;"USE 600"                !Use voltmeter in mf slot 6
170  OUTPUT 709;"SCANMODE ON"            !Set Scanner Mode
180  OUTPUT 709;"CONF DCV"               !Conf for DC volts
190  OUTPUT 709;"ARMODE BEFORE"          !Immediate autorange mode
200  OUTPUT 709;"SCDELAY 0,.1"           !0.1 sec between scan trigs
210  OUTPUT 709;"PRESCAN 10"             !Make 10 scan passes
220  OUTPUT 709;"CLWRITE 500"            !Set ch 500 as scan list
230  !
240  !Make measurements
250  !
260  OUTPUT 709;"SCTRIG INT"             !Internal scan trig source
270  OUTPUT 709;"XRDGS 500 INTO Vlts"    !Trans rdgs to mainframe array
280  OUTPUT 709;"STAT L,H,M,S,Vlts"      !Compute statistics
```

```
290  !
300  !Display statistics
310  !
320  OUTPUT 709;"VREAD L"                   !Read low value
330  ENTER 709;A                            !Enter low value
340  PRINT "Low Value  =";A;"Volts"         !Display low value
350  OUTPUT 709;"VREAD H"                   !Read high value
360  ENTER 709;B                            !Enter high value
370  PRINT "High Value =";B;"Volts"         !Display high value
380  OUTPUT 709;"VREAD M"                   !Read mean value
390  ENTER 709;C                            !Enter mean value
400  PRINT "Mean Value =";C;"Volts"         !Display mean value
410  OUTPUT 709;"VREAD S"                   !Read std deviation
420  ENTER 709;D                            !Enter std deviation
430  PRINT "Stand Dev  =";D;"Volts"         !Display std deviation
440  PRINT                                  !Space
450  PRINT "Reading    Volts"               !Display Heading
460  PRINT                                  !Space
470  !
480  !Display input values
490  !
500  OUTPUT 709;"VREAD Vlts"                !Trans rdgs to out buffer
510  ENTER 709;E(*)                         !Enter rdgs
520  FOR I=0 TO 9                           !Start print loop
530    PRINT I+1,E(I)                       !Display readings values
540  NEXT I                                 !Increment print loop
550  END
```

Since the channel gain is 10, a typical return for a varying input of about 0.5 V is:

```
Low Value  = 4.2 Volts
High Value = 5.6775 Volts
Avg Value  = 4.8225 Volts
Stand Dev  = .5314132 Volts

 Reading     Volts

  1          5.665
  2          5.6675
  .            .            10 readings
  .            .
  10         4.2
```

GAIN

- **HP 44730A Track/Hold with Signal Conditioning**
- **HP 44732A 120 Ohm Dynamic Strain Gage FET Multiplexer**
- **HP 44733A 350 Ohm Dynamic Strain Gage FET Multiplexer**

Description

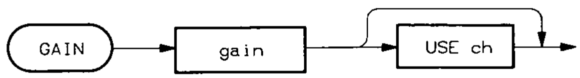
Sets the gain on a specified channel.

Prerequisites

Command requires mainframe firmware revision 3.5 or greater. The system must not be scanning with the HP 44702A/B voltmeter when GAIN is executed.

Syntax

GAIN *gain* [USE *ch*]



Parameters

gain — Set channel gain to 1, 10, or 100. Range of *gain* is $\geq$0 to $\leq$100. Values are rounded <u>up</u> to the next allowable gain (1, 10, or 100). Power-on/reset *gain* = 1.

USE *ch* — Specify channel to be used for the GAIN command. Channel number range = ES00 through ES03.

Remarks

Data Returned

None.

Related Commands

FILTER, FUNC

Examples

Example: Setting Channel Gain

This program measures the input to channel 500 of an HP 44730A. The channel is set for amplification, gain of 10, and no filtering. Measurements are made with an HP 44702A/B voltmeter in slots 6 and 7 via the ribbon cable.

```
10  OUTPUT 709;"FUNC AMPLIFY,USE 500"    !Ch 500 to amplify
20  OUTPUT 709;"GAIN 10,USE 500"         !Ch 500 gain = 10
30  OUTPUT 709;"SCANMODE ON,USE 600"     !Set Scanner mode
40  OUTPUT 709;"CONFMEAS DCV,500"        !Conf/meas DC volts, ch 500
50  ENTER 709;A                          !Enter reading
60  PRINT A                              !Display reading
70  END
```

For a 0.5 V, 5% input, a typical return is:

```
4.995
```

• Mainframe

Description

HP 3852A Group Execute Trigger. Causes a backplane (system) trigger if the system trigger command (TRG) is set to TRG GET.

Prerequisites

TRG *source* must be set to TRG GET (no action if TRG GET not set).

Syntax

GET

RR_57

Parameters

None

Remarks

GET is Equivalent to Group Execute Trigger

The GET command is equivalent to a Group Execute Trigger received over HP-IB. However, the GET command is not recognized until all previously entered commands have been executed, while the HP-IB command Group Execute Trigger will be done "immediately", regardless of previous commands entered.

Related Commands

TRG

Examples

Example: Setting GET Command

The following program lines set the HP 3852A to execute the command GET. Note, however, that the command will not be executed until all previously entered commands have executed.

```
           .
           .
50 OUTPUT 709;"TRG GET"  !Set trig source to HP 3852A Group Execute Trigger
60 OUTPUT 709;"GET"      !Send backplane (system) trigger
           .
           .
```

HALT

- **HP 44714A 3-Channel Stepper Motor Controller/Pulse Output**

Description

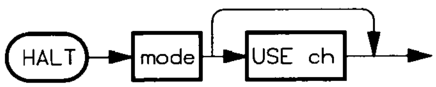
Specifies the conditions under which a stepper motor is to be brought to an abrupt stop. For general purpose pulse output, specifies conditions for stopping pulses with a HALTn input.

Prerequisites

1. Requires mainframe firmware revision 3.0 or greater.
2. A halt switch must be connected to the HALTn input to be monitored by the channel used.

Syntax

HALT *mode* [USE *ch*]



Parameters

mode The *mode* parameter specifies whether a high or low state initiates the halt, whether the HALTn input is disabled or enabled, and if an immediate halt is to be activated.

mode	Definition
OFF	The OFF parameter is used to disable the HALTn input. The HALTn input is not recognized when the OFF mode is active and a motor cannot be stopped with this input. Pulses continue to be generated by the HP 44714A.
ON	Enables the HALTn input after it has been disabled with the OFF parameter. When enabled, the HALTn input is either active low or active high as previously set.
HI	Sets the HALTn input to initiate a halt when a high is present on the input.
LO (power-on)	Sets the HALTn input to initiate a halt when a low is present on the input. This is the power-on mode.
SGL	Initiates an immediate halt. After a SGL halt is generated, HALT returns to its previous setting of OFF, HI, or LO. If a mode is not specified in the command, the default is SGL.

USE *ch* Channel that is to be halted. Channel range can be ES00 to ES02.

Remarks

Power-On State

The power-on state for the HALTn input is LO.

Data Returned

None

Related Commands

None

Examples

Example: Initiate an Immediate Halt With No Deceleration

```
10 OUTPUT 709;"HALT USE 201"          !Set ch 1 in slot 2
                                       !Halt SGL
20 END
```

HARDLIM

- **HP 44714A 3-Channel Stepper Motor Controller/Pulse Output**

Description

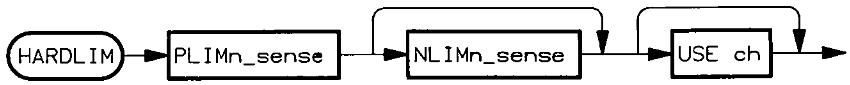
Determines how the positive (PLIMn) and negative (NLIMn) hard limit inputs will operate. When a limit switch is activated in the positive direction, further positive motion is prohibited but negative motion is allowed. The opposite case occurs when the negative limit switch is activated.

Prerequisites

1. Requires mainframe firmware revision 3.0 or greater.
2. Must have limit switch inputs connected to the PLIMn and NLIMn inputs.

Syntax

HARDLIM *PLIMn_sense* [*NLIMn_sense*] [USE *ch*]



RR.HAR

Parameters

PLIMn_sense

The *PLIMn_sense* parameter specifies whether the positive limit is recognized by a high or low state. This parameter can also be used to turn the hard limit off so it is not recognized and to turn it on again with the same HI/LO sense previously active. The power-on state is LO.

PLIMn sense	Definition
OFF	The positive hard limit can be disabled with the OFF parameter. When off, there is no positive limit.
HI	A high occurring on the PLIMn input indicates the positive motion limit has been reached. No further motion in the positive direction can take place but motion in the negative direction is allowed.
LO (power-on)	A low occurring on the PLIMn input indicates the positive motion limit has been reached. No further motion in the positive direction can take place but motion in the negative direction is allowed.
ON	Causes a positive hard limit that has been disabled to be enabled with the same HI/LO state declared before the limit was disabled. If the state has not previously been set to HI, the power-on hard limit state of LO is activated.

NLIMn_sense (defaults to PSTPn_sense)

The *NLIMn_sense* parameter controls the negative hard limit the same way the *PLIMn_sense* parameter controls the positive hard limit. If the *NLIMn_sense* parameter is not specified, it defaults to the *PLIMn_sense* setting. The power-on state is LO.

USE *ch* Channel that the hard limits apply to. Channel number range can be ES00 to ES02.

Remarks

Power-On State

The power-on state for the HARDLIM command is LO, LO. The PLIMn and NLIMn inputs will activate a limit when a low is detected.

Data Returned

None

Related Commands

None

Examples

Example: Set Channel 2 Hard Limit to Immediately Terminate Pulses On a Low

```
10 OUTPUT 709;"HARDLIM LO USE 202"      !Set ch 2 in slot 2 hard limit
20                                      !PLIM_sense = LO
30 END                                  !NLIM_sense default to LO
```

ID?

• **Mainframe**

Description

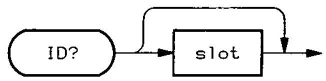
Identify the HP 3852A mainframe or plug-in accessory. ID? returns HP3852A or the accessory number of the plug-in accessory in the slot addressed.

Prerequisites

None

Syntax

ID? [*slot*]



Parameters

slot Slot number addressed. See Glossary. If *slot* is not specified, HP 3852A is returned.

Remarks

Special Considerations

If a plug-in accessory is installed but the terminal module is not connected or if the terminal module is connected AFTER power is applied to the HP 3852A, 447XXX is returned. In this case, you must reset the HP 3852A (RST command) or cycle power to return the correct identity.

Data Returned

The ID? command returns the following:

HP3852A	System Identity (default)
000000	No accessory in specified slot
447XXX	Terminal module not connected
44701A	HP 44701A Integrating Voltmeter
44702A	HP 44702A High-Speed Voltmeter
44702B	HP 44702A High-Speed Voltmeter with 64k reading memory
44705	HP 44705A 20-Channel Relay Multiplexer
44705	HP 44705H 20-Channel High-Voltage Relay Multiplexer
44706A	HP 44706A 60-Channel Relay Multiplexer
44708	HP 44708A 20-Channel Relay Multiplexer/TC
44708	HP 44708H 20-Channel High-Voltage Relay Multiplexer/TC
44709A	HP 44709A 20-Channel FET Multiplexer
44710A	HP 44710A 20-Channel FET Multiplexer/TC
44711A	HP 44711A 24-Channel High-Speed FET Multiplexer
44712A	HP 44712A 48-Channel High-Speed FET Multiplexer
44713A	HP 44713A 24-Channel High-Speed FET Multiplexer/TC
44714A	HP 44714A 3-Channel Stepper Motor Controller/Pulse Output
44715A	HP 44715A 5-Channel Counter/Totalizer
44717A	HP 44717A 10-Bridge 120Ω Static Strain Gage Relay Multiplexer
44718A	HP 44718A 10-Bridge 350Ω Static Strain Gage Relay Multiplexer

44719A	HP 44719A 10-Bridge 120Ω Static Strain Gage FET Multiplexer
44720A	HP 44720A 10-Bridge 350Ω Static Strain Gage FET Multiplexer
44721A	HP 44721A 16-Channel Digital Input
44722A	HP 44722A 8-Channel Digital Input
44723A	HP 44723A 16-Channel High-Speed Digital Sense/Control
44724A	HP 44724A 16-Channel Digital Output
44725A	HP 44725A 16-Channel General Purpose Switch
44726A	HP 44726A 2-Channel Arbitrary Waveform DAC
44727X	HP 44727A 4-Channel Voltage DAC
44727X	HP 44727B 4-Channel Current DAC
44727X	HP 44727C 2-Channel Voltage/2-Channel Current DAC
44728A	HP 44728A 8-Channel Relay Actuator
44729A	HP 44729A 8-Channel Power Controller
44730A	HP 44730A 4-Channel Track/Hold with Signal Conditioning
44732A	HP 44732A 4-Bridge 120Ω Dynamic Strain Gage FET Multiplexer
44733A	HP 44733A 4-Bridge 350Ω Dynamic Strain Gage FET Multiplexer
44736A	HP 44736A Breadboard
44788A	HP 44788A HP-IB Controller

Related Commands

IDN?

Examples

Example: Reading HP 3852A Identity

This program shows how to use the ID? command to read the identity of the HP 3852A.

```
10 OUTPUT 709;"ID?"         !Send ID command
20 ENTER 709;Identity$      !Return HP 3852A identity
30 PRINT Identity$          !Display identity
40 END
```

Since the *slot* parameter was not specified, the return is:

```
HP3852A
```

Example: Reading Accessory Identity

This program uses the ID? command to read the identity of a plug-in accessory in slot 1 of the mainframe.

```
10  OUTPUT 709;"ID? 100"    !Determine identity of accessory in slot 1
20  ENTER 709;Identity$     !Enter identity
30  PRINT Identity$         !Display identity
40  END
```

If an HP 44705A 20-Channel Relay Multiplexer is installed in slot 1 (and the terminal module is connected), the return is:

```
44705
```

IDN?

• **Mainframe**

Description

Query the identity of the HP 3852A. IDN? returns four constants which identify the HP 3852A model number and firmware revision.

Prerequisites

None

Syntax

IDN?

Parameters

None

Remarks

Entering IDN? From the HP 3852A Front Panel

When entering IDN? from the HP 3852A front panel keyboard, use "FASTDISP OFF" to see each constant. (Refer to FASTDISP command for information on slowing down the display.)

Data Returned

IDN? returns the sequence shown. (The firmware revision number may be different than 2.1.)

```
HEWLETT PACKARD          (Company name)
3852A                    (Model number)
0                        (Serial number unknown)
2.1                      (Firmware revision 2.1)
```

Related Commands

ID?

Examples

Example: Identifying the HP 3852A

This program uses IDN? to identify the HP 3852A model number and firmware revision.

```
10 DIM Identity$(0:3)[17]          !Define array Identity
20 OUTPUT 709; "IDN?"              !Query HP 3852A
30 ENTER 709; Identity$(*)         !Enter model number, firmware rev
40 PRINT USING "K,/";Identity$(*)  !Display number, rev
50 END
```

For firmware revision 2.1, a typical display is:

```
HEWLETT PACKARD
3852A
0
2.1
```

IF...END IF

- **Mainframe**

Description

This command provides conditional branching within an HP 3852A subroutine.

Prerequisites

The IF...END IF construct must be stored within an HP 3852A subroutine.

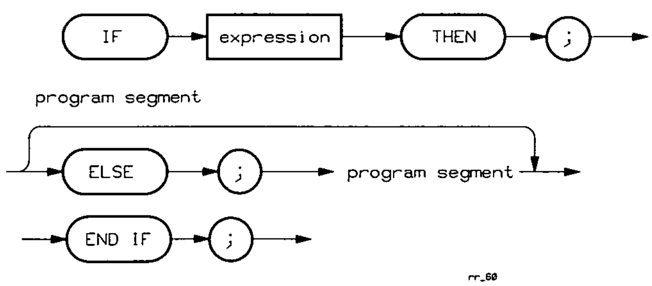
Syntax

IF *expression* **THEN**

program segment

[**ELSE** program segment]

END IF



Parameters

expression A boolean expression evaluated as TRUE if non-zero or evaluated as FALSE if zero. OR or AND numeric expressions can be combined with >, <, =, <=, >=, or <> to form boolean expressions.

Remarks

<u>IF...END IF Operation</u>

1. The IF...THEN, ELSE, and END IF statements must be stored in an HP 3852A subroutine and the subroutine must be called (CALL command) to execute the commands.

2. The END IF statement must follow the IF...THEN command somewhere in the subroutine. ELSE is an optional statement, but if used must appear before the END IF statement. All commands after the IF...THEN statement and before the ELSE or END IF statements will be executed if the expression evaluates to TRUE.

3. When ELSE is specified, only one of the the two program segments will be executed. When the condition is TRUE, the segment between IF...THEN and ELSE is executed. When the condition is FALSE, the segment between ELSE and END IF is executed. In either case, when the construct is exited, program execution continues with the statement after the END IF.

4. When using relational operators (>, <, =, <=, >=, or <>) in the IF expression, the IF statement must be entered over HP-IB. All non-trivial subroutines should be stored via HP-IB so the subroutine may be edited in the controller and re-sent to the HP 3852 if necessary.

Data Returned

None (however, commands in program segments may return data).

Related Commands

SUB, SUBEND

Examples

Example: IF...END IF Construct

This program uses an IF...END IF construct within HP 3852A subroutine IFSQR to display the square root of a number or to indicate that the number is <0. As the program executes the first pass through the subroutine, the the front panel display shows DISP in the left window and 2.000000E+00 in the right window.

For the second pass through the subroutine, since the number is <0, the HP 3852A BEEPS once and displays SUBEND in the left window and NUMBER <0 in the right window. (Note that FASTDISP OFF is set so that you can see both results on the display.)

```
 10 OUTPUT 709;"RST"                !Reset the HP 3852A
 20 OUTPUT 709;"FASTDISP OFF"       !Turn fast display off
 30 OUTPUT 709;"REAL R"             !Define R as REAL variable
 40 OUTPUT 709;"SUB IFSQR"          !Define subroutine IFSQR
 50 OUTPUT 709;"IF R<0 THEN"        !Start IF...THEN construct
 60 OUTPUT 709;"BEEP"               !BEEP once
 70 OUTPUT 709;"DISP 'NUMBER <0'"   !Display Number <0 for R<0
 80 OUTPUT 709;"ELSE"               !Do 80 - 100 only if R >=0
 90 OUTPUT 709;"DISP SQR(R)"        !Display square root of R
100 OUTPUT 709;"END IF"             !End IF...THEN construct
110 OUTPUT 709;"SUBEND"             !End subroutine
120 OUTPUT 709;"LET R=4"            !Set R = 4
130 OUTPUT 709;"CALL IFSQR"         !Call subroutine IFSQR
140 OUTPUT 709;"LET R = -1"         !Set R = -1
150 OUTPUT 709;"CALL IFSQR"         !Call subroutine IFSQR again
160 END
```

INBUF

- **Mainframe**

Description

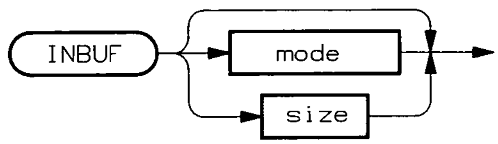
Enables/disables command input buffering and sets the size of the input buffer.

Prerequisites

The *size* parameter of the INBUF command is only available with mainframe firmware revision 3.0 or greater.

Syntax

INBUF [*mode*] or [*size*]



R. INB

Parameters

mode Enables or disables the mainframe's input buffer. Power-on *mode* = OFF. Default *mode* = ON.

mode	Definition
OFF	Disables the input buffer. When disabled, the HP-IB is held off after the end of each command until the command has been executed (or stored if a subroutine entry).
ON	Enables the input buffer. When enabled, the input buffer stores multiple commands sent over the HP-IB while executing the current command. In this mode, HP-IB is not held off after each command. When the buffer is full, no new bytes are accepted until the mainframe processes an input byte, thus making space available in the buffer.

size Sets the size of the input buffer. The minimum *size* that can be specified is 5 bytes. The maximum *size* depends on the amount of mainframe memory available. Once a size has been entered with the INBUF command, the HP 3852A must be reset in order to load the value into the operating system. Cycling power sets the buffer size to 198 bytes.

Remarks

Using Status Register RDY Bit

When the input buffer is enabled, commands sent from the controller may be stored in the buffer waiting to process. You can use the RDY bit (bit 4) in the status register and the serial poll response to determine when commands stored in the buffer have completed. The RDY bit goes true when all commands in the buffer have been executed.

Setting the Input Buffer Size

The size of the input buffer set by the INBUF command does not become effective until the mainframe is reset (RST). The buffer size is stored in volatile memory and will remain the same until changed by subsequent INBUF (and RST) commands or until power is cycled.

Note that a reset erases all previously declared variables, arrays, and downloaded subroutines. Thus, before using any of the mainframe's memory, ensure that the buffer size is set as desired.

Specifying an Input Buffer Size that is Too Large

The memory used to increase the size of the input buffer is taken from the mainframe memory available to the user. If the buffer size specified is extremely large (e.g. INBUF 4200000), ERROR 24: ARGUMENT OUT OF RANGE is reported on execution of the INBUF command. If the size specified is within the argument range but not enough memory is available, ERROR 1: OUT OF MEMORY - HP-IB BUFFERS/SYMTAB is reported during the next reset. When this occurs, the input buffer is set to a default size of 198 bytes, the output buffer is set to 1029 bytes, and the symbol table size (SYMSIZE) is set to accomodate 150 entries.

Input Buffer Size at Power-On

Anytime power is cycled, the size of the input buffer is set to 198 bytes.

Sending A Single Command With INBUF OFF

Typically, HP Series 200/300 output statements send a command string followed by carriage return/line feed (CR/LF). When a single command is sent, the HP 3852A does not hold off further input until the byte after the command terminator (LF in this case). Thus, the output statement will finish and the controller can execute other statements while the HP 3852A is processing the command sent. Valid terminators for the HP 3852A are line feed (LF), semicolon (;), and any byte sent with EOI set.

Data Returned

None

Related Commands

OUTBUF, SYMSIZE

Examples

Example: Monitoring HP 3852A Command Processing

Sometimes, you want to be sure the HP 3852A has finished executing a command before the controller takes further action. Two ways to do this are to wait for ready or to send an extra command terminator, as shown in the following program segments. The first program segment works with INBUF ON or INBUF OFF and monitors the RDY bit in the status register.

```
10 OUTPUT 709;"RST"                    !Reset the HP 3852A
20 OUTPUT 709;"TRG GET"                !Set GET or Group Execute Trigger
30 OUTPUT 709;"CLOSE 100"              !Close channel
40 WHILE NOT BIT (SPOLL (709),4)       !Wait for RDY bit to be set
50 END WHILE                           !Continue
60 TRIGGER 709                         !Trigger voltmeter
              .
              .
```

The second program segment works ONLY with INBUF OFF. In this segment, the HP 3852A sees the semi-colon (;) at the end of the CLOSE command as a command terminator and doesn't accept CR/LF until the command finishes. This keeps the controller on the OUTPUT statement until CLOSE is completed.

```
10 OUTPUT 709;"RST"            !Reset the HP 3852A
20 OUTPUT 709;"CONF DCV"       !Set DC voltage measurements
30 OUTPUT 709;"TRG GET"        !Set GET or Group Execute Trigger
40 OUTPUT 709;"CLOSE 100;"     !Semi-colon (;) acts as command terminator
50 TRIGGER 709                 !Won't be executed until CLOSE completes
              .
              .
```

Example: Setting the Input Buffer Size

A typical application for increasing the size of the input buffer is prior to downloading a long subroutine which ordinarily would fill the buffer and hold up the HP-IB.

```
10 OUTPUT 709;"INBUF 300"          !Set buffer size to 300 bytes
20 OUTPUT 709;"RST"                !Load buffer size into the system
30 OUTPUT 709;"INBUF ON"           !Enable input buffering
40 OUTPUT 709;"SUB A"              !Begin downloading subroutine A
              .
              .
              .
```

INDEX

• Mainframe

Description

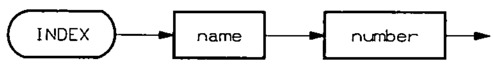
Sets the starting location in an array where the next data element will be stored.

Prerequisites

None

Syntax

INDEX *name number*



RR_62

Parameters

name Name of an array as defined with the DIM, REAL, INTEGER, or PACKED command.

number Starting location (element) in the array where the next reading will be stored. *number* must be ≤ the maximum index of the array.

Remarks

Setting the Array Index

Use INDEX to set the index pointer to any element in the array. When a command enters data into the array, the data is entered beginning with element 0 or the element at which the index pointer was set. For example, if the index pointer of a 10 element array (maximum index is 9) is set at 5, the next reading(s) to be stored will start at element 5 which is the sixth element of the array.

Data Returned

None

Related Commands

INDEX?, REAL, DIM, INTEGER, PACKED

Examples

Example: Setting Array Index Pointer

This program defines a REAL array with 10 elements, sets the index pointer at 5, then stores five readings in the array. Note that the readings will be stored in elements 5, 6, 7, 8, and 9.

```
10 OUTPUT 709;"RST"                          !Reset the HP 3852A
20 OUTPUT 709;"USE 100"                      !Use voltmeter in slot 1
30 OUTPUT 709;"REAL DAT(9)"                  !Declares the array
40 OUTPUT 709;"INDEX DAT 5"                  !Set index pointer to 5
50 OUTPUT 709;"CONFMEAS DCV,0-4,INTO DAT"
60                                           !Store data in elements 5-9
70 END
```

INDEX?

• Mainframe

Description

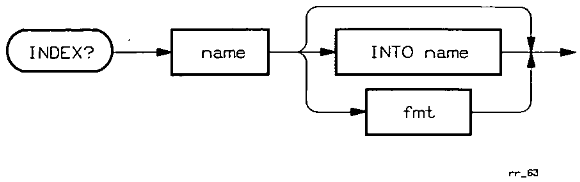
Reads the location of the index pointer in the specified array.

Prerequisites

None

Syntax

INDEX? *name* [INTO *name*] or [*fmt*]



Parameters

name Name of the array whose index is to be read.

INTO *name* See Glossary.

fmt See Glossary. Default format is LASC.

Remarks

Data Returned

INDEX? returns the position of the index pointer for the array specified.

Related Commands

INDEX, REAL, DIM, INTEGER, PACKED

Examples

Example: Using INDEX? Command

In this program, the CONFMEAS command enters five readings into REAL array BUFF. After CONFMEAS is executed, the index pointer is at 5 (sixth element). The index can be moved back to 0 with a VREAD or an INDEX command.

```
10 OUTPUT 709;"RST"                        !Reset the HP 3852A
20 OUTPUT 709;"USE 100"                    !Use voltmeter in slot 1
30 OUTPUT 709;"REAL BUFF (9)"              !BUFF is 10-element array
40 OUTPUT 709;"CONFMEAS DCV,0-4,INTO BUFF" !Store 5 readings in BUFF
50 OUTPUT 709;"INDEX? BUFF"                !Read index pointer position
60 ENTER 709;A                             !Enter pointer position
70 PRINT "Index position in BUFF = ";A     !Display position
80 END
```

A typical display for this program is:

```
Index position in BUFF = 5
```

INITIAL

• **HP 44788A HP-IB Controller**

Description

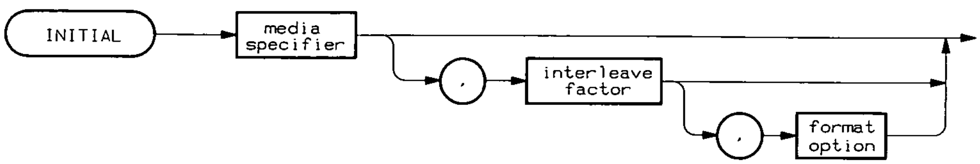
This statement prepares mass storage media for use by the computer or HP 44788A. When INITIAL is executed, **any data on the media is lost.**

Prerequisites

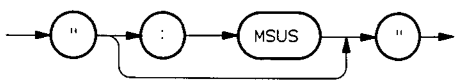
Requires firmware revision 3.5 or greater.

Syntax

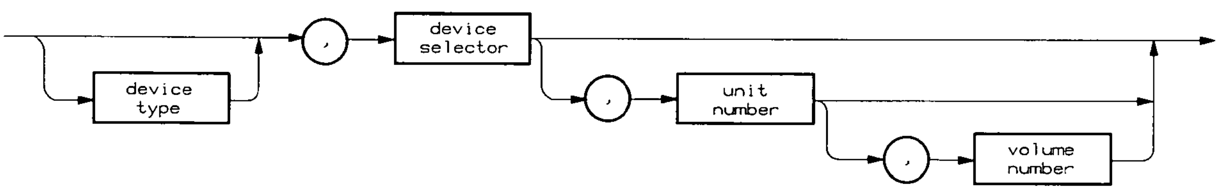
INITIAL "*media specifier*"**[**,*interleave factor* **[**,*format option***]]**



literal form of media specifier



msus:



Parameters

media specifier — The address for the mass storage device.

interleave factor — Establishes the distance in physical records between consecutively numbered records. If 0 is specified for the interleave factor, the default for the device is used.

format option — Allows selection of the format to which the disc is initialized. Omitting the format option or specifying a format option of 0 initializes the disc to its default format.

Remarks

Initializing Media

Any media used by a computer or HP 3852A must be initialized before its **first** use. Initialization rewrites the directory, thus eliminating any access to old data. The media is partitioned into physical records and the quality of the media is checked during initialization. Defective tracks are "spared" (marked so that they will not be used).

NOTE

*All open files on the destination device/unit/volume must be closed by an ASSIGN @I/O path TO * statement before initialization can take place.*

Specifying the Format Option

Some mass storage devices allow you to select the format to which the disc is initialized. Omitting the format option or specifying a format option of 0 initializes the disc to its default format. (Refer to the disc drive manual for format options available with your disc drive.) For example, when initializing a single-sided flexible disc on an HP 9122D double-sided flexible disc drive, use format option 4.

Data Returned

None

Related Commands

None

Examples

Example: Initialize Disc Drive

The following statements show two ways to initialize a disc drive with an HP 44788A in slot 2 for disc drive address 00. The first statement is entered from the HP 3852A front panel, while the second statement is entered from a system controller.

```
INITIAL ":,200,1"                    !Initialize disc drive
100 OUTPUT 709;"INITIAL ':,200,1'"   !Initialize disc drive
```

INT

• Math Function

Description

Math function. Returns the largest integer that is less than or equal to the number or expression specified.

Prerequisites

The INT function is only available with mainframe firmware revision 3.0 or greater.

Syntax

INT (*number*)

R. INT

Parameters

number Number or numeric expression.

Remarks

Data Returned

The data returned is the largest integer that is less than or equal to the number or expression specified. This result will be the same type (REAL or INTEGER) as the argument.

Related Commands

None

Examples

Example: Determining the Largest Integer That is Less Than the Sum of Two Real Numbers

This example demonstrates the use of the INT command by returning the largest integer that is less than the sum of two real numbers.

```
10   OUTPUT 709;"VREAD INT(2.2+2.3)"
20   ENTER 709;A
30   PRINT A
40   END
```

When this program runs, the following is displayed:

```
4
```

• **Mainframe**

Description

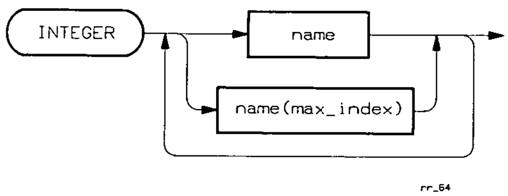
Defines an INTEGER (IN16) variable or array in HP 3852A mainframe memory.

Prerequisites

None

Syntax

INTEGER *name* [*(max_index)*] [*name* [*(max_index)*]...]



Parameters

name Name of the INTEGER variable or array. *name* must be a string <= eight characters. It cannot begin with a digit and must contain only letters, digits, '_', or '?'.

max_index Maximum index (number of elements) of the array. *name* without *max_index* specifies an INTEGER variable, while *name (max_index)* specifies the name and number of elements in the array. Parentheses are required around the *max_index* number. Note that *max_index* = 0 declares a one-element array, not a variable.

Since array indices begin at 0, the number of elements in the array = *max_index* + 1. For example, INTEGER A(9) defines an INTEGER array of 10 elements, with elements 0 through 9.

Remarks

Redeclaring Arrays

Executing INTEGER declares an INTEGER variable or array and fills the elements with zeroes. An INTEGER array if redeclared, must remain an INTEGER array. The redeclared array will be initialized to zeroes in all elements. (If the array is redclared to the same size, it will just be zeroed.)

Arrays and Variables are Global

All arrays and variables are global. Therefore, arrays or variables declared outside a subroutine are accessible in the subroutine and vice-versa.

INTEGER Statement Stored in Subroutine

If an INTEGER statement is stored in a subroutine, the variable or array name is defined immediately but data storage for the array won't be set up until the subroutine is run. (The name can't be used for another subroutine, another array type, or variable.)

Recovering Array Storage Space With DELVAR

Send the DELVAR command to recover storage space allocated by the INTEGER command. The array name and type will remain but the memory allocated to the array is released. Thus, the array becomes an INTEGER array with zero space.

Data Returned

None

Related Commands

REAL, PACKED, DIM, DELVAR

Examples

Example: Declaring Variables and Arrays

In the following program lines, line 100 declares an INTEGER array A of 10 elements, starting with element 0. Since the *max_index* parameter is not used, line 110 declares INTEGER variable B.

```
100 OUTPUT 709;"INTEGER A(9)"   !Declares INTEGER array A with 10 elements
110 OUTPUT 709;"INTEGER B"      !Declares INTEGER variable B
```

• **Mainframe**

Description

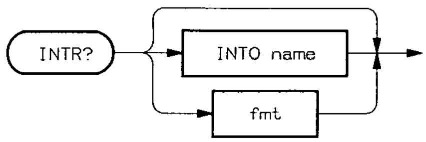
Returns the address of the last channel whose interrupt was serviced. If no interrupt has been serviced since power-on or a system reset, -1 is returned.

Prerequisites

None

Syntax

INTR? [INTO *name*] or [*fmt*]



rr_65

Parameters

INTO *name* See Glossary.

fmt See Glossary. Default format is IASC.

Remarks

Interrupt Priorities

The lowest channel (address) which has been enabled has the highest interrupt priority and will always be serviced first.

Servicing Interrupts

Interrupt service begins when the HP 3852A completes the command (or subroutine) being executed when the interrupt occurred. Interrupt servicing is when the mainframe uses a polling routine to identify the accessory channel which interrupted. When the channel is located, its interrupt capability is then disabled (i.e. DISABLE INTR is performed) and the servicing is complete. The mainframe or controller will then respond to the interrupt.

If an interrupt is cleared before it is serviced, INTR? will indicate that servicing did not occur.

Data Returned

INTR? returns the address of the channel whose interrupt was most recently serviced. If no channel interrupt has been serviced since power-up or the last RST command, -1 is returned.

Related Commands

ENABLE INTR, DISABLE INTR, ON INTR, RQS, STA?

Examples

Using INTR? After an Interrupt Has Been Serviced

This program causes the HP 44701A voltmeter in slot 7 to interrupt and the mainframe to respond by displaying "DATA READY". The mainframe also executes INTR? in response to the interrupt to verify that the interrupt was serviced by the mainframe.

```
 10 OUTPUT 709;"RST"                 !Reset the HP 3852A
 20 OUTPUT 709;"USE 700"             !Use voltmeter in slot 7
 30 OUTPUT 709;"SUB DATA"            !Beginning of subroutine
 40 OUTPUT 709;"DISP 'DATA READY'"   !Define message
 50 OUTPUT 709;"INTR?"               !Determine if interrupt was serviced
 60 OUTPUT 709;"SUBEND"              !End subroutine
 70 OUTPUT 709;"ON INTR CALL DATA"   !Call sub in response to interrupt
 80 OUTPUT 709;"CONF DCV"            !Configure vm for measurement
 90 OUTPUT 709;"TERM EXT"            !Vm input is rear panel terminals
100 OUTPUT 709;"ENABLE INTR"         !Enable channel interrupt capability
110 OUTPUT 709;"ENABLE INTR SYS"     !Enable mainframe to sense interrupt
120 OUTPUT 709;"TRIG SGL"            !Trigger the voltmeter
130 ENTER 709;A                      !Enter results of INTR?
140 PRINT A                          !Display results
150 END
```

As this program completes, the controller displays the address of the channel (slot) whose interrupt was serviced:

```
700
```

• **Mainframe**

Description

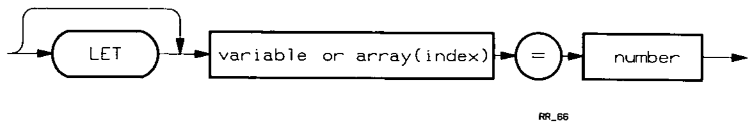
Assigns a value to a variable or to an array element. The value can be a number assigned directly, the result of a numeric expression, or a value copied from one variable or array element to another.

Prerequisites

The specified variable or array must previously be defined using the REAL, DIM, or INTEGER command. [LET] cannot be used with PACKED arrays.

Syntax

[LET] *variable* or *array(index)* = *number*



Parameters

variable Real or Integer variable.

array(index) Specific element of a Real or Integer array.

number Constant, variable, or numeric expression.

Remarks

[LET] Header is Optional

The [LET] header is optional and does not need to be stated when assigning a value to an array element or variable.

Logical Expressions

Firmware revisions 3.5 and greater allow logical expressions (e.g. $B = C$, $B < C$, $B > C$, $B \leq C$, $B \geq C$) to be specified for the *number* parameter. For example:

```
40    OUTPUT 709;"A=(B>C)"
```

will set $A = 1$ if $B > C$, or to 0 if $B \leq C$.

Data Returned

None

Related Commands

REAL, DIM, INTEGER, PACKED, VREAD, VWRITE

Examples

Example: Evaluate Sine of Angle

This program evaluates the sine of 0.223 radians, transfers and displays the result (.2211563) on the controller CRT.

```
10     OUTPUT 709;"RST"                   !Reset the HP 3852A
20     OUTPUT 709;"REAL A"                !Define A as REAL variable
30     OUTPUT 709;"LET A = SIN (0.223)"   !Place sine of 0.223 in A
35 !   OUTPUT 709;"A = SIN (0.223)"       !Valid expression
40     OUTPUT 709;"VREAD A"               !Read value of A into output buffer
50     ENTER 709;A                        !Enter value
60     PRINT A                            !Display value
70     END
```

Example: Copying Data From One Array Element to Another

This program shows how the LET command can be used to copy data from one array element to another.

```
10     OUTPUT 709;"RST"                   !Reset the HP 3852A
20     OUTPUT 709;"INTEGER A(4),B(9)"     !Declare arrays A and B
30     OUTPUT 709;"VWRITE A 1,2,3,4,5"    !Write values to array A
40     OUTPUT 709;"B(4)=A(2)"             !Copy the number in element 2 of A
50                                        !(3) into element 4 of B
60     END
```

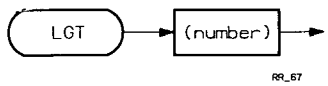
• **Math Function**

Description

Numeric expression evaluated as a command parameter. Returns the logarithm (base 10) of its argument.

Syntax

LGT (*number*)



Parameters

number Number or numeric expression that must evaluate to a range >0.

Remarks

None

Examples

Example: Logarithm, Base 10

The following program computes the logarithm (base 10) of 15 and displays the result (1.176091) on the controller CRT.

```
10 OUTPUT 709;"RST"             !Reset the HP 3852A
20 OUTPUT 709;"VREAD LGT (15)"  !Compute and read log (base 10) of 15
30 ENTER 709;A                  !Enter value in controller
40 PRINT "Log = ";A             !Display value
50 END
```

For this program, a typical display is:

```
Log = 1.176091
```

LMT (post processing)

- **Mainframe**

Description

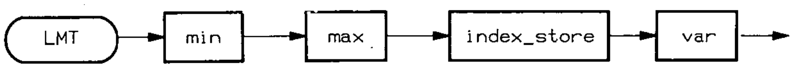
Post processing limit testing. LMT compares readings stored in an array (*var*) to minimum and maximum limits (*min, max*). The indices of any reading out of limits are stored in (*index_store*).

Prerequisites

The *min, max, index_store* and *var* arrays or variables must have been previously defined with a DIM, INTEGER, or REAL command.

Syntax

LMT *min max index_store var*



Parameters

min Name of REAL or INTEGER variable or array containing the lower limit(s).

max Name of REAL or INTEGER variable or array containing the upper limit(s).

index_store Name of REAL or INTEGER array that will store index numbers for readings out of specified limits. The index numbers correspond to the index numbers of the stored readings.

var Name of array containing the stored readings to be compared against the limits.

Remarks

Post Processing LMT vs. Real Time LMT

If *var* is included in the LMT command, LMT is post processing (uses stored readings). If *var* is not included, LMT is real time (uses readings as they are generated). LMT (post processing) is not affected by ENABLE LMT or DISABLE LMT and will not set the LMT bit in the status register.

Single vs. Multiple Limits

If you specify a single value for the upper limit (in the *max* array or variable) and a single value for the lower limit (in the *min* array or variable), every reading stored in the *var* array is compared with those limits.

If you specify more than one upper and lower limit, each reading is compared to its corresponding limits. That is, the first reading is compared to the first set of limits, the second reading to the second set of limits, and so on.

Declaring the *min* and *max* Arrays

If arrays are declared for the minimum and maximum limits, the maximum index of the arrays should be equal to the number of readings you wish to compare (*var* array), and contain a number of limits equivalent to the number of readings.

When an array is declared, each element of the array contains the number 0 until a value (limit) is stored in the array. For example, if you declare the *max* array with a maximum index of 9 (10 elements) and store only five limits, the first five readings in the *var* array will be compared to these corresponding limits. However, elements 5-9 of the *max* array contain 0's and, therefore, the remainder of the readings in the *var* would be compared to the 0's which may be undesirable.

Wraparound for Multiple Limits

The *min* and *max* arrays will wraparound from the last element to the first element. That is, if there are a greater number of readings than limits defined, the first limits are re-used. This allows a small number of limits to be repetitively compared to a larger number of readings.

When the *index_store* array is filled, an error is generated and a wraparound occurs. After the error, out of limit indices overwrite the previous indices.

Index Presetting

Executing the LMT command presets the *min*, *max*, and *var* index pointers to 0 (comparisons start with the first values in the array). LMT, however, does not affect the pointer for the *index_store* array - it resumes where it left off.

Index Position Zero

If no readings are out of limits, the *index_store* array is left unchanged. For example, if seven readings were compared to seven sets of limits and each reading was within its limits, the *index_store* array still contains the seven zeroes assigned to the array when it was first declared.

If, however, the first reading (element 0) was out of limits, the *index_store* array will still contain all zeroes, since an out-of-limit reading in element zero returns 0. Note that the index pointer will have been advanced.

To determine whether the reading in *var* array element 0 is out of limits, you will need to check the index pointer position in the *index_store* array with the INDEX? command. If the pointer is at position 1, the first reading was out of limits. If the pointer is at position 0, the first reading was within limits.

Data Returned

Indices of out of limit readings are stored in the *index_store* array.

Related Commands

LMT (real time), REAL, DIM, INTEGER, VWRITE, INDEX?

Examples

Example: Using LMT (post processing) Command

The following program performs a DC voltage measurement on channels 200-619 of 20-channel relay multiplexers in slots 2 through 6 of the mainframe and stores the readings in the VOLTS array. Then, post processing limit-checking is done on these stored readings.

Since only one set of limits is specified (lower limit = 14.6 volts, upper limit = 16.5 volts), each stored reading is compared against these limits. The index numbers of any readings out of limits are stored in the INDX array.

```
 10 INTEGER Ind (0:99)                          !INTEGER array Ind
 20 OUTPUT 709;"RST"                             !Reset the HP 3852A
 30 OUTPUT 709;"REAL UPPER,LOWER"                !Upper,lower lim arrays
 40 OUTPUT 709;"INTEGER INDX(99)"                !Array to store indices
 50 OUTPUT 709;"REAL VOLTS(99)"                  !Array to store readings
 60 OUTPUT 709;"VWRITE UPPER 16.5"               !Upper limit value
 70 OUTPUT 709;"VWRITE LOWER 14.6"               !Lower limit value
 80 OUTPUT 709;"USE 100"                         !Use voltmeter in slot 1
 90 OUTPUT 709;"CONFMEAS DCV,200-619,INTO VOLTS" !Meas DCV, store in VOLTS
100 OUTPUT 709;"LMT LOWER,UPPER,INDX,VOLTS"      !Execute limit checking
110 OUTPUT 709;"INDEX? INDX"                     !Check index pointer pos
120 ENTER 709;A                                  !Enter pointer position
130 PRINT "No. readings out of limits = ;A       !Disp number rds out of lim
140 IF A<1 THEN 220                              !End if no rds out of lim
150 OUTPUT 709;"VREAD INDX"                      !Read indices
160 ENTER 709;Ind(*)                             !Enter indices
170 PRINT "Indices are: "
180 PRINT
190 FOR I=0 TO A-1
200 PRINT Ind(I)                                 !Print indices
210 NEXT I
220  END
```

If no readings are out of limits, a typical controller CRT display is: "`No. readings out of limits = 0`".

If readings in *var* array elements 10, 23, 49, 55, 57, 68, 76, 80, 90, and 92 (10 readings) are out of limits, a typical controller CRT display is:

```
No. readings out of limits = 10
Indices are:

10
23
49
55
57
68
76
80
90
92
```

• **Mainframe**

Description

Real time limit testing. LMT sets up readings taken to be compared to minimum and maximum limits stored in variables or arrays. The indices of any reading out of limits are then stored. Any reading out of the limits will set the LMT bit in the status register.

Prerequisites

The *min, max, and index_store* arrays or variables must have been previously defined with a DIM, INTEGER, or REAL command. ENABLE LMT must be executed before checking will occur.

Syntax

LMT *min max index_store*

rr_69

Parameters

min — Name of REAL or INTEGER variable or array containing the lower limit(s).

max — Name of REAL or INTEGER variable or array containing the upper limit(s).

index_store — Name of REAL or INTEGER array that will store index numbers for readings out of specified limits. The index numbers correspond to the order of the readings taken in each command returning data.

Remarks

Post Processing LMT vs. Real Time LMT

If *var* is included in the LMT command, LMT is post processing (uses stored readings). If *var* is not included, LMT is real time (uses readings as they are generated). LMT (post processing) is not affected by ENABLE LMT or DISABLE LMT and will not set the LMT bit in the status register.

Single vs. Multiple Limits

If you specify a single value for the upper limit (in the *max* array or variable) and a single value for the lower limit (in the *min* array or variable), every reading is compared with those limits.

If you specify more than one upper and lower limit, each reading is compared to its corresponding limit. That is, the first reading is compared to the first set of limits, the second reading to the the second set of limits, and so on.

Declaring the *min* and *max* Arrays

If arrays are declared for the minimum and maximum limits, the maximum index of the arrays should be equal to the number of readings you wish to compare, and contain a number of limits equivalent to the number of readings.

When an array is declared, each element of the array contains the number 0 until a value (limit) is stored in the array. For example, if you declare the *max* array with a maximum index of 9 (10 elements) and store only five limits, the first five readings will be compared to these corresponding limits. However, elements 5-9 of the *max* array contain 0's and, therefore, any readings which follow would be compared to the 0's which may be undesirable.

Wraparound for Multiple Limits

The *min* and *max* arrays will wraparound from the last element to the first element. That is, if there are a greater number of readings than limits defined, the first limits are re-used. This allows a small number of limits to be repetitively compared to a larger number of readings.

When the *index_store* array is filled, an error is generated and a wraparound occurs. After the error, out of limit indices overwrite the previous indices.

Index Presetting

Executing the LMT command presets the *min* and *max* index pointer to 0 (comparisons start with the first limits in the arrays). LMT, however, does not affect the pointer for the *index_store* array - it resumes where it left off.

If you finish one group of measurements and want to start new comparisons beginning with the first limits in the arrays, you must set the *min* and *max* index pointers to zero by executing another LMT command or using the INDEX command.

Index Position Zero

If no readings are out of limits, the *index_store* array is left unchanged. For example, if seven readings were compared to seven sets of limits and each reading was within its limits, the *index_store* array still contains the seven zeroes assigned to the array when it was first declared.

If, however, the first reading was out of limits, the *index_store* array will still contain all zeroes, since the first reading would return a 0. Note that the index pointer in the *index_store* array will have been advanced.

To determine if the first reading is out of limits, you will need to check the position of the index pointer in the *index_store* array with the INDEX? command. If the pointer is at position 1, the first reading was out of limits. If the pointer is at position 0, the first reading was within limits.

Contents of the *index_store* Array During Testing

During real time limit testing, the contents of the *index_store* array will contain the index of all readings out of limit. Each command's output has a starting index of 0. This means, for example, that if the CONFMEAS command takes 10 measurements, the first measurement has index 0, the second measurement has index 1, and so on. If the real time limit test revealed that the fourth and seventh readings of the 10 readings taken were out of limit, the *index_store* array would contain the indices 3 (index of fourth reading) and 6 (index of the seventh reading).

The *index_store* array must be an array and it must be declared prior to initiating limit testing. If readings from only a single channel are tested, the *index_store* array should be a one element array with a maximum index of 0.

When performing real time limit testing on data returned by several commands, the contents of the *index_store* array should be read prior the execution of each command in order to accurately determine those readings out of limit.

Real Time Limit Testing Must be Enabled

When performing real time limit testing, the actual testing is initiated by the ENABLE LMT command. The LMT command identifies the *min*, *max*, and *index_store* arrays to be used. Through this process you can select the particular readings you want tested against the limits.

Real Time Limit Testing is for Measured Results Only

Real time limit testing can only be used on readings which are the result of a measurement command (i.e. CHREAD, CONFMEAS, MEAS, XRDGS). Real time testing cannot be used with readings generated by TIME, ALRM, VREAD, etc..

Data Returned

Indices of all out of limit readings.

Related Commands

LMT (post processing), REAL, DIM, INTEGER, VWRITE, INDEX?, ENABLE LMT, DISABLE LMT, ENABLE

Examples

Example: Using LMT (real time) Command

The following program performs limit checking of DC voltage measurements on channels 0-2 of a 20-channel relay multiplexer in slot 4 of the mainframe using the voltmeter in slot 1. The first reading is compared against 5.25 (upper limit) and 4.75 (lower limit), the second reading against 16.5 and 14.6, and the third reading against -14.5 and -15.5.

```
 10 OUTPUT 709;"RST"                              !Reset the HP 3852A
 20 INTEGER Ind (0:2)                             !INTEGER array Ind
 30 OUTPUT 709;"REAL UPPER(2),LOWER(2)"           !Upper,lower lim arrays
 40 OUTPUT 709;"INTEGER INDX(2)"                  !Array to store indices
 50 OUTPUT 709;"REAL RD(2)"                       !Array to store readings
 60 OUTPUT 709;"VWRITE UPPER 5.25,16.5,-14.5"     !Upper limit values
 70 OUTPUT 709;"VWRITE LOWER 4.75,14.6,-15.5"     !Lower limit values
 80 OUTPUT 709;"LMT LOWER,UPPER,INDX"             !Limit checking arrays
 90 OUTPUT 709;"ENABLE LMT"                       !Enable limit checking
100 OUTPUT 709;"USE 100"                          !Use voltmeter in slot 1
110 OUTPUT 709;"CONFMEAS DCV,400-402,INTO RD"     !direct rdgs into memory
120 OUTPUT 709;"INDEX? INDX"                      !Check index pointer pos
130 ENTER 709;A                                   !Enter pointer position
140 PRINT "No. readings out of limits = ;A        !Disp number rds out of lim
150 IF A<>0 THEN                                  !disp rdgs out of limit
160   OUTPUT 709;"VREAD INDX"                     !Read indices
170   ENTER 709;Ind(*)                            !Enter indices
180   PRINT "Indices are: "
190   PRINT
200   FOR I=0 TO A-1
210    PRINT Ind(I)                               !Print indices
220   NEXT I
230 END IF
240 END
```

If no readings are out of limits, a typical controller CRT display is:

```
No. readings out of limits = 0
```

If the first and third readings are out of limits, the indices that will be displayed are:

```
No. readings out of limits = 2
Indices are:

0
2
```

• Mainframe

Description

Restores front panel control of the HP 3852A mainframe. If control was restored by pressing the front panel LOCAL key, the LCL bit in the status register is set. (The LCL bit is not set if the LOCAL command is sent over HP-IB.)

Prerequisites

In order to execute the LOCAL command, LOCK OFF must be sent over the HP-IB if the front panel has been disabled by LOCK ON. If the HP-IB is in the remote with lockout state (RWLS), front panel control must be re-established over the HP-IB (e.g. LOCAL 7).

Syntax

LOCAL

RR_70

Parameters

None

Remarks

Commands Executed in Local or Remote States

When the HP 3852A is in the local (LOCS) or local with lockout state (LWLS), all commands can be executed from the front panel. In these states, commands sent over the HP-IB will be accepted but discarded by the HP 3852A if the HP 3852A is not addressed to listen.

When the HP 3852A is in the remote (REMS) or remote with lockout (RWLS) state, only the following commands can be executed from the front panel.

ADDR?	EXTEND?	MONITOR	SUB	WHILE...END WHILE
ALRM	FASTDISP	NEXT	SUBEND	
BEEP	ID?	PAUSE	TIME	
CAT	IDN?	POWEROFF	TIMEDATE	
CLOSE?	IF...END IF	RQS?	USE?	
DISP	INDEX?	SIZE?	VREAD	
ERR?	INTR?	SRQ	WAIT	
ERRSTR?	LOCAL	STATE?	WAITFOR	

HP-IB Remote With Lockout State

If the HP-IB is in the remote with lockout state (RWLS), the HP 3852A LOCAL command (or key) will not restore local (front panel) control. Front panel control must be restored over the HP-IB.

Data Returned

None

Related Commands

LOCK

Examples

Example: Restoring Front Panel Control

The following program segments show how to restore front panel control with the HP-IB in the remote state (REMS), and in the remote with lockout state (RWLS). The syntax shown applies to HP Series 200 and Series 300 controllers. Both examples assume the HP 3852A keyboard is enabled.

HP-IB in Remote State

```
10 OUTPUT 709;"LOCAL"
```

or

```
10 LOCAL 709
```

or

```
10 LOCAL 7
```

Note that sending OUTPUT 709;"LOCAL" is equivalent to pressing the LOCAL key on the front panel. If local control is restored using LOCAL 7, REN (remote enable) must be re-asserted (using REMOTE 7 or REMOTE 709) before the HP 3852A can leave the local (LOCS) state.

HP-IB in Remote With Lockout State

```
10 LOCAL 7
```

This command places the HP-IB into the local state (LOCS) state which restores front panel control. Again, REN (remote enable) must be re-asserted in order to leave LOCS.

```
10 LOCAL 709
```

This command places the HP-IB into the local with lockout state (LWLS) which also enables front panel control.

• Mainframe

Description

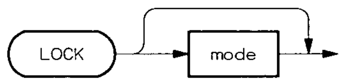
Enables and disables the front panel keyboard.

Prerequisites

None

Syntax

LOCK [*mode*]



rr_71

Parameters

mode Enables or disables front panel keyboard. Power on and default *mode* = OFF.

OFF When the HP 3852A is in local (LOCS) or local with lockout state (LWLS), all HP 3852A commands can be entered from the front panel. Commands entered over the HP-IB are accepted but discarded by the instrument if the HP 3852A is not addressed to listen.

When the HP 3852A is in remote (REMS) or remote with lockout (RWLS) state, only the subset of commands shown can be entered from the front panel.

ADDR?	FASTDISP	POWEROFF	VREAD
ALRM	ID?	RQS?	WAIT
BEEP	IDN?	SIZE?	WAITFOR
CAT	IF...ENDIF	SRQ	WHILE... END WHILE
CLOSE?	INDEX?	STATE?	
DISP	INTR?	SUB	
ELSE	LOCAL	SUBEND	
ERR?	MONITOR	TIME	
ERRSTR?	NEXT	TIMEDATE	
EXTEND?	PAUSE	USE?	

ON Front panel keyboard totally disabled, including LOCAL, CLR, RST, and SRQ keys. No commands can be entered from keyboard, regardless of the state (remote/local) of the HP 3852A.

Remarks

Data Returned

None

<u>Related Commands</u>

LOCAL

Examples

Example: Disable Keyboard

This program line disables the front panel keyboard. In this mode, no commands can be entered from the keyboard. Front panel control is restored by sending LOCK OFF over the HP-IB.

```
10 OUTPUT 709;"LOCK ON"
```

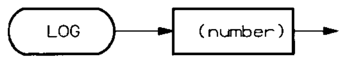
• **Math Function**

Description

Numeric expression evaluated as a command parameter. Returns the natural logarithm (base e) of the specified number.

Syntax

LOG (*number*)



RR_72

Parameters

number Number or numeric expression that must evaluate to a range >0.

Remarks

None

Examples

Example: Logarithm, Base e

The following program takes the natural logarithm (base e) of 2.345 and returns the result (.8522854) to the controller.

```
10 OUTPUT 709;"RST"                 !Reset the HP 3852A
20 OUTPUT 709;"VREAD LOG (2.345)"   !Compute nat log of 2.345
30 ENTER 709;Log                    !Enter nat log
40 PRINT "Nat log =  ";Log          !Display nat log
50 END
```

For this program, a typical display is:

```
Nat log = .8522854
```

LOGCHAN

- Mainframe

Description

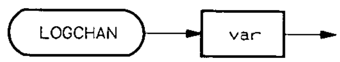
Designates the array in which channel numbers are stored as a result of the ENABLE LOGCHAN command.

Prerequisites

None

Syntax

LOGCHAN *var*



RR_73

Parameters

var Name of the REAL or INTEGER array that will store the channel numbers.

Remarks

Index Pointer Is Not Preset

The *var* array index pointer is not preset to 0 by LOGCHAN, but remains at its previous position. When logging is disabled and later re-enabled (using the same array), logging will continue to fill the array from the point where it was disabled. When the array is filled, further attempts to log channel numbers generates an error and the index must be reset with an INDEX or VREAD command.

Channel/Slot Logged

When using CONFMEAS or MEAS, multiplexer channels are logged. When using CHREAD or XRDGS, the counter channel, digital input channel, or voltmeter slot is logged.

Data Returned

None

Related Commands

ENABLE LOGCHAN, DISABLE LOGCHAN, INTEGER, REAL, DIM, INDEX, VREAD

Examples

Example: Logging Channel Numbers

This program configures the voltmeter in mainframe slot 1 for DC voltage measurements. Channels 2 through 5 of a multiplexer in slot 2 of the mainframe are scanned and measured and the channel numbers are stored in mainframe array CHS. After the measurements are taken, both the channel numbers and readings are transferred to the controller and displayed.

```
 10 INTEGER Chan(0:3)                                !Declare controller arrays
 20 REAL Chrdgs(0:3)
 30 OUTPUT 709;"RST"                                 !Reset the HP 3852A
 40 OUTPUT 709;"USE 100"                             !Use vm in slot 1
 50 OUTPUT 709;"INTEGER CHS(3)"                      !Array to store ch nos
 60 OUTPUT 709;"REAL RGS(3)"                         !Array for readings
 70 OUTPUT 709;"LOGCHAN CHS"                         !Array to receive ch nos
 80 OUTPUT 709;"ENABLE LOGCHAN"                      !Enable ch logging
 90 OUTPUT 709;"CONFMEAS DCV,202-205,INTO RGS"       !Meas DCV, store in RGS
100 OUTPUT 709;"VREAD CHS"                           !Ch nos to output buffer
110 ENTER 709; Chan(*)                               !Enter ch nos
120 OUTPUT 709;"VREAD RGS"                           !Readings to output buffer
130 ENTER 709;Chrdgs(*)                              !Enter readings
140 FOR I=0 TO 3                                     !Print chs and readings
150 PRINT Chan(I),Chrdgs(I)
160 NEXT I
170 END
```

For the channels measured, a typical display is:

```
202     5.20265
203     4.527
204     5.11727
205     .513794
```

MAT

• **Array Operator**

Description

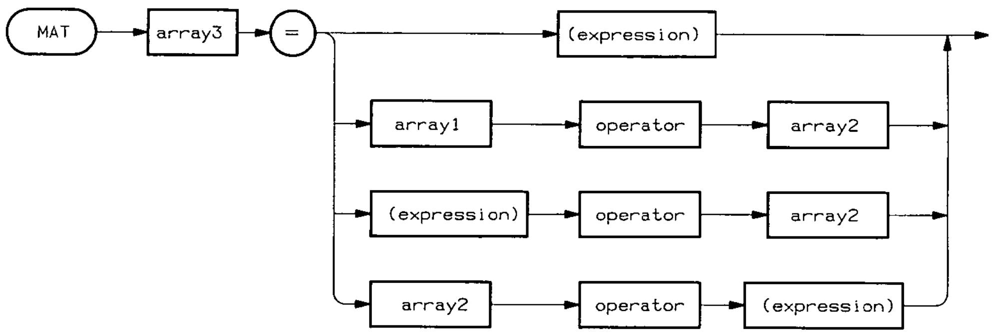
Array Operator. MAT enables you to initialize arrays and perform arithmetic operations with arrays or with arrays and numeric expressions.

Prerequisites

Requires firmware revision 3.5 or greater. Only Real or Integer arrays can be used and the arrays must be the same size.

Syntax

MAT *array3 = (expression)*
MAT *array3 = array1 operator array2*
MAT *array3 = (expression) operator array2*
MAT *array3 = array2 operator (expression)*



Parameters

array1, array2, array3 — Real or Integer arrays on which the operation is performed. All arrays must be the same size.

(expression) — See the *number* parameter in the Glossary.

operator — Math operation performed on the arrays. Operators include +, -, *, /. The operation is performed on every array element, and the results are placed in the corresponding elements of the result array. For example, **MAT** *array3 = array1 * array2* multiplies element i of *array1* by element i of *array2* and places the product in element i of *array3*.

Remarks

Rules Regarding Math Operations

For the *, +, and - math operations, the calculations are done in Real math unless both of the operand arrays are Integer. In this case, the calculations are also Integer. Division (/) is done in Real math regardless of the type of arrays.

When the result array is a different type than the operand arrays (i.e., one is Real and the others are Integers), the HP 3852A makes the conversion necessary for the result array. The conversion, however, is made after the operation is performed. Thus, if the array receiving the results is Real, an integer overflow can still occur when the operand arrays are Integer.

The Result Array Can be one of the Operand Arrays

When performing any of the math functions on arrays, the result array can be the same as either of the operand arrays (e.g. MAT A = A * B).

Data Returned

The data returned following the operation is returned to the result array.

Related Commands

None

Examples

Example: Using the MAT Command

The following program segments demonstrate the function of the MAT command.

```
60    OUTPUT 709;"REAL A(99),C(99)"
70    OUTPUT 709;"INTEGER B(99)"
   .
   .
   .
110   OUTPUT 709;"MAT B = (5)"
120   OUTPUT 709;"MAT A = B+C"
```

In this segment, the MAT command sets the value of each element in array B to 5 (line 110). The MAT command in line 120 adds that value to the values (stored previously) in array C and sends the results to array A.

```
40    OUTPUT 709;"REAL A(49)"
   .
   .
   .
120   OUTPUT 709;"MAT A = A*(2.5)"
```

In this segment, each value (previously stored) in array A is multiplied by 2.5. The result is then returned to array A.

- **HP 44701A Integrating Voltmeter**
- **HP 44702A/B High-Speed Voltmeter (Scanner or System Mode)**

Description

MEAS selects the HP 44701A or HP 44702A/B voltmeter measurement function and then initiates a scan and measurement of specified channels.

Prerequisites

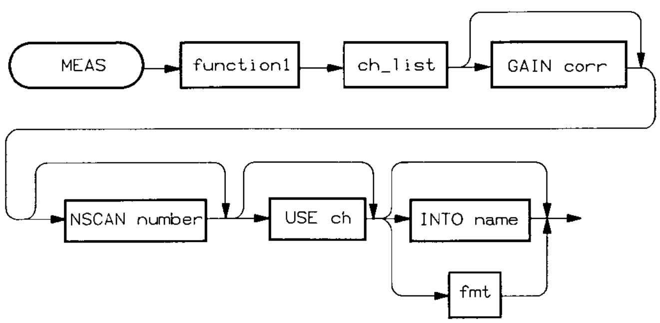
MEAS requires mainframe firmware revision 2.0 or greater for use with an HP 44717A, HP 44718A, HP 44719A, or HP 44720A and requires mainframe firmware revision 3.5 or greater for use with an HP 44730A, HP 44732A, or HP 44733A. The NSCAN parameter is available with firmware revision 2.2 or greater and the GAIN parameter is available with firmware revision 3.5 or greater. The system must not be scanning with an HP 44702A/B voltmeter when MEAS is executed.

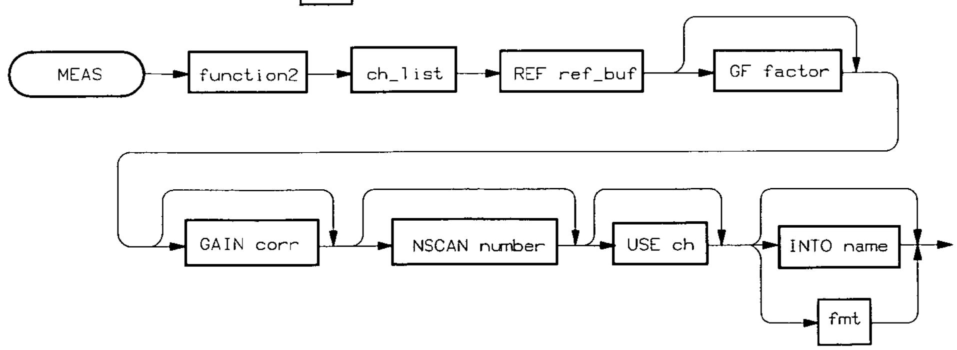
Syntax

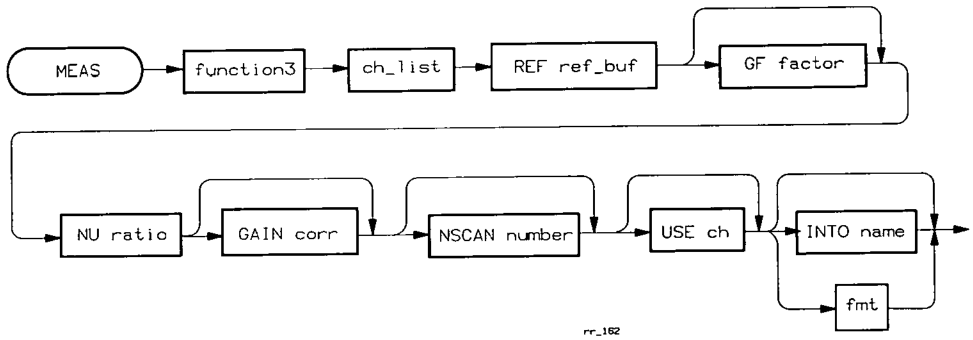
MEAS *function1 ch_list* [**GAIN** *corr*] [**NSCAN** *number*] [**USE** *ch*] [**INTO** *name*] or [*fmt*]

MEAS *function2 ch_list* **REF** *ref_buf* [**GF** *factor*] [**GAIN** *corr*] [**NSCAN** *number*] [**USE** *ch*] [**INTO** *name*] or [*fmt*]

MEAS *function3 ch_list* **REF** *ref_buf* [**GF** *factor*] **NU** *ratio* [**GAIN** *corr*] [**NSCAN** *number*] [**USE** *ch*] [**INTO** *name*] or [*fmt*]







Parameters

function HP 44701A or HP 44702A/B voltmeter measurement function. The *function1*, *function2*, or *function3* parameter specified causes the voltmeter to be configured for and make the measurement shown. Unless noted, *function* applies to both voltmeters. *function1* is used for general measurements (voltage, resistance, thermocouple, thermistor, RTD, and strain references), *function2* is used for non-Poisson arrangement strain gage measurements, and *function3* is used for Poisson arrangement strain measurements.

function1	Measurement	Note
ACV	AC voltage	[1]
DCV	DC voltage	
OHM	2-wire ohms	
OHM10K	2-wire ohms up to 10 kΩ	[2]
OHM100K	2-wire ohms up to 100 kΩ	[2]
OHM1M	2-wire ohms up to 1 MΩ	[2]
OHMF	4-wire ohms	
OHMF10K	4-wire ohms up to 10 kΩ	[2]
OHMF100K	4-wire ohms up to 100 kΩ	[2]
OHM1M	4-wire ohms up to 1 MΩ	[2]
TEMPtype	Thermocouple temperature (include ref temp) type = B, E, J, K, N14, N28, R, S, and T	

REFT	Reference temperature (isothermal block)	
THMtype	Thermistor (2-wire ohms) type = 2252 (2252 Ω thermistor) type = 5K (5 kΩ thermistor) type = 10K (10 kΩ thermistor)	
THMFtype	Thermistor (4-wire ohms) type = same as THMtype	
RTDtype	RTD (2-wire ohms) type = 85 (RTDs with α = 0.00385 Ω/Ω/°C) type = 92 (RTDs with α = 0.003916 Ω/Ω/°C)	
RTDFtype	RTD (4-wire ohms) type = same as RTD type	
STRVEX	Strain gage bridge excitation voltage	[3]
STRUN	Strain gage unstrained bridge output	[4]

function2	Measurement	Note
STRQ	1/4 bridge strain	[4]
STRHB	Bending 1/2 bridge strain	[4]
STRFB	Bending full bridge strain	[4]
STRQTEN	Tension shunt verification	[5]
STRQCOMP	Compression shunt verification	[5]

function3	Measurement	Note
STRHP	1/2 bridge Poisson strain	[4]
STRFBP	Full bridge bending Poisson strain	[4]
STRFP	Full bridge Poisson strain	[4]

[1] = HP 44701A voltmeter only.
[2] = HP 44702A/B voltmeter only.
[3] = HP 44717A/44718A/44719A/44720A/44730A/44732A/44733A only.
[4] = HP 44717A/44718A/44719A/44720A/44732A/44733A only.
[5] = HP 44717A and HP 44718A only.

ch_list Address of channel list. See Glossary.

REF *ref_buf* For MEAS when *function2* or *function3* is used, array or number containing the unstrained bridge output voltage (reference voltage) measurements for the corresponding channel list (*ch_list*). When a one-element array, a variable, or a number is declared or specified for *ref_buf*, the single reference is used for all channels in *ch_list*. The contents of the array or variable are typically generated by CONFMEAS STRUN ... INTO *ref_buf*.

GF *factor* — For MEAS when *function2* or *function3* is used, array or number containing or representing a gage factor. When several gage factors are stored in an array, the gage factor in array element 0 is used for the strain measurement on the first channel in the list, the gage factor in array element 1 is used for the second channel in the list, etc.

If the number of channels on which strain is measured is greater than the maximum index of the array, the array "wraps around" and the factors are used again as necessary. When a one-element array, variable, or number is declared or specified for the gage factor, the single gage factor is used for all channels in the channel list.

Default gage factor = 2.0. When a gage factor is specified with an exponent of -6 (e.g., GF 2.E-6), the result is returned in microstrain.

NU *ratio* — When *function3* is used (for Poisson configuration only), array or number containing or representing a Poisson ratio. When several ratios are stored in an array, the Poisson ratio in array element 0 is used for the strain measurement on the first channel in the list, the ratio in array element 1 is used with the second channel in the list, etc.

If the number of channels on which strain is measured is greater than the maximum index of the array, the array "wraps around" and the ratios are used again as necessary. When a one-element array, variable, or number is declared or specified for the Possion ratio, the single ratio is used for all channels in the channel list.

GAIN *corr* — Real or Integer array or a number containing value(s) by which the measured readings are divided. The corrected values are stored in the mainframe or are sent to the output buffer and/or to the display in RASC format. Default GAIN *corr* = no correction and thus no format conversion.

If the number of channels on which strain is measured is greater than the maximum index of the array, the array "wraps around" and the factors are used again as necessary. When a one-element array, variable, or number is declared or specified for the gain correction, the single correction factor is used for all channels in the channel list.

The channel gain can be from an external source or from the amplification function of an HP 44730A, HP 44732A, or HP 44733A (see the GAIN command).

NSCAN *number* — Number of scans to be made through the *ch_list*. Default NSCAN = 1. NSCAN is only available with mainframe firmware revision 2.2 or greater. (NSCAN *number*)*(number of channels in *ch_list*)* (NRDGS *number*) must result in 67,108,863 readings or less. For an HP 44702A/B in scanner mode and RDGS GPIO set, NSCAN *number* range = 1 to 2147483647.

When measuring thermocouples, the reference is measured prior to each pass through the channel list. When measuring strain (using *function2* or *function3*), the bridge excitation voltage (STRVEX) is measured prior to each pass through the channel list.

USE *ch* Slot where the voltmeter is installed.

INTO *name* Mainframe array or variable in which data is stored. For strain gage reference measurements (STRUN), the array should be Real. When references are stored into this array, each bridge completion channel reference is stored sequentially into the array starting at the current index. (See **REF** *ref_buf*.)

fmt Data format in which the measured function is displayed or returned to the output buffer. The default format is RASC. For STRVEX, the PACK format is the voltmeter packed format. For all other strain functions (STRxx), the PACK format is RL64. If the GAIN *corr* parameter is specified, *fmt* is always RL64.

Remarks

Configuring the Voltmeter With CONF

Before any measurement is made with MEAS, configure the voltmeter with the CONF command (refer to CONF for settings). If required, the conditions set by CONF can then be changed with low-level commands.

MEAS Disables and Clears Interrupts

Executing MEAS disables and clears all interrupts enabled on the measurement channel(s).

4-Wire Ohms Measurements

For 4-wire ohms measurements with MEAS, *ch_list* identifies the "sense" input channels and the "current source" channels are automatically selected. Inclusive channel specifiers will "skip" over "current source" channels.

Using CONF and MEAS with the HP 44701A

When using CONF and MEAS with the HP 44701A, CONF sets TERM BOTH so the mainframe backplane (i.e., multiplexer) and voltmeter external (rear panel) terminals both serve as inputs to the voltmeter. When multiplexer channels are closed, the voltage on the channel also appears on the external terminals and any voltage on the external terminals appears on the closed multiplexer channel.

CAUTION

To prevent damage to the voltmeter accessory and to the system connected to the multiplexer channel(s) during backplane scans (using CONF and MEAS), ensure that no voltage (or system) is connected to the HP 44701A voltmeter external terminals.

Incompatible Voltmeter Configurations

If MEAS detects that the voltmeter function is incompatible for the measurement functions specified, the voltmeter is reconfigured as necessary and autorange is set. In addition, MEAS checks/changes the following voltmeter functions:

HP 44701A Integrating Voltmeter

function	Setting
TRIG	TRIG HOLD or TRIG AUTO is changed to TRIG SCAN.
DISABLE INTR	DISABLE INTR is set.

HP 44702A/B High-Speed Voltmeter (System Mode)

function	Setting
TRIG	TRIG HOLD or TRIG INT is changed to TRIG SCAN.
TERM	TERM ZERO is changed to TERM INT.
DISABLE INTR	DISABLE INTR is set if RDGS SYS is set.

HP 44702A/B High-Speed Voltmeter (Scanner Mode)

function	Setting
TERM	TERM ZERO is changed to TERM RIBBON.
SCTRIG	SCTRIG HOLD or SCTRIG INT is changed to SCTRIG SCAN if TERM EXT or TERM INT is set. SCTRIG HOLD or SCTRIG SCAN is changed to SCTRIG INT if TERM RIBBON is set.
TRIG	TRIG HOLD is changed to TRIG INT.
STTRIG	STTRIG INT is set.
PRESCAN	PRESCAN 1 is set.
POSTSCAN	POSTSCAN 0 is set.
DISABLE INTR	DISABLE INTR is set if RDGS SYS is set.

Potential Deadlock

With INBUF OFF, the controller and the HP 3852A can deadlock if multiple commands (or command terminators) are sent in a single command line and the command generates enough data to fill the output buffer (MEAS can fill the output buffer). The best way to avoid a deadlock is to send a single command per command line and to read the results as soon as possible after a data-generating command is sent.

Static vs. Dynamic Strain Measurements

With static strain measurements, the specimen is strained to a final position and the difference in bridge output ratios between the unstrained and strained states is measured. Generally, static strain measurements are made with an HP 44701A Integrating Voltmeter (or equivalent) for greater measurement accuracy. When MEAS is used, maximum reading rate for an HP 44701A is about 25 readings per second.

With dynamic strain measurements, the specimen undergoes a changing strain as the measurement is made. Generally, the HP 44702A/B voltmeter is used for high-speed dynamic strain measurements, since 100 kHz reading rates are available. However, when MEAS is used, maximum reading rate with an HP 44702A/B voltmeter is about 500 readings/second so MEAS cannot be used for high-speed dynamic strain measurements. See CONF for for an example of high-speed dynamic strain measurements.

Declaring REF *ref_buf*, GF *factor*, and NU *ratio* Arrays

If arrays are declared for the reference measurement, gage factor, and Poisson ratio, the maximum index of the array should be equal to the number of channels (*ch_list*) on which strain measurements will be taken. The array should also be filled with references, factors, or ratios for the channels in the corresponding channel list.

When an array is declared, each element of the array contains the number 0 until data is written to, or stored in that particular element. For example, if strain is measured on 10 channels and the reference array (*ref_buf*) with a maximum index of 9 (starting index is 0) contains only five reference readings, the reference used to determine the strain measured on the last five channels will be 0, resulting in invalid measurements on those channels.

Data Returned

Data returned by an HP 44701A or HP 44702A/B using MEAS follows. Note that OHM10K, OHMF10K, OHM100K, OHMF100K, OHM1M, and OHMF1M apply only when an HP 44702A/B voltmeter is used. Also, ACV applies only when an HP 44701A voltmeter is used.

```
function            Data Returned

Voltage

ACV          AC voltage (volts)
DCV          DC voltage (volts)

Resistance

OHM          Resistance (ohms)
OHM10K       Resistance (≤ 10 kΩ)
OHM100K      Resistance (≤ 100 kΩ)
OHM1M        Resistance (≤ 1 MΩ)

OHMF         Resistance (ohms)
OHMF10K      Resistance (≤ 10 kΩ)
OHMF100K     Resistance (≤ 100 kΩ)
OHMF1M       Resistance (≤ 1 MΩ)

Temperature

TEMPtype     Thermocouple temp (deg C)
REFT         Ref temp of isothermal block (deg C)
THMtype      Thermistor temp (deg C)
THMFtype     Thermistor temp (deg C)
RTDtype      RTD temp (deg C)
RTDFtype     RTD temp (deg C)
```

Strain

STRVEX	Bridge excitation voltage
STRUN	Unstrained bridge output
STRQ	1/4 bridge strain
STRHB	Bending 1/2 bridge strain
STRFB	Bending full bridge strain
STRQTEN	Tension shunt verification
STRQCOMP	Compression shunt verification
STRHP	1/2 bridge Poisson strain
STRFBP	Full bridge bending Poisson strain
STRFP	Full bridge Poisson strain

Related Commands

CONF, CONFMEAS

Examples

Example: DC Voltage Measurements

This program uses CONF to configure an HP 44701A voltmeter in mainframe slot 7 for DC voltage measurements. Next, the autozero mode is changed with AZERO. Then, MEAS is used to make 100 DC voltage measurements on channels 0 through 9 of an HP 44705A. Since NSCAN sets 10 scans through the channel list, 100 readings are made. For this program, the input is assumed to have been externally amplified by 10, so GAIN 10 divides each measurement by 10 before storing the value in mainframe array A.

```
10   DIM B(0:99)                              !Dim controller array
20   OUTPUT 709;"USE 700"                     !Use voltmeter in slot 7
30   OUTPUT 709;"REAL A(99)"                  !Declare 100-element array
40   OUTPUT 709;"CONF DCV"                    !Configure for DC volts
50   OUTPUT 709;"AZERO OFF"                   !Turn autozero off
60   OUTPUT 709;"MEAS DCV,0-9,GAIN 10,NSCAN 10,INTO A"  !Meas
70   OUTPUT 709;"VREAD A"
80   ENTER 709;B(*)
90   PRINT USING "K,/";B(*)
100  END
```

A typical return for 5V, 5% inputs to the voltmeter (actual voltage of 0.5V multiplied by external gain of 10) follows.

```
.4995
.5002
  .        100 readings
.499
```

Example: 4-Wire Ohms Measurements

This program uses CONF to configure an HP 44701A voltmeter in mainframe slot 7 for 4-wire ohms measurements. Next, the offset compensation mode is changed with OCOMP. Then, MEAS is used to make ten 4-wire ohms measurements on channels 0 through 9 of an HP 44705A.

```
10  DIM Ohms(0:9)                      !Dim controller array
20  OUTPUT 709;"USE 700"               !Use voltmeter in slot 7
30  OUTPUT 709;"CONF OHMF"             !Configure for 4-wire ohms
40  OUTPUT 709;"OCOMP ON"              !Turn offset comp on
50  OUTPUT 709;"MEAS DCV,0-9"          !Meas 4-wire ohms
60  ENTER 709;Ohms(*)
70  PRINT USING "K,/";Ohms(*)
80  END
```

A typical return for a set of 1 kΩ 10% resistors (values in Ω) follows.

```
1000.234
 999.723
    .         100 readings
    .
1001.665
```

Example: 1/4 Bridge Strain Measurement

This program uses CONF and MEAS to measure 1/4 bridge strain on channels 400 through 402 of an HP 44717A, HP 44718A, HP 44719A, or HP 44720A using an HP 44701A voltmeter in slot 1 of the mainframe. Results are returned in microstrain.

The PAUSE statement (line 80) halts the program after the reference voltages have been measured to allow time for the specimens to be strained. Press the Continue (or equivalent) key to continue the program after the specimens are strained.

```
10   DIM A(0:2)                                    !Define cont array
20   !
30   !Make ref measurements and pause program
40   !
50   OUTPUT 709;"REAL Str_ref(2)"                  !Define reference array
60   OUTPUT 709;"USE 100"                          !Use HP 44701A in slot 1
70   OUTPUT 709;"CONFMEAS STRUN,400-402,INTO Str_ref"   !Meas ref voltage
80   PAUSE                                         !Pause - strain specimens
90   !
100  !Measure strain on channels
110  !
120  OUTPUT 709;"CONF STRQ"                        !1/4 bridge strain conf
130  OUTPUT 709;"MEAS STRQ,400-402,REF Str_ref"    !Meas strains
140  !
150  !Enter and display strains (in microstrain)
160  !
170  ENTER 709;A(*)                                !Enter strain values
```

```
180  PRINT "Channel   Microstrain"          !Display header
190  PRINT                                  !Space
200  FOR I=0 TO 2                           !Start display loop
210    PRINT 400+I,A(I)*1.E+6               !Display microstrain
220  NEXT I                                 !Increment loop
230  END
```

A typical return (expressed in microstrain) is:

```
Channel     Microstrain

 400        455.2426
 401        1077.261
 402        1696.887
```

MOD

• **Math Function**

Description

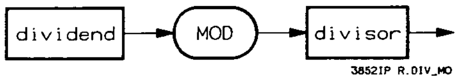
Returns the remainder of a division.

Prerequisites

Requires mainframe firmware revision 3.5 or greater.

Syntax

dividend **MOD** *divisor*



Parameters

dividend Constant, variable, or numeric expression.

divisor Constant, variable, or numeric expression.

Remarks

Data Returned

Returned is the remainder of a division.

Related Commands

DIV

Examples

Example: Demonstrating the MOD Function

The following program gives an example of the data returned by the MOD function.

```
10    OUTPUT 709;"RST"
20    OUTPUT 709;"INTEGER A,B"
30    OUTPUT 709;"A=11"
40    OUTPUT 709;"B=10"
50    OUTPUT 709;"VREAD (A MOD B)"
60    ENTER 709;Rslt
70    PRINT Rslt
80    END
```

When the program executes, 1 is displayed on the controller.

- Mainframe

Description

Enable/disable monitor mode. With MON ON or MON ALL, all commands and readings appear on the HP 3852A front panel display. With MON OFF, front panel commands and readings are still displayed, however, commands sent over the HP-IB and the readings which result are not displayed. The MON command also allows you to dedicate the display to readings from a single channel.

Prerequisites

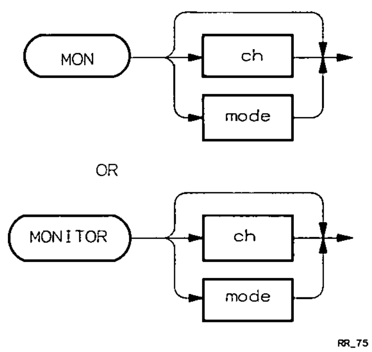
None

Syntax

MON [*ch*] or [*mode*]

or

MONITOR [*ch*] or [*mode*]



Parameters

ch Address of the channel to be monitored (i.e., the channel to which the display will be dedicated). Must be a channel from which numerical data is returned. Channel number range depends on accessory used.

mode Enables and disables the monitor mode. Power-on and default *mode* = ON.

mode	Definition
ON or ALL	Monitor all channels.
OFF	Monitor no channels.

Remarks

Front Panel Displays

When monitoring a single channel, the left display window is updated only by commands involving the channel specified or by error messages.

MONITOR may be executed from the front panel while in remote, but MON cannot. With DISP ON and MON ON (or MON ALL), the MON annunciator is ON. Note that Binary formats (IN16 or RL64) and packed formats (PACK) are not displayed.

DISP Command vs. MON Command

In order to display commands and readings on the mainframe display, DISP ON must be set in addition to MON ON or MON ALL.

Data Returned

None

Related Commands

None

Examples

Example: Disabling the Monitor Mode

This program line disables the monitor mode. When the monitor mode is disabled, commands sent over the HP-IB and the readings which result are not displayed.

```
10 OUTPUT 709;"MON OFF"        !Disable monitor mode
```

Example: Monitoring a Single Channel

This example shows one way to monitor a single channel. The CONFMEAS command is used to make DC voltage measurements on channels 0 through 4 of a multiplexer accessory in slot 1 of the mainframe using a voltmeter in slot 0. However, the MON command allows only the results from channel 0 to appear on the front panel display.

```
10 OUTPUT 709;"RST"                          !Reset the HP 3852A
20 OUTPUT 709;"MON 100"                      !Display ch 0 results only
30 OUTPUT 709;"CONFMEAS DCV,100-104,USE 0"   !Conf/meas chs 0-4 in slot 1
40 END
```

MONITOR

• Mainframe

Description

See **MON** Command

- **Mainframe**

Description

See **MONMEAS** Command

MONMEAS

- **HP 44701A Integrating Voltmeter**
- **HP 44702A/B High-Speed Voltmeter (Scanner or System Mode)**

Description

Select voltmeter measurement function, scan, measure, and display. MONMEAS selects the voltmeter function then causes the HP 3852A to step to the first channel in the *ch_list* and repeatedly make measurements on this channel. (If required, conversion to temperature equivalent is performed.) The results (in RASC format) are sent to the front panel display only. Pressing SADV KEY advances the scan to the next channel in the *ch_list*.

Prerequisites

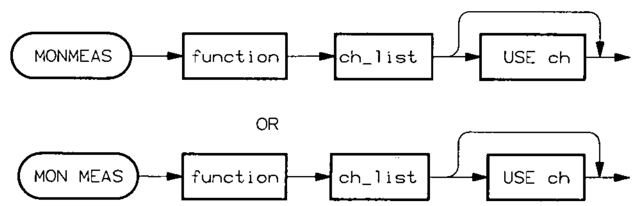
The voltmeter to perform the measurements should be previously configured for the function with the CONF command.

Syntax

MONMEAS *function ch_list* [USE *ch*]

or

MON MEAS *function ch_list* [USE *ch*]



RR_76

Parameters

function HP 44701A or HP 44702A/B voltmeter measurement function.

HP 44701A Integrating Voltmeter

function	Description
ACV	AC voltage measurements.
DCV	DC voltage measurements.
OHM	2-wire ohms measurements.
OHMF	4-wire ohms measurements.
REFT	Reference temperature measurement of thermistor on isothermal block.
THMtype	2-wire ohms measurement of a thermistor. type = 2252 (2252 ohm thermistor) type = 5K (5k ohm thermistor) type = 10K (10k ohm thermistor)

function	Description
THMFtype	4-wire ohms measurement of a thermistor. type = same as THMtype.
RTDtype	2-wire ohms measurement of an RTD. type = 85 (RTDs with alpha = 0.00385 ohms/ohm/degree C) type = 92 (RTDs with alpha = 0.003916 ohms/ohm/degree C)
RTDFtype	4-wire ohms measurement of an RTD. type = same as RTDtype.

HP 44702A/B High-Speed Voltmeter

function	Description
DCV	Measure DC voltage.
OHM	Measure 2-wire ohms (same as OHM100K).
OHM10K	Measure 2-wire ohms up to 10k ohms.
OHM100K	Measure 2-wire ohms up to 100k ohms.
OHM1M	Measure 2-wire ohms up to 1M ohms.
OHMF	Measure 4-wire ohms (same as OHMF100K).
OHMF10K	Measure 4-wire ohms up to 10k ohms.
OHMF100K	Measure 4-wire ohms up to 100k ohms.
OHMF1M	Measure 4-wire ohms up to 1M ohms.
REFT	Measure the reference temperature of thermistor on isothermal block.
THMtype	Measure 2-wire ohms of a thermistor. type = 2252 (2252 ohm thermistor) type = 5K (5k ohm thermistor) type = 10K (10k ohm thermistor)
THMFtype	Measure 4-wire ohms of a thermistor. type = same as THMtype.
RTDtype	Measure 2-wire ohms of an RTD. type = 85 (RTDs with alpha = 0.00385 ohms/ohm/degree C) type = 92 (RTDs with alpha = 0.003916 ohms/ohm/degree C)
RTDFtype	Measure 4-wire ohms of an RTD. type = same as RTDtype.

ch_list Address of channel list. See Glossary.

USE *ch* Voltmeter slot number. See Glossary.

Remarks

Configuring Voltmeter to Make Measurements

Before executing MONMEAS, configure the voltmeter by executing CONF (with appropriate function) or by using CONF to set preset values and then using lower level commands (FUNC, RANGE, etc.) to change desired preset values. This is important because MONMEAS doesn't do configuration and thus leaves the TERM setting as it was - even if the voltmeter is not connected to the backplane.

MONMEAS Disables Interrupts

Executing the MONMEAS command disables all interrupts enabled on the measurement channel.

Incompatible Voltmeter Configurations

If the MONMEAS command detects that the voltmeter configuration is incompatible for the measurement function specified, the voltmeter is re-configured as necessary and autorange is set. Additionally, MONMEAS checks/changes the following voltmeter functions:

HP 44701A Integrating Voltmeter

function	Setting
TRIG	TRIG SGL is set.
DISABLE INTR	DISABLE INTR is set.

HP 44702A/B High-Speed Voltmeter (System Mode)

function	Setting
TRIG	TRIG SGL is set.
TERM	TERM ZERO is changed to TERM INT.
RDGS	RDGS SYS is set.
RDGSMODE	RDGSMODE DAV is set.
NRDGS	NRDGS 1 is set.
DISABLE INTR	DISABLE INTR is set.

HP 44702A/B High-Speed Voltmeter (Scanner Mode)

function	Setting
SCTRIG	SCTRIG SGL is set.
TRIG	TRIG INT is set.
STTRIG	STTRIG INT is set.
TERM	TERM ZERO is changed to TERM RIBBON.
PRESCAN	PRESCAN 1 is set.
POSTSCAN	POSTSCAN 0 is set.
RDGS	RDGS SYS is set.
RDGSMODE	RDGSMODE DAV is set.
NRDGS	NRDGS 1 is set.
DISABLE INTR	DISABLE INTR is set.

Making 4-Wire Ohms Measurements

With 4-wire ohms measurements, each channel specifier in the *ch_list* identifies an independent 4-wire ohms measurement. The specified channels indicate the "sense" input channels (ohms HIGH/LOW). The "current source" channel (voltmeter HIGH/LOW) is automatically determined for each multiplexer. Inclusive channel specifiers will "skip" over "current source" channels.

Data Returned When HP 44701A Integrating Voltmeter Used

function	Data Returned	Method
ACV	AC voltage on channel monitored	
DCV	DC voltage on channel monitored	
OHM	Resistance on channel monitored	2-wire ohms
OHMF	Resistance on channel monitored	4-wire ohms
REFT	Reference temp (deg C) of isothermal block	
THMtype	Thermistor temp (deg C) on ch monitored	2-wire ohms
THMFtype	Thermistor temp (deg C) on ch monitored	4-wire ohms
RTDtype	RTD temp (deg C) on channel monitored	2-wire ohms
RTDFtype	RTD temp (deg C) on channel monitored	4-wire ohms

Data Returned When HP 44702A/B High-Speed Voltmeter Used

function	Data Returned	Method
DCV	DC voltage on channel monitored	
OHM	Resistance on channel monitored	2-wire ohms
OHM10K	Resistance (<= 10 kohm) on ch monitored	2-wire ohms
OHM100K	Resistance (<= 100 kohm) on ch monitored	2-wire ohms
OHM1M	Resistance (<=1 Mohm) on ch monitored	2-wire ohms
OHMF	Resistance on channel monitored	4-wire ohms
OHMF10K	Resistance (<= 10 kohm) on ch monitored	4-wire ohms
OHMF100K	Resistance (<= 100 kohm) on ch monitored	4-wire ohms
OHMF1M	Resistance (<= 1 Mohm) on ch monitored	4-wire ohms
REFT	Reference temp (deg C) of isothermal block	
THMtype	Thermistor temp (deg C) on ch monitored	2-wire ohms
THMFtype	Thermistor temp (deg C) on ch monitored	4-wire ohms
RTDtype	RTD temp (deg C) on channel monitored	2-wire ohms
RTDFtype	RTD temp (deg C) on channel monitored	4-wire ohms

Related Commands

CONF, MEAS, FUNC, RANGE, USE

Examples

Example: Monitoring DC Voltage Measurements

This program uses the CONF command to configure an HP 44701A voltmeter in mainframe slot 1 to its preset values to make DC voltage measurements. Then, the MONMEAS command causes the first channel in the *ch_list* (channel 2) to be scanned and repeatedly measured.

Pressing SADV KEY on the front panel advances the scan to the next channel in the *ch_list* (channel 3). Pressing SADV KEY three more times advances through the remaining channels in the *ch_list*.

```
10 OUTPUT 709;"RST"                     !Reset the HP 3852A
20 OUTPUT 709;"USE 100"                 !Use voltmeter in mainframe slot 1
30 OUTPUT 709;"CONF DCV"                !Configure for DC voltage
40 OUTPUT 709;"MONMEAS DCV,402-405"     !Meas/display DCV on chs 2-5
50 END
```

- **HP 44701A Integrating Voltmeter**
- **HP 44702A/B High-Speed Voltmeter (Scanner or System Mode)**

For

- **HP 44717A 10 Bridge 120Ω Static Strain Gage Relay Multiplexer**
- **HP 44718A 10 Bridge 350Ω Static Strain Gage Relay Multiplexer**
- **HP 44719A 10 Bridge 120Ω Static Strain Gage FET Multiplexer**
- **HP 44720A 10 Bridge 350Ω Static Strain Gage FET Multiplexer**

Description

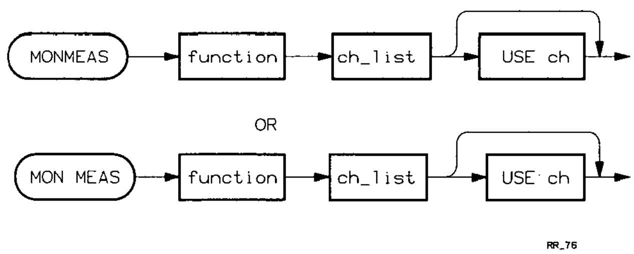
Select voltmeter measurement function, scan, and measure. The MONMEAS command initiates a scan and measurement of the specified strain gage multiplexer channels. Repeated measurements are made on the channels in the list with SADV KEY used to advance the scan. Measurements using MONMEAS are sent only to the display.

Prerequisites

The HP 44701A or HP 44702A/B voltmeter should be previously configured by the CONF command. For voltage or resistance measurements, the corresponding channel jumper must be set to the "full bridge" position. The strain gage accessories require mainframe firmware revision 2.0 or greater.

Syntax

MONMEAS *function ch_list* [USE *ch*]



Parameters

function Measurement function is the strain gage excitation voltage (±VS) or a DC voltage or 2-wire ohms measurement.

function	Description
STRVEX	Measure the bridge excitation voltage (±VS). The excitation voltage is the same for all bridge completion channels (0-9) and can be measured on any channel.
DCV	Initiate a scan and perform a DC voltage measurement. For DC voltage measurements involving FET strain gage multiplexers, input voltages should not exceed ±10V. For the relay strain gage multiplexers, there should not be greater than ±170V between any two points on the circuit or between circuit and chassis.

Function	Description
OHMxx	Initiate a scan and perform a 2-wire ohms measurement. For the HP 44702A/B, xx = measurement range (e.g. 10K, 1M). xx is not specified when using the HP 44701A.

ch_list Address of the channel on which the bridge excitation voltage, DC voltage, or resistance is measured. Valid channels for these functions are the bridge completion channels 0-9.

USE *ch* Slot where the voltmeter is installed.

Remarks

Configuring the Voltmeter With the CONF Command

Before any measurement is made with the MONMEAS command, the voltmeter that will perform the measurement should be configured with the CONF command. The voltmeter functions configured are listed under the "Remarks" section for the CONF command. If desired, the CONF command settings can be altered by the lower-level voltmeter commands to achieve the desired configuration.

Incompatible Voltmeter Configurations

If the MONMEAS command detects that the voltmeter configuration is incompatible for the measurement function specified, the voltmeter is re-configured as necessary and autorange is set. Additionally, MONMEAS checks/changes the following voltmeter functions:

HP 44701A Integrating Voltmeter

function	Setting
TRIG	TRIG SGL is set.
DISABLE INTR	DISABLE INTR is set.

HP 44702A/B High-Speed Voltmeter (System Mode)

function	Setting
TRIG	TRIG SGL is set.
TERM	TERM ZERO is changed to TERM INT.
RDGS	RDGS SYS is set.
RDGSMODE	RDGSMODE DAV is set.
NRDGS	NRDGS 1 is set.
DISABLE INTR	DISABLE INTR is set.

HP 44702A/B High-Speed Voltmeter (Scanner Mode)

function	Setting
SCTRIG	SCTRIG SGL is set.
TRIG	TRIG INT is set.
STTRIG	STTRIG INT is set.
TERM	TERM ZERO is changed to TERM RIBBON.
PRESCAN	PRESCAN 1 is set.
POSTSCAN	POSTSCAN 0 is set.
RDGS	RDGS SYS is set.
RDGSMODE	RDGSMODE DAV is set.
NRDGS	NRDGS 1 is set.
DISABLE INTR	DISABLE INTR is set.

Data Returned

The data returned by the MONMEAS command depends on the measurement function specified. Note that data is returned to the mainframe display only.

Related Commands

CONF, CONFMEAS, MEAS, USE

Examples

Example: Monitoring the Bridge Excitation Voltage

This example shows how the MONMEAS command is used to measure the bridge excitation voltage. Since the excitation voltage is the same for all bridge completion channels, only one channel is monitored.

```
10 OUTPUT 709;"RST"                     !reset the HP 3852A
20 OUTPUT 709;"USE 700"                 !use the voltmeter in slot 7
30 OUTPUT 709;"CONF STRVEX"             !configures the voltmeter
40 OUTPUT 709;"MONMEAS STRVEX, 400"     !monitors the excitation voltage
50 END
```

When this program is executed, the excitation voltage will appear in the display. By watching the display, you can then set the voltage to the desired level. Pressing the SADV KEY or CLEAR keys terminates the MONMEAS function and enables the next command to execute.

Example: Monitoring DC Voltage

This example shows how to use the MONMEAS command to monitor DC voltages input through strain gage multiplexer channels. Note that to advance to the next channel in this example, SADV KEY on the front panel must be pressed. Strain gage accessory configuration and wiring for DC voltage measurements is covered in the strain gage manual.

```
10 OUTPUT 709;"RST"                      !reset the HP 3852A
20 OUTPUT 709;"USE 700"                  !use the voltmeter in slot 7
30 OUTPUT 709;"CONF DCV"                 !configures the voltmeter
40 OUTPUT 709;"MONMEAS DCV, 400-402"     !monitors the DC voltage
50 END
```

In this program, the scan is advanced from channel 0 through channel 2 by pressing SADV KEY. Pressing the key three times completes the scan.

• **HP 44714A 3-Channel Stepper Motor Controller/Pulse Output**

Description

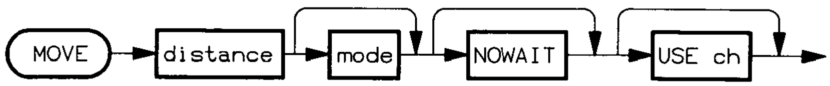
Specifies a distance to be moved and determines the number of pulses required for the move (the parameters of the PROFILE command and the scale factor of the PSCALE command are used to calculate the number of pulses required).

Prerequisites

Requires mainframe firmware revision 3.0 or greater.

Syntax

MOVE *distance* [*mode*] [*NOWAIT*] [USE *ch*]



Parameters

distance The *distance* parameter is a positive or negative number that specifies the distance to be moved. It can be an absolute distance or a distance relative to the present location which is determined by the mode option.

mode The *mode* parameter specifies whether the move is to be an absolute or a relative move. If no parameter is specified, the default is ABS.

mode	Definition
REL	The REL mode means the move is relative to the current position. If the current position is 5 and a relative distance of 10 is specified, the move will end at 15.
ABS (default)	The ABS mode tranforms the *distance* parameter into a final position. If the current position is 5 and an absolute distance of 10 is specified, the move will end at position 10.

NOWAIT Dependent on the PROFILE parameters, the PSCALE scale factor, and whether the move is relative or absolute, the processor calculates the number of pulses to be generated. This is called the set-up. The MOVE command normally completes after the last pulse has been generated restricting the processor from participating in other activities. If the *NOWAIT* option is specified, the processor is free to go to other activities after the MOVE command is set up (could include TRIG SGL or HALT SGL). Pulses are generated when a TRIG is recieved.

USE *ch* Channel which the move pulses are to output from. Channel range can be ES00 to ES02.

Remarks

Power-On State

The power-on state for the mode is ABS.

The MOVE Pulse Train

The pulse train generated by a MOVE command is dictated by the trapezoidal motion profile specified in the PROFILE command. Pulses start at the PROFILE "min" value, accelerate to the "max" value, decelerate to the "min" value, and stop. On short moves, the "max" value may not be reached.

Position Errors Due to Scale Factor

The scale factor declared in PSCALE can cause position errors if there is not enough resolution. For example, if you set PSCALE = 2 and execute a MOVE 1, 1/scale_factor = 1/2 is rounded to 1 and one step is made but the position counter is incremented to 2.

Data Returned

None

Related Commands

PROFILE, PSCALE, DONE?

Examples

Example: Move the Motor to Position 20

```
10 OUTPUT 709;"USE 201"                !Use channel 1 slot 2
20 OUTPUT 709;"TRIG HOLD"              !Put trigger on hold
30 OUTPUT 709;"MOVE 20 ABS NOWAIT"     !Move position 20 (absolute)
40 OUTPUT 709;"TRIG SGL"               !Trigger MOVE to initiate
50 END
```

```
NOTE: Line 30 stated as OUTPUT 709;"MOVE 20 NOWAIT" without the
ABS parameter will default to ABS and give the same result.
```

- **HP 44788A HP-IB Controller**

Description

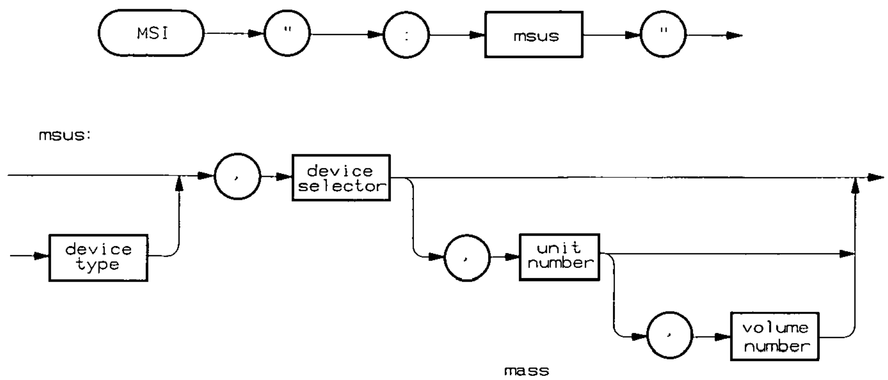
This statement specifies the current default mass storage device.

Prerequisites

Requires firmware revision 3.5 or greater.

Syntax

MSI "[*media specifier*]"



Parameters

media specifier The designation of the mass storage device to be used.

Remarks

Specifying a Source or Destination

All mass storage operations which do not specify a source or destination by either an I/O path name or msus in the file specifier use the current MSI mass storage device.

Data Returned

None.

Related Commands

None.

Examples

Example: Activate Mass Storage Device

The following statements show two ways to activate mass storage device 202, drive number 1. The first statement is entered from the HP 3852A front panel, while the second statement is entered from the system controller.

```
MSI ":,202,1"                     !Activate MS device 202, drive 1
110 OUTPUT 709;"MSI ':,202,1'"    !Activate MS device 202, drive 1
```

NEXT

- Mainframe

Description

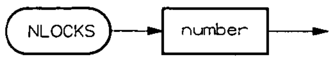
See **FOR...NEXT** Command

• Mainframe

Description Specifies the number of locks that can be requested within a multitasking system.

Prerequisites Requires firmware revision 3.5 or greater.

Syntax **NLOCKS** *number*



Parameters

number Maximum number of locks that can be requested. The range for *number* is 0 to 10.

Remarks

Loading the Number of Locks (NLOCKS) into the Operating System

The number of locks specified in the NLOCKS command do not become effective until the number is loaded into the mainframe's operating system. This is accomplished by either resetting the mainframe or cycling power after you execute the NLOCKS command. Since the number specified by NLOCKS is stored in nonvolatile memory, it does not have to be reloaded each time you run the multitasking program.

Resetting the instrument or cycling power erases all variables, arrays, and downloaded subroutines from mainframe memory. Therefore, before using any of the memory for your multitasking applications, ensure that: 1) the mainframe is in the multitasking mode (ENABLE MULTI), 2) the time-slice period has been set and loaded (TSLICE), 3) the number of run tasks and the queue size have been specified (NTASKS), and 4) the number of locks has been set and loaded. This is necessary since each of the above causes or requires a system reset before taking effect.

Cycling Power Removes the HP 3852A from the Multitasking Mode

Anytime power is cycled, the HP 3852A is removed from the multitasking mode. Thus, if power is cycled as the means for loading the NLOCKS value into the operating system, multitasking must be re-enabled.

NLOCKS Allocates Mainframe Memory for Overhead

For each lock specified by the NLOCKS command, 60 bytes of mainframe memory are allocated as overhead to "manage" the lock. For example, if `NLOCKS 2` is executed, 120 bytes are allocated. Note that this memory is in addition to the memory allocated by the NTASKS command.

Data Returned

None

Related Commands

REQUEST/RELEASE

Examples

Example: Setting the Number of Locks

The following segment shows where the NLOCKS command would be placed in the multitasking program should locks be used. Note that the reset caused by ENABLE MULTI loads the number of locks available into the operating system.

```
10   OUTPUT 709;"TSLICE 0.065"
20   OUTPUT 709;"NTASKS 5,8"
30   OUTPUT 709;"NLOCKS 2"
40   OUTPUT 709;"ENABLE MULTI"
       .
       .
       .
```

• Logical Operator

Description

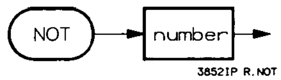
Returns a 1 if the specified argument equals 0. Otherwise, a 0 is returned.

Prerequisites

Requires mainframe firmware revision 3.5 or greater.

Syntax

NOT *number*



Parameters

number Constant, variable, or numeric expression. If the argument specified has a value of zero, 1 is returned. Otherwise, 0 is returned.

Remarks

Data Returned

See Parameters.

Related Commands

None

Examples

Demonstrating the NOT Function

The following program shows the data returned by the NOT function.

```
10    OUTPUT 709;"RST"
20    OUTPUT 709;"INTEGER A,B"
30    OUTPUT 709;"A=1"
40    OUTPUT 709;"B=0"
50    OUTPUT 709;"SUB COMP"
60    OUTPUT 709;"  IF NOT (A EXOR B) THEN"
70    OUTPUT 709;"    DISP'THE COMPLEMENT OF 0 IS 1"
80    OUTPUT 709;"  ELSE"
90    OUTPUT 709;"    DISP'THE COMPLEMENT OF 1 IS 0'"
100   OUTPUT 709;"  END IF"
110   OUTPUT 709;"SUBEND"
120   OUTPUT 709;"CALL COMP"
130   END
```

When the subroutine is called, A is assigned the value of 1 and B is assigned the value of 0. Given these values, A exclusive-or'ed with B returns a 1. Since this argument of the NOT function does not equal 0, the NOT function itself evaluates to 0, and

```
THE COMPLEMENT OF 1 IS 0
```

is displayed on the mainframe.

- **HP 44715A 5-Channel Counter/Totalizer**

Description

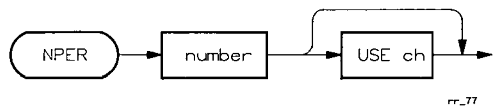
Sets the number of periods over which an input is measured, or the value minus 1 at which a counting sequence resets to zero.

Prerequisites

NPER is not used with the frequency configuration.

Syntax

NPER *number* [USE *ch*]



Parameters

number Number of periods or value - 1 where the count sequence resets to 0. At power-on, *number* = 10 for all functions.

function	Setting
TOTALM, UDCM, CDM	NPER-1 is where the counting sequence resets to 0. NPER range is 2 to 65535.
RAT	Ratio of A input counts to B input counts over NPER periods of the B input. NPER range is 1 to 65535.
PER	Measures the average of NPER periods of the A input. NPER range is 1 to 65535.
PERD	Single period measurement taken on the NPERth gated period of the A input. NPER range is 1 to 65534.

USE *ch* Specifies channel for NPER. Channel number range (depends on HP 44715A hardware configuration) = ES00 through ES04.

Remarks

NPER number Values

Each channel has a separate *number* value, with power-on value = 10. *number* remains the same when the channel is reprogrammed to another function.

Data Returned

None

Related Commands

FUNC

Examples

Example: Set Channel to Average 1000 Periods

This program enables channel 3 of a counter in mainframe slot 2 to average 1000 periods of the user's input signal and return the result to the controller. Note that the HP 44715A card configuration jumper must be set to the "CH 3" or "CH 4" position.

```
 10 OUTPUT 709;"RST"                    !Reset HP 3852A and HP 44715A
 20 OUTPUT 709;"USE 203"                !Use ch 3 in slot 2
 30 OUTPUT 709;"EDGE LH"                !Set period to start with pos edge
 40 OUTPUT 709;"FUNC PER, 1E-3"         !Avg periods using a tbase of 1 ms
 50 OUTPUT 709;"NPER 1000"              !Avg 1000 periods of user input
 60 OUTPUT 709;"TRIG SGL"               !Trigger ch 3
 70 OUTPUT 709;"CHREAD 203"             !Read average period
 80 ENTER 709;Per                       !Enter average period
 90 PRINT "Avg period = ";Per; "sec"    !Display average period
100 END
```

If the average period of the input signal is 0.0002 seconds, a typical return is:

```
Avg period = .0002 sec
```

- **HP 44701A Integrating Voltmeter**

Description

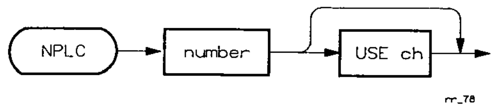
Select number of power line cycles (integration time) during which the HP 44701A Integrating Voltmeter samples the input signal.

Prerequisites

None

Syntax

NPLC *number* [USE *ch*]



Parameters

number Minimum number of power line cycles (PLC) of integration time, where integration time = PLC x power line period. For example, the period of a 50 Hz power line = 1/50 = 20 msec. If *number* = #PLC = 0.1, integration time = 20 msec x 0.1 = 2 msec.

The *number* parameter must be a number between 0 and 16. Power-on *number* = 1 @ 60Hz. *number* values other than those shown are rounded to acceptable values.

number (# PLC)	Integration Time 60 Hz	50 Hz	Max Number of Digits	Normal Mode Rejection
0.0005	10 usec	10 usec	3 1/2	0 dB
0.005	100 usec	100 usec	4 1/2	0 dB
0.1	1.66 msec	2.0 msec	5 1/2	0 dB
1	16.6 msec	20.0 msec	6 1/2	60 dB
16	267 msec	320 msec	6 1/2	60 dB

USE *ch* Voltmeter slot number. See Glossary.

Remarks

Power-On Value Assumes 60Hz

At power-on or when the HP 44701A is reset (RST or RST *slot*), the HP 44701A bases its integration time on a line frequency of 60Hz regardless of the actual power line frequency. For line frequency of 50 or 400Hz, execute the NPLC command or CONF command BEFORE making measurements. Executing NPLC or CONF selects the proper line frequency reference.

Integration Time vs. Measurement Parameters

With longer integration times (larger *number*), measurement resolution, accuracy, and normal mode rejection increases, but measurement speed decreases. Note that only integral numbers of power line cycles (1 and 16) provide normal mode rejection.

Data Returned

None

Related Commands

USE

Examples

Example: Setting Integration Time

This program sets the integration time of an HP 44701A Integrating Voltmeter in mainframe slot 1 to 0.1 PLCs. For 60 Hz line frequency, integration time = (1/60) x 0.1 = 1.66 msec. For 50 Hz line frequency, integration time = (1/50) x 0.1 = 2.00 msec.

```
10 OUTPUT 709;"RST"              !Reset the HP 3852A and HP 44701A
20 OUTPUT 709;"REAL DCRDGS(9)"   !Declare controller array
30 OUTPUT 709;"USE 100"          !Use voltmeter in mainframe slot 1
40 OUTPUT 709;"CONF DCV"         !Configure for DC voltage measurements
50 OUTPUT 709;"NPLC .1"          !Set .1 PLC integration time
60 OUTPUT 709;"MEAS DCV 0-9 INTO DCRDGS"
70                               !Measure voltage, store in mainframe
80 END
```

- **HP 44701A Integrating Voltmeter**
- **HP 44702A/B High-Speed Voltmeter (Scanner or System Mode)**

Description

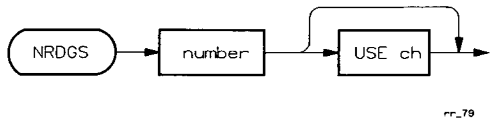
Set number of readings per trigger or number of readings per channel.

Prerequisites

None

Syntax

NRDGS *number* [USE *ch*]



Parameters

number

HP 44701A Integrating Voltmeter

Number of readings per trigger. The *number* parameter must be a number between 1 and 65535. Power-on *number* = 1.

HP 44702A/B High-Speed Voltmeter - Scanner Mode

For TERM RIBBON, *number* = number of readings per channel and the range for *number* is (number of channels - 1) x *number* < 4095.

When TERM RIBBON is NOT set, *number* is interpreted by the MEAS command as the number of readings per channel. At power-on, *number* = 1.

HP 44702A/B High-Speed Voltmeter - System Mode

number = number of readings per trigger. Range of *number* is 1 to 65535. Power-on *number* = 1.

USE *ch* — Voltmeter slot number. See Glossary.

Remarks

Data Returned

None

Related Commands

DELAY, SCTRIG, STTRIG, TRIG, USE

Examples

Example: Setting Number of Readings per Trigger

This program enables an HP 44701A Integrating Voltmeter in mainframe slot 1 to take 10 readings per trigger, triggers the voltmeter, and transfers the readings from the voltmeter to the controller. The input to the HP 44701A is via the rear panel terminals of the voltmeter.

```
 10 DIM Volt(0:9)                  !Dimension  array
 20 OUTPUT 709;"RST"               !Reset the HP 3852A and HP 44701A
 30 OUTPUT 709;"USE 100"           !Use voltmeter in mainframe slot 1
 40 OUTPUT 709;"CONF DCV"          !Configure for DC voltage measurements
 50 OUTPUT 709;"NRDGS 10"          !Select 10 readings per trigger
 60 OUTPUT 709;"TERM EXT"          !Set rear panel terminals as vm input
 70 OUTPUT 709;"TRIG SGL"          !Trigger the voltmeter
 80 OUTPUT 709;"XRDGS 100,10"      !Transfer 10 readings to output buffer
 90 ENTER 709;Volt(*)              !Enter 10 readings
100 PRINT USING "K,/";Volt(*)      !Display 10 readings
110 END
```

A typical display for ten measurements of a 5V source connected to the rear terminals of the HP 44701A is:

```
4.97776
4.95632
4.99954
4.97000
4.93257
4.94521
4.92753
4.92189
4.94563
4.96543
```

- **HP 44726A 2-Channel Arbitrary Waveform DAC**

Description

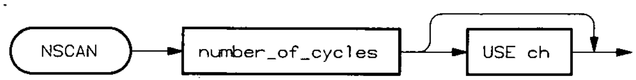
Specifies the number of waveform cycles output from the DAC. The scan ends when the last amplitude point of the final cycle is reached, regardless of the number of time base intervals the point is held.

Prerequisites

Requires firmware revision 3.5 or greater. For more than one cycle, TARM OFF or TARM AUTO must also be set.

Syntax

NSCAN *number_of_cycles* [USE *ch*]



Parameters

number_of_cycles Number of waveform cycles output from the DAC. Channel 0 can be programmed for 1 to 65536 cycles, or for a continuous output by specifying CONT. Channel 1 can be programmed for 1 cycle, or for a continuous output by specifying CONT. The power-on setting for *number_of_cycles* is CONT.

USE *ch* Channel programmed for the specified number of cycles. The default USE *ch* is channel 0.

Remarks

Setting NSCAN 1 During a Continuous or Multiple Cycle Output

When NSCAN cycles have completed, the channel is disarmed and the last amplitude point of the waveform remains at the output.

If NSCAN 1 is set during a continuous or multiple cycle output, the waveform completes its current cycle, the channel is disarmed, and the last amplitude point of the waveform remains at the output. Note that NSCAN 1 can be executed without previously setting TARM OFF or TARM AUTO.

HP 44726A Interrupts

The only time the HP 44726A can generate an interrupt is after NSCAN cycles of the waveform have occurred. Thus, NSCAN must be included with the commands which set up and enable the interrupt. Examples of how NSCAN is used for this application can be found under the HP 44726A's ENABLE INTR command.

Data Returned

None

Related Commands

ENABLE INTR

Examples

Example: Setting NSCAN 1 During A Continuous Output

The following program shows how NSCAN can be used to stop a waveform following completion of the current cycle.

In the example, a 5 Vp-p, 200 kHz square wave is defined and stored. Since NSCAN is not specified in the program, the power-on NSCAN setting (NSCAN CONT) is used. By executing NSCAN 1 as shown below the program listing, the waveform completes the current cycle, the channel becomes disarmed, and the last amplitude point on the waveform (-2.5V) remains at the output.

If NSCAN CONT (also shown below the listing) is executed, the waveform resumes as a continuous cycle output. NSCAN CONT is allowed because of the source which arms the channel (TARM AUTO - line 140). After NSCAN CONT is executed, TARM AUTO is re-asserted and another arming pulse is issued.

```
10    !Reset the mainframe and the HP 44726A. Define and store a special
20    !function square wave that is 5 Vp-p with a frequency of 200 kHz.
30    !
40    OUTPUT 709;"RST"
50    WAIT 1
60    OUTPUT 709;"WFWRITE SQV 7 5 TBASE(1/(200E3*4))"
70    !
80    !Select waveform 7 in channel 0 memory. Set the source of the
90    !trigger which advances the DAC output. Arm the channel so that the
100   !trigger signals will be accepted.
110   !
120   OUTPUT 709;"APPLY WFV 0,7"
130   OUTPUT 709;"TRIG INT"
140   OUTPUT 709;"TARM AUTO"
150   END
```

```
OUTPUT 709;"NSCAN 1"

OUTPUT 709;"NSCAN CONT"
```

• **Mainframe**

Description

Specifies the number of run tasks and sets the size of the queue from which subroutines are taken and executed.

Prerequisites

The NTASKS command is used with the multitasking capability of the HP 3852A. Multitasking is only available with mainframe firmware revision 3.0 or greater.

Syntax

NTASKS *number* [*size*]

Parameters

number Number of run tasks the system will allow. The range for *number* is 0 to 8.

size Size of the queue in which subroutine names are held until they begin execution in a run task. The range for *size* is 0 to 20 and the queue size specified should be the same or larger than the number of tasks specified by *number*. If *size* is not specified, the size of the queue is set equal to the number of run tasks.

Remarks

Loading the NTASKS Values into the HP 3852A Operating System

The number of run tasks and the queue size do not become effective until these values are loaded into the mainframe's operating system. This is accomplished by either resetting the mainframe or cycling power after you execute the NTASKS command. Since the NTASKS values are stored in nonvolatile memory, the values do not have to be reloaded each time your multitasking system runs.

Resetting the instrument or cycling power erases all variables, arrays, and downloaded subroutines in mainframe memory. Therefore, before using any of the memory for your multitasking applications, ensure that: 1) the mainframe is in the multitasking mode (ENABLE MULTI), 2) the time-slice period has been set and loaded (TSLICE), and 3) the NTASKS parameters have been set and loaded. This is necessary since each of the above causes or requires a system reset before taking effect.

Cycling Power Removes the HP 3852A from the Multitasking Mode

Anytime power is cycled, the multitasking capability of the HP 3852A is disabled. Thus, if power is cycled as a means for loading the NTASKS parameters into the operating system, multitasking must be re-enabled (ENABLE MULTI) as NTASKS applies only to the multitasking mode.

NTASKS Allocates Mainframe Memory for Overhead

By specifying the number of run tasks and the size of the queue, you minimize the amount of mainframe memory (overhead) required to manage the multitasking system. Each run task specified by the NTASKS command uses 160 bytes of memory. When the run task is activated by the RUN command, another 1216 bytes of memory are consumed. The amount of memory allocated for the queue is the queue size specified times 24 bytes.

Since memory used as overhead is no longer available to the user, it is often helpful to determine the total amount of overhead your multitasking system will use before downloading your system. The amount of overhead can be found with the formula:

```
N1*160 + N2*1216 + N3*24
```

where:

```
N1 = number of run tasks specified by NTASKS
N2 = number of run tasks activated by the RUN command
N3 = size of the queue specified by NTASKS
```

The use of this formula can be demonstrated by calculating the amount of memory allocated as overhead for the multitasking program shown in the "Examples" section. By referring ahead to the program, notice the NTASKS command in line 70 specifies 2 run tasks and a queue size also of 2. These numbers correspond to N1 and N3 in the formula. Since two RUN commands are required to activate the run tasks (lines 370 and 380), N2 in the formula is also 2. Thus, the formula with values specified for N1, N2, and N3 appears as shown below:

```
2*160 + 2*1216 + 2*24 = 2800 bytes of overhead
```

Therefore, 2800 bytes of memory are used by the multitasking program in addition to the memory used by the subroutines and the variables declared in the subroutines.

Data Returned

None

Related Commands

ENABLE MULTI, DISABLE MULTI, ENABLE EOL SWAP, DISABLE EOL SWAP, URGENCY, RUN

Examples

Example: Specifying the Number of Run Tasks and the Queue Size

This example shows how the NTASKS command is used to specify the number of run tasks and also how it sets the size of the queue. Other multitasking commands are included to show the context in which NTASKS is used.

In the program, two subroutines are downloaded and directed to run tasks. One subroutine displays the numbers from 0 to 9 and the other subroutine repeatedly displays the sequence A B C. When the run tasks are activated, they will time-slice and swap between each other and the mainframe will alternately count and display A B C.

```
10   !Set a 1.5 second time-slice and set the number of run tasks and
20   !queue size the system will allow. Place the HP 3852A in the
30   !multitasking mode. The reset caused by ENABLE MULTI will load
40   !the TSLICE and NTASKS parameters into the operating system.
50   !
60   OUTPUT 709;"TSLICE 1.5"
70   OUTPUT 709;"NTASKS 2,2"
80   OUTPUT 709;"ENABLE MULTI"
90   !
100  !Write and download the subroutines to be directed to run tasks.
110  !Subroutine A when active, displays the numbers 0 through 9.
120  !Subroutine B when active, repeatedly displays A B C.
130  !
140  OUTPUT 709;"SUB A"
150  OUTPUT 709;"  INTEGER I"
160  OUTPUT 709;"  FOR I=0 TO 9"
170  OUTPUT 709;"    DISP I"
180  OUTPUT 709;"    WAIT .5"
190  OUTPUT 709;"  NEXT I"
200  OUTPUT 709;"SUBEND"
210  OUTPUT 709;"SUB B"
220  OUTPUT 709;"  INTEGER J"
230  OUTPUT 709;"  FOR J=0 TO 9"
240  OUTPUT 709;"    DISP 'A'"
250  OUTPUT 709;"    WAIT .5"
260  OUTPUT 709;"    DISP 'B'"
270  OUTPUT 709;"    WAIT .5"
280  OUTPUT 709;"    DISP 'C'"
290  OUTPUT 709;"    WAIT .5"
300  OUTPUT 709;"  NEXT J"
310  OUTPUT 709;"SUBEND"
320  !
330  !Direct the subroutines to run tasks. Set run task 1 (subroutine
340  !A) to execute three times and run task 2 (subroutine B) to
350  !execute one time.
360  !
370  OUTPUT 709;"RUN 1 A 3"
380  OUTPUT 709;"RUN 2 B"
390  END
```

NULL

- **HP 44730A Track/Hold with Signal Conditioning**
- **HP 44732A 120 Ohm Dynamic Strain Gage FET Multiplexer**
- **HP 44733A 350 Ohm Dynamic Strain Gage FET Multiplexer**

Description

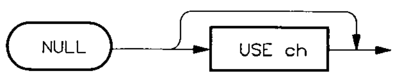
Nulls offset voltage on the specified channel.

Prerequisites

Command requires mainframe firmware revision 3.5 or greater. The system must not be scanning with the HP 44702A/B voltmeter when NULL is executed.

Syntax

NULL [USE *ch*]



Parameters

USE *ch* Specify channel to be used for the NULL command. Channel number range = ES00 through ES03.

Remarks

NULL Command Overrides AZERO Settings

The NULL command nulls offset voltages generated by both external and internal error sources on the channel. The NULL command is performed with the channel inputs connected to the user signal inputs. After a NULL command, the channel output is referenced to the offset present on the channel inputs. Note that NULL overrides the setting due to an AZERO command.

NULL Command Range

The NULL command can correct user input voltage offsets in the range of $\pm$ 4.9 Vdc/gain. Range is $\pm$ 4.9 Vdc for gain = 1; $\pm$ 0.49 Vdc for gain = 10; and $\pm$ 0.049 Vdc for gain = 100.

Data Returned

None.

Related Commands

AZERO

Examples

Example: Performing Channel Null

This program uses FUNC to set channel 500 of an HP 44730A to amplify the input signal with a gain of 10 and NULL to null channel offset errors. (The filter is not used.) Channel measurements are made with an HP 44702A/B voltmeter in slots 6 and 7 via the ribbon cable (SCANMODE ON).

```
10   OUTPUT 709;"USE 500"                 !Use ch 500
20   OUTPUT 709;"FUNC AMPLIFY,10"         !Ch 500 to amplify w/gain 10
30   OUTPUT 709;"FILTER OFF"              !Ch 500 filtering off
40   OUTPUT 709;"NULL"                    !Null ch 100 offset errors
50   OUTPUT 709;"USE 600"                 !Use voltmeter in mf slot 6
60   OUTPUT 709;"SCANMODE ON"             !Set Scanner Mode
70   OUTPUT 709;"CONFMEAS DCV,500"        !Conf/meas DC volts, ch 500
80   ENTER 709;A                          !Enter reading
90   PRINT A                              !Display reading
100  END
```

Example: Dynamic Strain Measurements

This program makes ten 1/4 bridge strain gage measurements at 10 μsec intervals on channel 500 of an HP 44732A or HP 44733A using an HP 44702A/B voltmeter in slots 6 and 7. Channel 500 is set for amplification with gain = 100, no filtering, and is configured for 1/4 bridge strain measurements. The gage factor (GF) assumed = 2.

The HP 44702A/B voltmeter is set for Scanner mode, 1 prescan, no postscans, 10 readings per channel, and internal scan, measure, and stop triggers. The ribbon cable must be connected between the HP 44732A or HP 44733A and the HP 44702A/B. Also, the EXC SENSE and EXC terminals must be connected (H to +E and L to -E2) and the bridge completion jumper must be set to the 1/2, 1/4 position.

When the program executes, the bridge offset is first nulled with the NULL command. Next, the excitation voltage is measured once with CONFMEAS STRVEX and stored in mainframe variable Exc. Then, the HP 44702A/B is configured for strain measurements and the program is paused to allow time for the specimen to be strained. Press the Continue key (or equivalent) to continue the program after the specimen is strained.

Since CONF sets SCDELAY .001, when the program is continued, a 10 msec delay occurs and then 10 measurements are made at 10 μsec intervals. The results are stored in mainframe array Strn. Since the channel is configured for 1/4 bridge measurements, the strain is computed by the COMPEN command for 1/4 bridge strain measurements. Results are displayed in microstrain.

In the COMPEN command, note that the GAIN 100 parameter is used which divides the strain measurement by 100. Since the user inputs have been multiplied by 100 with the FUNC AMPLIFY,100 command, it is also necessary to divide the results by 100 to arrive at correct strain computations.

```
10   !
20   !Define arrays and variables
30   !
40   DIM A(0:9)                           !Define controller array
50   OUTPUT 709;"REAL Strn(9)"            !Define HP 3852A REAL array
60   OUTPUT 709;"REAL Exc"                !Define HP 3852A REAL var
70   !
80   !Null bridge offset on channel 500
90   !
100  OUTPUT 709;"USE 500"                 !HP 44733A use ch is 500
```

```
110  OUTPUT 709;"FUNC AMPLIFY,100"                !Set ampl w/gain of 100
120  OUTPUT 709;"FILTER OFF"                      !Ch 500 no filtering
130  OUTPUT 709;"TRIG HOLD"                       !No triggering on 44733A
140  OUTPUT 709;"NULL"                            !Null ch 500 offset voltage
150  !
160  !Measure ch 500 exc voltage
170  !
180  OUTPUT 709;"USE 600"                         !Use 44702 vm in slot 6
190  OUTPUT 709;"SCANMODE ON"                     !Set Scanner mode
200  OUTPUT 709;"CONFMEAS STRVEX,500,INTO Exc"    !Meas and store exc voltage
210  !
220  !Set HP 44702A/B for strain measurements
230  !
240  OUTPUT 709;"CONF DCV"                        !Conf for DC volts
250  OUTPUT 709;"NRDGS 10"                        !Make 10 readings
260  OUTPUT 709;"SPER 10E-6"                      !10 usec between meas trigs
270  OUTPUT 709;"CLWRITE 500"                     !Set ch 500 as scan list
280  !
290  !Pause program (apply strain to specimen)
300  !
310  PAUSE                                        !Pause - strain specimen
320  !
330  !Make 10 strain measurements
340  !
350  OUTPUT 709;"SCTRIG INT"                      !Internal scan trig
360  OUTPUT 709;"XRDGS 600 INTO Strn"             !Trans rdgs to mf array
370  !
380  !Compute and display microstrain
390  !
400  OUTPUT 709;"COMPEN STRQ Strn STRVEX Exc,REF 0,GF 2.E-6,GAIN 100" !Comp
410  ENTER 709;A                                  !Enter microstrains
420  PRINT "Reading    Microstrain"                !Display header
430  PRINT                                        !Space
440  FOR I=0 TO 9                                 !Start computation loop
450    PRINT I+1,A(I)                             !Display microstrain
460  NEXT I                                       !Increment count
470  END
```

A typical return is:

```
Reading      Microstrain

   1         298.5461
   2         325.9725
   3         291.0972
   4         324.2795
   5         333.7605
   6         545.0994
   7         521.961
   8         383.2004
   9         329.02
   10        299.5618
```

- **HP 44701A Integrating Voltmeter**

Description

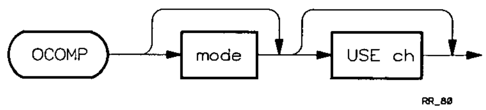
Enables or disables the HP 44701A Integrating Voltmeter offset compensation function on the 30Ω through 30kΩ ranges. Offset compensation is not done on the 300kΩ and 3MΩ ranges. Offset compensation can be used for both 2-wire and 4-wire ohms measurements.

Prerequisites

None

Syntax

OCOMP [*mode*] [USE *ch*]



Parameters

mode Offset compensation control mode. With *mode* = ON, the HP 44701A Integrating Voltmeter measures the *external* offset voltage after each ohms measurement and subtracts the offset voltage from the measurement. Power-on *mode*= OFF. Default *mode* = ON.

mode	Definition
OFF	Offset compensation disabled
ON	Offset compensation enabled

USE *ch* Voltmeter slot number. See Glossary.

Remarks

Measurement Speed

With offset compensation enabled (OCOMP ON), the HP 44701A Integrating Voltmeter measures the *external* offset voltage after each resistance measurement. This prevents the offset voltage from affecting the ohms reading, but doubles the amount of time required per reading.

No Autozero With OCOMP ON For Some Ranges

With OCOMP ON, autozero is not done on the 30 Ω to 30k Ω ranges.

Data Returned

None

Related Commands

USE

Examples

Example: Enabling Offset Compensation Function

This program configures an HP 44701A Integrating Voltmeter in slot 1 of the mainframe for 4-wire ohms measurements, enables offset compensation, then performs the measurement and transfers the reading to the computer.

```
10 OUTPUT 709;"RST"               !Reset the HP 3852A
20 OUTPUT 709;"USE 100"           !Use voltmeter in mainframe slot 1
30 OUTPUT 709;"CONF OHMF"         !Configure for 4-wire ohms measurements
40 OUTPUT 709;"OCOMP ON"          !Enable OCOMP
50 OUTPUT 709;"MEAS OHMF, 0"      !Take measurement on channel 0
60 ENTER 709;A                    !Enter reading
70 END
```

• Mainframe

Description

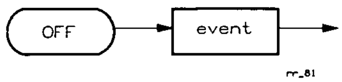
Prevents a subroutine from being called following an interrupt or exception.

Prerequisites

None

Syntax

OFF *event*



Parameters

event Interrupt or exception (alarm or limit) which calls an HP 3852A subroutine.

ALRM	Prevents a subroutine from being called when an alarm occurs.
INTR [USE ch]	Prevents a subroutine from being called when a plug-in accessory channel generates an interrupt.
LMT	Prevents a subroutine from being called when a limit is exceeded.

Remarks

Related Commands

LMT, SET ALRM, ENABLE, DISABLE, ON, STA?

Examples

Example: Using OFF Command

These program lines set up an ON *event* CALL *name* and OFF *event* sequence. In line 90, an interrupt on channel 1 in slot 2 calls subroutine Serv. However, in line 157, subroutine Serv will not be called if channel 1 interrupts.

```
90 OUTPUT 709;"ON INTR,USE 201 CALL Serv"  !Call sub if ch 1 intr
   .
   .
157 OUTPUT 709;"OFF INTR,USE 201"          !Don't call sub if ch 1 intr
   .
   .
```

ON

• **Mainframe**

Description

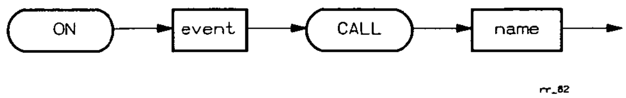
Enables a subroutine to be called following an interrupt or exception.

Prerequisites

The *event* which causes the interrupt must be enabled with the ENABLE ALRM, ENABLE LMT, or ENABLE INTR command. ENABLE INTR SYS must be set for the mainframe to recognize an accessory (ENABLE INTR) interrupt.

Syntax

ON *event* **CALL** *name*



Parameters

event Interrupt or exception (alarm or limit) which calls HP 3852A subroutine *name*.

ALRM	Allows subroutine to be called when an alarm occurs.
INTR [USE ch]	Allows subroutine to be called following an interrupt from the plug-in accessory channel specified by USE ch.
LMT	Allows subroutine to be called when a limit is exceeded.

name Name of subroutine to be called.

Remarks

Interrupt Priorities

The channel with the lowest channel number (based on the HP 3852A addressing convention) which has been enabled has the highest priority and will always be serviced first.

Servicing Interrupts

Interrupt service begins when the HP 3852A completes the command (or subroutine) being executed when the interrupt occurred. Interrupt servicing is when the mainframe uses a polling routine to identify the accessory channel which interrupted. When the channel is located, its interrupt capability is then disabled (i.e. DISABLE INTR is performed) and the servicing is complete. The mainframe (ON...CALL) or controller will then respond to the interrupt.

Using STA? to Determine if Interrupt Occurred

You can send the STA? command to determine if any channel has interrupted since power-on or since the last STA? command. Also, STA? resets (clears) the FPS, LCL, INTR, LMT, and ALRM bits in the status register (see the STA? command).

Data Returned

None

Related Commands

LMT, SET ALRM, ENABLE, DISABLE, OFF, STA?

Examples

Example: Call Subroutine on Voltmeter Data Available

This program enables an HP 44701A voltmeter in slot 1 of the mainframe to interrupt when a voltage reading is available. When the interrupt occurs (at data available), ON INTR CALL calls subroutine "Data" to respond to the interrupt.

```
10 OUTPUT 709;"RST"                    !Reset the HP 3852A
20 OUTPUT 709;"USE 100"                !Use voltmeter in slot 1
30 OUTPUT 709;"SUB Data"               !Define subroutine Data
40 OUTPUT 709;"OPEN 0,91"              !Open ch 0 and tree relay
50 OUTPUT 709;"DISP'DATA READY'"       !Display msg on front panel
60 OUTPUT 709;"SUBEND"                 !End subroutine
70 OUTPUT 709;"ON INTR CALL Data"      !Call Data on data available
80 OUTPUT 709;"CONF DCV"               !Configure for DC volts
90 OUTPUT 709;"CLOSE 0,91"             !Close ch 0 (slot 0) and tree relay
100 OUTPUT 709;"ENABLE INTR"           !Enable voltmeter to interrupt
110 OUTPUT 709;"ENABLE INTR SYS"       !Sense the backplane interrupt
120 OUTPUT 709;"TRIG SGL"              !Trigger the voltmeter
130 END
```

ON ... RUN

• **Mainframe**

Description

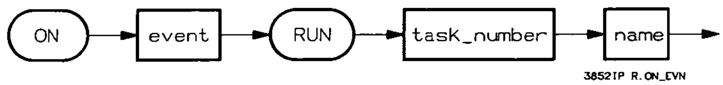
Directs a subroutine to the specified run task when a backplane interrupt, alarm interrupt, or limit interrupt occurs.

Prerequisites

Requires firmware revision 3.5 or greater and the HP 3852A must be in the multitasking mode.

Syntax

ON *event* **RUN** *task_number name*



Parameters

event Interrupt (backplane, alarm, limit) which directs the specified subroutine to the specified run task.

`INTR [USE ch]`	Directs a subroutine to a run task when a backplane interrupt occurs on the USE channel specified.
`ALRM`	Directs a subroutine to a run task when an alarm occurs.
`LMT`	Directs a subroutine to a run task when a limit is exceeded during real-time limit testing.

task_number Run task to which the subroutine is directed when the interrupt occurs. The range for *task_number* is 0 to 7. Only one run task number can be specified per command.

name Name of the subroutine directed to the run task.

Remarks

Interrupt-Run Subroutines Execute at the Urgency of the Run Task

When an interrupt occurs and directs a subroutine to a run task, the interrupt is handled at the interrupt task priority (default = 65). However, the subroutine executes at the priority of the run task (default = 85). When used with CREATE RUN, ON ... RUN can direct a subroutine to a run task at any priority level.

The USE Channel as a Command and as a Command Parameter

The USE channel is both a command and a command parameter. In the multitasking mode, the USE channel is local to the task which means the channel specified by the USE command applies only to the commands in the task (front panel, HP-IB, interrupt, run task) from which the USE command was executed. When the USE channel is a parameter, the channel specified applies only to that command. All other commands in the task use the channel specified by the USE command or the power-on USE channel (see Glossary).

Related Commands

ENABLE MULTI, CREATE RUN

Data Returned

None

Examples

Example: Directing Subroutines to Run Tasks Using Interrupts

The following example demonstrates how interrupts can be used to direct subroutines to run tasks when certain conditions occur.

In the program, three subroutines are used to control the temperature and the level of a liquid. Subroutine TEMPTASK which is directed to a run task by the RUN command, continuously monitors the temperature of the liquid. When the temperature falls below 100° C, an HP 44727A/B/C DAC accessory is programmed to apply 10V to a heater which warms the liquid to the desired temperature. When the temperature is reached, the output of the DAC is set to 0V, which turns the heater off.

Subroutines LEVEL_UP and LEVEL_DN control the level of the liquid and are directed to run tasks when an interrupt occurs indicating the tank is either full or empty. When the tank is empty, an interrupt directs subroutine LEVEL_UP to a run task. As the subroutine executes, it programs another DAC channel to apply 10V to a pump which fills the tank. When the tank is full, an interrupt directs LEVEL_DN to a run task which sets the output of the channel to 0V and turns the pump off. Since the run tasks have the same priority, LEVEL_UP and LEVEL_DN will time-slice with TEMPTASK when the corresponding interrupt occurs.

```
10    !Set a 65 ms time-slice and set the number of run tasks and
20    !queue size the system will allow. Place the HP 3852A in the
30    !multitasking mode. The reset caused by ENABLE MULTI will load
40    !the TSLICE and NTASKS parameters into the operating system.
50    !
60    OUTPUT 709;"TSLICE 0.065"
70    OUTPUT 709;"NTASKS 3,3"
80    OUTPUT 709;"ENABLE MULTI"
90    !
100   !Download the subroutine which will continually monitor the
110   !temperature of the liquid in the tank. When the temperature
120   !falls below 100 degrees C, program the DAC accessory to
130   !output 10V which will turn on the heater. When the temperature
140   !reaches 100 degrees, output 0V to turn off the DAC.
150   !
160   OUTPUT 709;"SUB TEMPTASK"
170   OUTPUT 709;"  REAL T"
180   OUTPUT 709;"  USE 400"
```

```
190   OUTPUT 709;"  CONF TEMPT"
200   OUTPUT 709;"  WHILE 1"
210   OUTPUT 709;"    MEAS TEMPT 200 INTO T "
220   OUTPUT 709;"    IF T<100 THEN"
230   OUTPUT 709;"      APPLY DCV 300,10"
240   OUTPUT 709;"    ELSE"
250   OUTPUT 709;"      APPLY DCV 300,0"
260   OUTPUT 709;"    END IF"
270   OUTPUT 709;"  END WHILE"
280   OUTPUT 709;"SUBEND"
290   !
300   !Download the subroutine which programs the DAC accessory to
310   !apply 10V to the pump when the liquid in the tank falls below
320   !the minimum level. Direct the subroutine to a run task when
330   !the interrupt caused by that transition occurs.
340   !
350   OUTPUT 709;"SUB LEVEL_UP"
360   OUTPUT 709;"  APPLY DCV 301,10"
370   OUTPUT 709;"  ENABLE INTR USE 121"
380   OUTPUT 709;"SUBEND"
390   OUTPUT 709;"ON INTR USE 121 RUN 0 LEVEL_UP"
400   !
410   !Download the subroutine which turns off the output of the DAC
420   !accessory when the liquid in the tank reaches its maximum
430   !level. Direct the subroutine to a run task when the interrupt
440   !caused by that transition occurs.
450   !
460   OUTPUT 709;"SUB LEVEL_DN"
470   OUTPUT 709;"  APPLY DCV 301,0"
480   OUTPUT 709;"  ENABLE INTR USE 123"
490   OUTPUT 709;"SUBEND"
500   OUTPUT 709;"ON INTR USE 123 RUN 1 LEVEL_DN"
510   !
520   !Program digital input channel 5 (logical channel 21) to
530   !interrupt when a high-to-low transition occurs, and channel 7
540   !(logical channel 23) to interrupt when a low-to-high transition
550   !occurs. Enable the channels to interrupt and enable the
560   !mainframe to sense backplane (accessory) interrupts.
570   !
580   OUTPUT 709;"EDGE HL USE 121"
590   OUTPUT 709;"ENABLE INTR USE 121"
600   OUTPUT 709;"EDGE LH USE 123"
610   OUTPUT 709;"ENABLE INTR USE 123"
620   OUTPUT 709;"ENABLE INTR SYS"
630   !
640   !Direct the subroutine TEMPTASK to run task 2.
650   !
660   OUTPUT 709;"RUN 2 TEMPTASK"
670   END
```

- HP 44705A 20-Channel Relay Multiplexer
- HP 44705H 20-Channel High-Voltage Relay Multiplexer
- HP 44706A 60-Channel Relay Multiplexer
- HP 44708A 20-Channel Relay Multiplexer/TC
- HP 44708H 20-Channel High-Voltage Relay Multiplexer/TC

- HP 44709A 20-Channel FET Multiplexer
- HP 44710A 20-Channel FET Multiplexer/TC

- HP 44711A 24-Channel High-Speed FET Multiplexer
- HP 44712A 48-Channel High-Speed FET Multiplexer
- HP 44713A 24-Channel High-Speed FET Multiplexer/TC

Description

Open multiplexer channels. OPEN opens a single multiplexer channel or a list of multiplexer channels. OPEN is a low-level command intended for individual switch control in special signal-routing applications.

NOTE

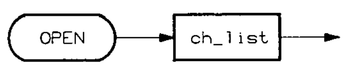
Using OPEN is not the easiest way to do routine measurements since the tree switches are not automatically opened after the measurement.

Prerequisites

None

Syntax

OPEN *ch_list*



rr_83

Parameters

ch_list Address of channel list. See Glossary.

HP 44705A/HP 44705H/HP 44709A (20-Channel Multiplexers)

CC Range	Definitions
0 - 9	Bank A Switches
10 - 19	Bank B Switches
90	Isolation Relays (HP 44709A only)
91	Sense Bus Tree Switch (Bank A)
92	Sense Bus Tree Switch (Bank B)
93	Source Bus Tree Switch (Bank A)
94	Source Bus Tree Switch (Bank B)

HP 44706A (60-Channel Multiplexer)

CC Range	Definitions
0 - 59	Bank Switches/Source Bus Tree Switch
91	Source Bus Tree Switch

HP 44708A/HP 44708H/HP 44710A (20-Channel Multiplexers/TC)

CC Range	Definitions
0 - 9	Bank A Switches
10 - 19	Bank B Switches
90	Isolation Relays (44710A only)
91	Sense Bus Tree Switch (bank switches)
92	Sense Bus Tree Switch (thermistor)
93	Source Bus Tree Switch (thermistor)
94	Source Bus Tree Switch (bank switches)

HP 44711A/HP 44713A (24-Channel Multiplexers)

CC Range	Definitions
0 - 11	BANK A Switches
12 - 23	BANK B Switches
90	Isolation Relays
91	Source Bus Tree Switch (Bank A or Bank B)
92	Sense Bus Tree Switch (Bank A or Bank B)
93	2-Wire Ohms Configuration
94	4-Wire Ohms Configuration

HP 44712A (48-Channel Multiplexer)

CC Range	Definitions
0 - 47	Bank Switches
90	Isolation Relays
91	Source Bus Tree Switch
92	Sense Bus Tree Switch
93	Source Bus and Sense Bus Tree Switches

Remarks

Controlling Multiplexer Channels

Channels are opened in the order listed in the *ch_list* parameter. All channels (and isolation relays for FET multiplexers) are opened at power-on. RST opens all channels in the HP 3852A, while RST *slot* opens all channels on the multiplexer in the slot addressed.

Controlling Tree Switches

Tree switches control signal flow between the multiplexers and the HP 3852A backplane. Sense Bus tree switches provide voltage connections to the backplane for making voltage measurements. Source Bus tree switches provide current source connections (+I and -I) to the backplane for making resistance measurements.

Controlling Isolation Relays (FET Multiplexers)

FET multiplexers contain isolation relays which allow the accessory to be isolated from the HP 3852A backplane. Isolation relays can be used to reduce leakage current on the backplane for critical measurements or when the backplane is used for voltages greater than FET multiplexer specifications. These relays will open automatically if an overvoltage situation occurs.

Channel 90 controls the isolation relays for the FET multiplexers. For normal operation, the isolation relays must be closed to enable the accessory to be used for measurements on the backplane.

Data Returned

None

Related Commands

CLOSE, CLOSE?, SCAN, MEAS

Examples

Example: Using OPEN When Making 4-Wire Ohms Measurements

This program uses OPEN together with the CLOSE command to make 4-wire ohms measurements using a 20-channel relay multiplexer and the HP 44701A voltmeter. After each measurement, the reading is stored into a mainframe array.

```
 10 OUTPUT 709;"REAL RESRDGS(1)"       !Declare array for 2 readings
 20 OUTPUT 709;"USE 700"               !Use the vm in slot 7
 30 OUTPUT 709;"CONF OHMF"             !Configure vm for 4-wire ohms
 40 OUTPUT 709;"CLOSE 0,10,91,93"      !Close channels and tree relays
 50 OUTPUT 709;"TRIG SGL"              !Trigger the voltmeter
 60 OUTPUT 709;"CHREAD 700 INTO RESRDGS"
 70                                    !Store reading in mf memory
 80 OUTPUT 709;"OPEN 0,10"             !Open channels
 90 OUTPUT 709;"CLOSE 1,11"            !Close next channels
100 OUTPUT 709;"TRIG SGL"              !Trigger the voltmeter
110 OUTPUT 709;"CHREAD 700 INTO RESRDGS"
120                                    !Store reading
130 OUTPUT 709;"OPEN 1,11,91,93"       !Open channels and tree relays
140 END
```

OPEN

- **HP 44717A 10 Bridge 120Ω Static Strain Gage Relay Multiplexer**
- **HP 44718A 10 Bridge 350Ω Static Strain Gage Relay Multiplexer**
- **HP 44719A 10 Bridge 120Ω Static Strain Gage FET Multiplexer**
- **HP 44720A 10 Bridge 350Ω Static Strain Gage FET Multiplexer**

Description

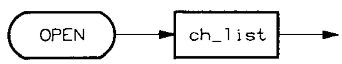
Open strain gage multiplexer channels. OPEN opens a single multiplexer channel or a list of multiplexer channels. This low-level command is necessary for accessing the "internal" strain gage channels associated with the configuration diagnostics.

Prerequisites

The strain gage accessories require mainframe firmware revision 2.0 or greater.

Syntax

OPEN *ch_list*



rr_83

Parameters

ch_list Address of the bridge completion channel, internal channel, or tree/isolation relay. Channel and diagnostic definitions for the strain gage accessories are given below:

Channel	Definition
0-9	Bridge Completion Channels. Closing channels 0-9 and 91 (and 90 if FET multiplexer) with the CLOSE command ties the bridge completion channels to the sense bus on the mainframe backplane. When performing 2-wire ohms measurements using low-level commands (e.g. OPEN, CLOSE), channel 94 must also be closed.
10	Bridge Excitation Voltage. Closing internal channels 10 and 91 (and 90 if FET multiplexer) with the CLOSE command enable the bridge excitation voltage to be measured using low-level commands (e.g. TRIG, CHREAD). Note that specifying STRVEX in the CONFMEAS, MEAS, and MONMEAS commands performs the same measurement.
11, 12	Shunt Verification. These channels are closed automatically by specifying either STRQTEN or STRQCOMP in the CONFMEAS or MEAS command. Specifying STRQTEN (tension shunt) closes channel 11 which places a resistor in parallel with the upper leg internal bridge resistor to simulate a known value of tensile strain. Specifying STRQCOMP (compression shunt) closes channel 12 which places a resistor in parallel with

Channel	Definition
	the strain gage on the bridge completion channel to simulate a known value of compressive strain.
13	Gage Isolation. Closing internal channels 13, 91, and 94 (and 90 if FET multiplexer) enable you to check the isolation between the strain gage and the test specimen. If the specimen is a conductor, the parallel combination of isolation on all bridge completion channels (0-9) is checked. If the specimen is not a conductor, only the isolation of channel 0 is checked.
14	Guard Voltage. Closing internal channels 14 and 91 (and 90 if FET multiplexer) enable you to measure the guard voltage to ensure that the guard lead is biased to the necessary potential to reduce stray leakage currents.
15, 16	Internal Half Bridge Voltage. Closing internal channels 15 or 16, and 91 (and 90 if FET multiplexer) enable you to measure the voltage on the upper and lower legs of the internal half bridge. This checks the stability of the 500Ω bridge completion resistors. Channel 15 measures the upper bridge leg and channel 16 measures the lower bridge leg.
17, 18, 19	Leadwire Resistance. Closing internal channels 17 or 18 or 19, and 91 (and 90 if FET multiplexer) enable you to indirectly determine the leadwire resistance of a 1/4 bridge arrangement by measuring the voltage on the leadwire from the strain gage to the bridge completion circuit. Knowing the leadwire resistance allows leadwire desensitization corrections to be made.
90	FET Isolation Relay. The FET strain gage multiplexers contain an isolation relay that isolates the accessory from the mainframe backplane. In addition to closing the tree switches (channels 91 and 94), the isolation relay must also be closed to tie the multiplexer channels to the backplane. (CONFMEAS, MEAS, and MONMEAS close channel 90 automatically.)
91	Volts Tree Relay/Switch. Closing channel 91 ties all accessory channels (internal and bridge completion) to the sense bus on the mainframe backplane. (Channel 91 is closed automatically by the CONFMEAS, MEAS, and MONMEAS commands.)

Channel	Definition
94	Resistance Tree Relay/Switch. Closing channel 94 ties all channels to the source bus on the mainframe backplane. This enables a voltmeter accessory to perform a 4-wire ohms measurement of gage isolation or 2-wire ohms measurements on the bridge completion channels. (Channel 94 is closed automatically by the CONFMEAS, MEAS, and MONMEAS commands when an OHMs function is specified.)

Remarks

Opening Internal Strain Gage Channels

Internal channels are opened in the order specified by the *ch_list* parameter. Channels closed by the CLOSE command remain closed until opened with the OPEN command, forced open by closing another channel on the same bank (see CAUTION on CLOSE command), or following the execution of the SCAN or MEAS command. Resetting the instrument (RST) or resetting the accessory (RST <slot>) also opens any closed channels.

Data Returned

None

Related Commands

CLOSE, SCAN, SADV

Examples

Example: Using OPEN after Measuring the Internal Half Bridge Voltages

This program shows how the OPEN command is used to open internal channels closed by the CLOSE command to measure the internal half bridge voltages. Other commands are included to show the context in which the OPEN command is used and when a channel is opened relative to the measurement.

```
10 REAL Half_bridgeup              !Real variables are declared
20 REAL Half_bridgelow
30 OUTPUT 709;"USE 700"            !use the voltmeter in slot 7
40 OUTPUT 709;"CONF DCV"           !configures the voltmeter
50 OUTPUT 709;"CLOSE 415,491"      !closes channel corresponding to
                                   !the upper bridge leg
60 OUTPUT 709;"TRIG SGL"           !triggers the voltmeter
70 OUTPUT 709;"CHREAD 700"         !retrieves reading from voltmeter
80 ENTER 709;Half_bridgeup
90 OUTPUT 709;"OPEN 415"
100 OUTPUT 709;"CLOSE 416"         !closes channel corresponding to
                                   !the lower bridge leg
110 OUTPUT 709;"TRIG SGL"          !triggers the voltmeter
120 OUTPUT 709;"CHREAD 700"        !retrieves reading from voltmeter
130 ENTER 709;Half_bridgelow
140 OUTPUT 709;"OPEN 416,491"      !opens channels 16 and 91 in slot 4
150 Print Half_bridgeup            !upper bridge leg voltage displayed
160 PRINT Half_bridgelow           !lower bridge leg voltage displayed
170 END
```

Example: Using OPEN after Measuring the Guard Voltage

This program shows how the OPEN command is used to open the internal channels closed for the Guard Voltage diagnostic.

```
10 REAL Guard_voltage               !a REAL variable is declared
20 OUTPUT 709;"USE 700"             !use the voltmeter in slot 7
30 OUTPUT 709;"CONF DCV"            !configures the voltmeter
40 OUTPUT 709;"CLOSE 414,490,491"   !closes guard voltage channel
                                    !(FET accessory)
50 OUTPUT 709;"TRIG SGL"            !triggers the voltmeter
60 OUTPUT 709;"CHREAD 700"          !retrieves reading from voltmeter
70 ENTER 709;Guard_voltage
80 PRINT Guard_voltage              !guard voltage is displayed
90 OUTPUT 709;"OPEN 414,490,491"    !opens channels 14, 90, and 91
100 END
```

- **HP 44724A 16-Channel Digital Output**
- **HP 44725A 16-Channel General Purpose Switch**
- **HP 44728A 8-Channel Relay Actuator**
- **HP 44729A 8-Channel Power Controller**

Description

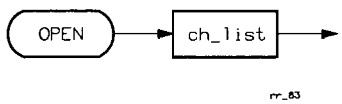
Open digital output and actuator channels.

Prerequisites

None

Syntax

OPEN *ch_list*



Parameters

ch_list Address of channel list. See Glossary. Channel number range = ES00 - ES15 (HP 44724A and HP 44725A) or ES00 - ES07 (HP 44728A and HP 44729A).

Remarks

Channels Not Addressed by OPEN

All channels not addressed by OPEN remain in previous state. Channels are opened in the order given in *ch_list*.

Data Returned

None

Related Commands

CHWRITE, WRITE, CLOSE

Examples

Example: Using OPEN to Open Digital Output Channels

This program line uses the OPEN command to open channel 0 and channels 3 through 6 of a 16-channel digital output in slot 4 of mainframe.

```
160 OUTPUT 709;"OPEN 400,403-406"
```

- **HP 44730A 4-Channel Track/Hold with Signal Conditioning**
- **HP 44732A 4-Channel 120 Ohm Strain Gage FET Multiplexer**
- **HP 44733A 4-Channel 350 Ohm Strain Gage FET Multiplexer**

Description

Open a single channel or a list of channels. OPEN is a low-level command intended for individual switch control in special signal-routing applications.

NOTE

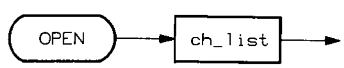
Using OPEN is not the easiest way to do routine measurements since the isolation relays are not automatically opened after the measurement.

Prerequisites

Mainframe firmware revision 3.5 or greater required to use OPEN with an HP 44730A, HP 44732A, or HP 44733A accessory. The system must not be scanning with the HP 44702A/B voltmeter when OPEN is executed.

Syntax

OPEN *ch_list*



Parameters

ch_list Address of channel list. See Glossary.

CC Range	Definitions
0 - 3	User Inputs
4 - 7	Exciation Voltage
90	Isolation Relay

Remarks

Ribbon Cable Operation

OPEN (and CLOSE and CLOSE?) can be used with an HP 44730A, HP 44732A, or HP 44733A even if the ribbon cable is connected between the accessory and an HP 44702A/B voltmeter.

Opening Accessory Channels

Channels are opened in the order listed in the *ch_list* parameter. All channels (and isolation relays) are opened at power-on. RST opens all channels in the HP 3852A, while RST *slot* opens all channels on the HP 44730A, HP 44732A, or HP 44733A in the slot addressed.

Controlling Isolation Relays

The HP 44730A, HP 44732A, and HP 4473A accessories contain isolation relays which allow the accessory to be isolated from the HP 3852A backplane. Isolation relays can be used to reduce leakage current on the backplane for critical measurements or when the backplane is used for voltages greater than FET multiplexer specifications. These relays will open automatically if an overvoltage situation occurs.

Channel ES90 controls the isolation relays for the HP 44730A, HP 44732A, and HP 44733A accessories. For normal operation, the isolation relays must be closed to enable the accessory to be used for measurements on the backplane.

Data Returned

None

Related Commands

CLOSE, CLOSE?, MEAS, SCAN

Examples

Example: Open Channel Using OPEN

This program uses an HP 44701A voltmeter to measure the +4.6 V source voltage for channel 500 of an HP 44730A, HP 44732A, or HP 44733A via the mainframe backplane and then opens the channel. The CLOSE command is used to close the channel and the isolation relay and the OPEN command is used to open the channel and isolation relay. For this program, the EXC and EXC SENSE terminals must be connected.

NOTE

Although this measurement is via the backplane, since CLOSE and OPEN are used, the measurement can still be made even if the ribbon cable is connected between an HP 44730A, HP 44732A, or HP 44733A and an HP 44702A/B.

```
10 OUTPUT 709;"USE 100"                    !Use 44701A vm in slot 100
20 OUTPUT 709;"CLOSE 504,590"              !Close ch 500, iso relay
30 OUTPUT 709;"CONF DCV"                   !Set DC volts measurement
40 OUTPUT 709;"TRIG SGL"                   !Trigger voltmeter
50 OUTPUT 709;"CHREAD 100"                 !Read ch 500 exc voltage
60 ENTER 709;A                             !Enter ch 500 exc voltage
70 PRINT "Ch 500 source =;A;"Volts"        !Display ch 500 exc voltage
80 OUTPUT 709;"OPEN 504,590"               !Open ch 500, iso relay
90 END
```

A typical return follows.

```
Ch 500 source = 4.62116 Volts
```

• **Logical Operator**

Description

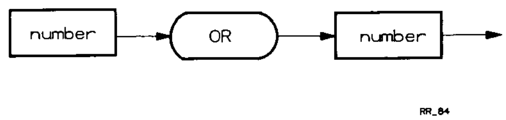
Command parameter evaluated as a numeric expression. Returns a TRUE (1) or FALSE (0) depending on the inclusive-OR result of the *numbers*.

Prerequisites

The OR statement is only used in an IF...END IF or in a WHILE...END WHILE construct which, in turn, must be included in an HP 3852A subroutine.

Syntax

number **OR** *number*



Parameters

number Constant, variable, or numeric expression. Result returned is TRUE (1) or FALSE (0) depending on the inclusive-OR of the two *numbers* as shown, where A and B are INTEGER or REAL numbers.

A / B	= 0	<> 0
= 0	F	T
<> 0	T	T

Remarks

AND Priority Higher than OR Priority

In firmware revision 3.5 and greater, the AND operator has a higher priority than the OR operator. Thus, an expression such as: A OR B AND C is interpreted as A OR (B AND C). In firmware revisions prior to 3.5, the AND and OR operators have the same priority.

Examples

Example: OR Statement in IF...END IF Loop

The following program shows one use of the OR statement in an IF...END IF loop. In line 50, the value of A OR B is calculated. If the A value is <>0 or the B value is <>0, "TRUE" is displayed. If both values = 0, "FALSE" is displayed.

On the first loop, since A = 3 and B = 0, A OR B > 0, the expression evaluates to 1 (TRUE) and "TRUE" is displayed. On the second loop, since A = 0 and B = 0, the expression evaluates to 0 (FALSE), and "FALSE" is displayed. FASTDISP OFF is set so that you can see the two displays.

```
 10 OUTPUT 709;"RST"                    !Reset the HP 3852A
 20 OUTPUT 709;"FASTDISP OFF"           !Turn fast display off
 30 OUTPUT 709;"INTEGER A,B"            !Define variables A and B
 40 OUTPUT 709;"SUB DEC"                !Start of subroutine
 50 OUTPUT 709;"  IF A OR B THEN"       !Start loop
 60 OUTPUT 709;"    DISP 'TRUE'"        !Disp TRUE if A <> 0 or B <> 0
 70 OUTPUT 709;"  ELSE"
 80 OUTPUT 709;"    DISP 'FALSE'"       !Disp FALSE if A = 0 and B = 0
 90 OUTPUT 709;"  END IF"               !End loop
100 OUTPUT 709;"SUBEND"                 !End subroutine
110 OUTPUT 709;"A=3;B=0"                !Define A and B values
120 OUTPUT 709;"CALL DEC"               !Call subroutine
130 OUTPUT 709;"A=0"                    !Re-define A value
140 OUTPUT 709;"CALL DEC"               !Call subroutine again
150 END
```

• **Mainframe**

Description

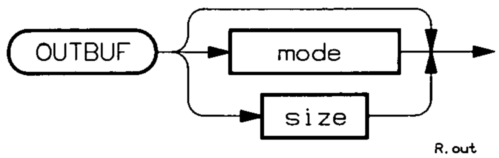
Specifies whether new data will overwrite, or be appended to the data currently in the output buffer. OUTBUF also sets the size of the output buffer.

Prerequisites

The *size* parameter of the OUTBUF command is only available with mainframe firmware revision 3.0 or greater.

Syntax

OUTBUF [*mode*] or [*size*]



Parameters

mode Overwrite, or append data to the data in the output buffer. Power-on *mode* = OFF. Default *mode* = ON.

mode	Definition
OFF	Overwrite data. Data sent to the output buffer overwrites data that is in the buffer from the previous command.
ON	Append data. Data sent to the output buffer is appended to the existing data in the buffer.

size Sets the size of the output buffer. The minimum *size* that can be specified is 5 bytes. The maximum *size* depends on the amount of mainframe memory available. Once a size has been entered with the OUTBUF command, the HP 3852A must be reset in order to load the value into the operating system. Cycling power sets the output buffer size to 1029 bytes.

Remarks

Setting the Output Buffer Size

The size of the output buffer set by the OUTBUF command does not become effective until the mainframe is reset (RST). Since the buffer size is stored in volatile memory, the size stays the same until changed by subsequent OUTBUF (and RST) commands or until power is cycled.

Note that a reset erases all previously declared variables, arrays, and downloaded subroutines. Thus, before using any of the mainframe's memory, ensure that the buffer size is set as desired.

Specifying an Output Buffer Size that is Too Large

The memory used to increase the size of the output buffer is taken from the mainframe memory available to the user. If the buffer size specified is extremely large (e.g. OUTBUF 4200000), ERROR 24: ARGUMENT OUT OF RANGE is reported on execution of the OUTBUF command. If the size specified is within the argument range but not enough memory is available, ERROR 1: OUT OF MEMORY - HP-IB BUFFERS/SYMTAB is reported during the next reset. When this occurs, the output buffer is set to a default size of 1029 bytes, the input buffer is set to 198 bytes, and the symbol table size (SYMSIZE) is set to accomodate 150 entries.

Output Buffer Size at Power-On

Anytime power is cycled, the size of the output buffer is set to 1029 bytes.

Filling the Output Buffer to Capacity

Regardless of its size, when the output buffer is full the mainframe suspends execution of the command until enough data is removed from the buffer to allow the command to complete. Note that associated with the data returned by each command is a 4-byte overhead.

Potential HP 3852A/Controller Deadlock

With INBUF OFF, the controller and the HP 3852A may deadlock if multiple commands (or delimiters) are sent in a single program line and a command generates enough data to fill the output buffer. The best way to avoid potential deadlock is to send a single command (and single delimiter) per program line and to read the results as soon as possible after a data-generating command is sent.

Potential Data Error With Multiple Commands

With OUTBUF OFF and old data in the output buffer, if the controller sends a data-generating command and immediately attempts to read the results, the controller may receive the old data, the new data, or a combination of old and new data.

This is because the old data is not overwritten until the new data is ready to be output by the command. The controller should wait until the command completes before reading the results (RDY = 1 in the Status Register), or use the CLROUT command to erase any old data within the buffer.

Data Returned

None

Related Commands

INBUF, END, SYMSIZE

Examples

Example: Enabling the Output Buffer

The following program demonstrates how data is appended in the output buffer and how the data can be entered into the controller.

```
10 OUTPUT 709;"OUTBUF ON"          !Append data in the output buffer
20 OUTPUT 709;"END OFF"            !Suppress EOI to enter data with
30                                 !a single ENTER statement
40 OUTPUT 709;"VREAD 1.234"        !Send data to the output buffer
50 OUTPUT 709;"VREAD 4.321"
60 ENTER 709;A,B
70 PRINT A,B
80 END
```

When this program executes, the following is displayed:

```
1.234    4.321
```

Example: Setting the Output Buffer Size

An application for increasing the size of the output buffer is when a command is expected to generate a large amount of data or which will read a large array.

```
10 OUTPUT 709;"OUTBUF 3000"      !Set the buffer size to 3000 bytes
20 OUTPUT 709;"RST"              !Load the buffer size into the system
                 .
                 .
                 .
70 OUTPUT 709;"VREAD RGS"        !Read the data from the array to the
80                               !output buffer
```

OUTPUT

- **HP 44788A HP-IB Controller**

Description

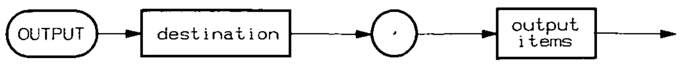
This statement outputs items to the specified destination.

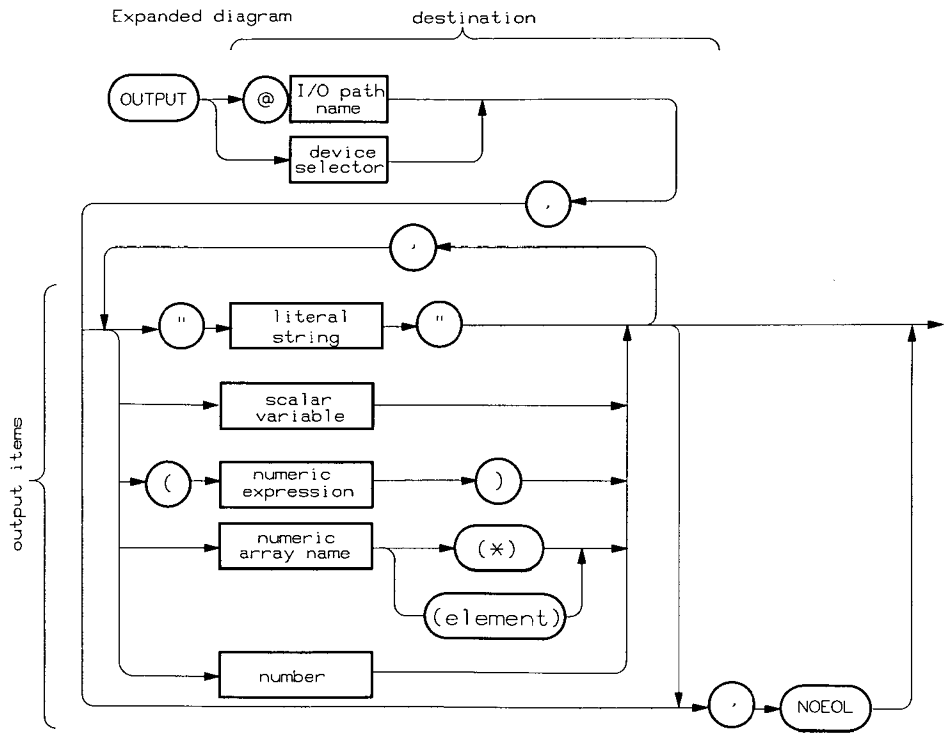
Prerequisites

Requires firmware revision 3.5 or greater.

Syntax

OUTPUT *destination output_items* **[,NOEOL]**





Parameters

destination A device or mass storage file to receive the output. Destination entries can be *I/O path name* or *device selector*.

@I/O path name

The name of the path assigned to a device or mass storage file.

device selector

The HP-IB select code (i.e. ESnn) for the device that data is to be output on. E = extender # (mainframe = 0), S = slot, and nn = device address.

output_items Text or variables that you wish to output.

NOEOL No end of line.

Remarks

Standard Numeric Format

IASC is used for integers, RASC for real numbers if they will fit otherwise, DASC is used. If ASCII output, packed data will use IASC, LASC, RASC, DASC as appropriate.

Entering Arrays

Entire arrays may be output by using the asterisk specifier. Each element in an array is treated as an item by the OUTPUT statement, as if the items were listed separately.

Serial access is used for both ASCII and BDAT files. The file pointer is set to the beginning of the file when the file is opened by an ASSIGN. The file pointer always points to the next byte to be written by serial OUTPUT operations.

The EOF pointer is read from the media when the file is opened by an ASSIGN. On a newly-created file, EOF is set to the beginning of the file. After each OUTPUT operation, the EOF is updated to maximum of the file pointer or the previous EOF value.

When data is written to an ASCII file, each item is sent as an ASCII representation with a 2-byte length header. Data sent to a BDAT file is sent in internal format if FORMAT is OFF, and is sent as ASCII characters if FORMAT is ON. (See "Devices as Destination" for a description of these formats).

Devices as Destination

An I/O path or a device selector may be used to direct OUTPUT to a device. If a device selector is used, the default system attributes are used (see ASSIGN). If an I/O path is used, the ASSIGN statement used to associate the I/O path with the device also determines the attributes used.

If FORMAT ON is the current attribute, the items are sent in ASCII. Items are sent with nothing following them. A CR/LF is sent after the last item unless

NOEOL is used or EOL OFF was specified in the ASSIGN @path TO device (if path name used as output destination). Trailing NOEOL eliminates the automatic EOL.

If FORMAT OFF is the current attribute, items are sent to the device in the HP 44788A format. Two bytes are sent for each INTEGER, eight bytes for each REAL. Each literal output consists of a four byte header containing the length of the string, followed by the actual literal characters. If the number of characters is odd, an additional byte containing a blank is sent after the last character. No CR/LF is sent regardless of NOEOL or EOL OFF.

Display as Destination

If the device selector is 1, the OUTPUT is directed to the HP 3852A display.

Data Returned

None

Related Commands

None

Examples

Example: Typical OUTPUT Statements

The following statements show ways to output data items to a device. The first statement is entered from the HP 3852A front panel, while the second statement is entered from the system controller.

```
OUTPUT @File,Array(*)                               !Output over I/O path
100 OUTPUT 709;"OUTPUT 401,'ID number is',Number"   !Output data to 401
```

- Mainframe

Description

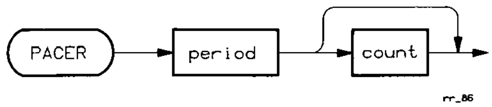
Sets the repetition period and number of pacer pulses to be output from the PACER OUT BNC connector.

Prerequisites

None

Syntax

PACER *period* [*count*]



Parameters

period Repetition period of multiple pacing pulses in seconds. The range for *period* is 1 µsec to 4.19430375 seconds. The value entered is rounded to 250 nsec steps. At power-on, *period* = 1 µsec.

count Number of output pulses to occur following a pacer trigger (PTRIG). The default and power-on *count* is a continuous train of pulses. *count* = 0 will stop the pacer. *count* range = 0 to 65535.

Remarks

Executing PACER During an Ongoing Pacer Output

Sending PACER halts any ongoing pacer output, requiring another pacer trigger (PTRIG) to restart the pulse train.

Related Commands

PTRIG, PDELAY

Examples

Example: Setting Pacer Output Pulses

This program segment sets the HP 3852A so that five pacer pulses are output from the PACER OUT BNC at 4 µsec intervals. There is a 10 µsec delay from the time of an external trigger (into the PACER TRIGGER IN BNC) to the first pacer pulse output.

PACER (cont)

```
External                          Pacer Pulses
Trigger

   |                     |      |      |      |      |
____________________________________________________________

     10 usec delay         4 us     4 us   4 us     4us
```

```
        .
        .
50 OUTPUT 709;"PACER 4.0E-6,5"    !Output 5 pulses 4 usec apart
60 OUTPUT 709;"PDELAY 10.0E-6"    !Delay first pulse 10 usec after trigger
70 OUTPUT 709;"PTRIG EXT"         !Set HP 3852A to trig on ext trigger
        .
        .
```

• Mainframe

Description

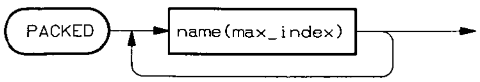
Declares a PACKed array in mainframe memory. The PACKED command declares arrays only.

Prerequisites

None

Syntax

PACKED *name (max_index)* [*name (max_index)*. . .]



rr_87

Parameters

name (max_index)

Specify PACKED array *name* and *max_index*. *name* specifies the name of a PACKED array. *name* must be a string ≤ eight characters. It cannot begin with a digit and can contain only letters, digits, '_', and '?'.

max_index specifies the number of bytes required to store data in the desired packed format. *max_index* is determined by multiplying the bytes/reading for the packed format you are going to receive, times the maximum number of readings you plan to take.

Remarks

CAT Command and PACKED Arrays

When a PACKED array has been declared, the CAT command returns the maximum number of bytes allocated to the array if no data has been stored in the array. If data has been stored in the array, CAT returns the maximum number of readings which can be stored in the array in that format. For example, a PACKED array with *max_index* = 99 contains 100 bytes for storing data. If no data is in the array, the CAT command returns 100 (the number of bytes). If packed readings are stored which are 2 bytes/reading, the CAT command returns 50 (= 100 bytes/2 bytes per reading), indicating a maximum of 50 readings can be stored.

Different Data Formats Cannot be Stored in the Same PACKED Array

When storing data in a packed array, the bytes/reading for the packed format must be the same and contain the same bit pattern. This means packed readings 2 bytes/reading cannot be stored in an array with packed readings 4 bytes/reading. Also, the mainframe conversion (unpacking) routines require readings the same length to have the same bit pattern (designation).

Redeclaring Arrays

Executing PACKED declares a packed array and fills the elements with zeroes. A PACKED array, if redeclared, must remain a PACKED array. A redeclared array will be set up with zeroes in all elements. (If the array is redeclared to the same size, it will just be zeroed.) After redeclaring, any packed format length can be stored.

Arrays are Global

All arrays are global. Therefore, arrays dimensioned outside a subroutine are accessible in the subroutine and vice-versa.

PACKED Command Stored in a Subroutine

If a PACKED command is stored in a subroutine, the array name and type are defined immediately but data storage for the array won't be set up until the subroutine executes. (The name can't be used for a subroutine name, another array name, or a variable name.)

Recovering Storage Space With DELVAR

Send the DELVAR command to recover storage space allocated by the PACKED command. The array name and type remain, but the memory allocated to the array is released. That is, the array becomes a PACKED array with zero bytes.

Data Returned

None

Related Commands

DIM, REAL, INTEGER, DELVAR, SCRATCH

Examples

Example: Declaring a PACKED Array

The following program declares a PACKED array to store 10 readings from the HP 44702A/B voltmeter. Since each reading is two bytes long and 10 readings are taken, the array must be assigned a minimum of 20 bytes.

```
10 OUTPUT 709;"PACKED HSRDGS(19)"  !Declares a PACKED array (20 bytes)
20 OUTPUT 709;"USE 500"            !Use the voltmeter in slot 5
30 OUTPUT 709;"CONFMEAS DCV 400-409 INTO HSRDGS"
40                                 !Take readings, store in memory
50 END
```

- **HP 44723A 16-Channel High-Speed Digital Sense/Control**

Description

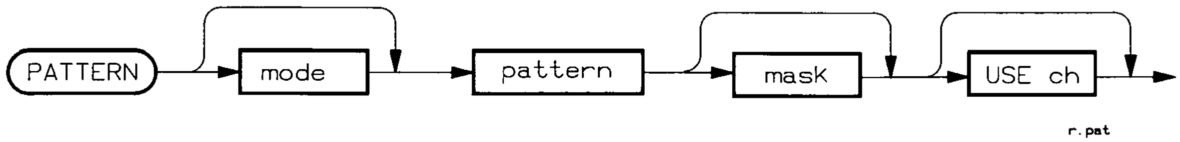
Specifies the input pattern and mask which will generate a pattern interrupt when enabled.

Prerequisites

Requires mainframe firmware revision 3.0 or greater.

Syntax

PATTERN [*mode*] *pattern* [*mask*] [USE *ch*]



Parameters

mode Sets the pattern interrupt mode. Valid *modes* are EQU (power-on, reset, and default) and NEQ. When enabled, *mode* = EQU generates a pattern interrupt when the input channel bit pattern for the channels specified by *mask* is the same as the bit pattern specified by *pattern*. Or, when enabled, *mode* = NEQ generates a pattern interrupt when the input channel bit pattern for the channels specified by *mask* is not the same as the bit pattern specified by *pattern*.

pattern Sets the bit pattern required to generate a pattern interrupt when enabled. Power-on and reset *pattern* = 0.

mask Sets the mask for pattern interrupts. A 1 bit in the mask includes the corresponding channel in the pattern, while a 0 bit omits the channel. Power-on and reset *mask* = 0. Default *mask* = -1 (all channels included).

USE *ch* For *ch* = ES90, sets the HP 44723A in the slot specified by ES for pattern interrupt.

Remarks

Generating Pattern Interrupts

Enabling pattern interrupts will generate an immediate interrupt if the input channel states match the mask/pattern. A pattern interrupt may be generated from even a momentary concidence of the input channel states and the programmed mask/pattern. This may occur when multiple input bits change "simultaneously" due to propagation delay times in the hardware.

Pattern Interrupts Cleared/Disabled

Pattern interrupts are generated from user input channel data. Pattern interrupts are cleared and disabled when serviced by the mainframe. DISABLE INTR clears and disables pattern interrupts.

Data Returned

None

Related Commands

ENABLE INTR

Examples

Example: Set Pattern Interrupt

This program enables an HP 44723A in slot 2 of the mainframe to generate a pattern interrupt on input channel pattern xxxx 1111 xxxx 0000. The *pattern* parameter is set for 0000 1111 0000 0000 (3840), the *mask* parameter is set for 0000 1111 0000 1111 (3855), and the *mode* parameter is set for EQU (equal). Since the mask enables only channels 0 through 3 and 8 through 11, when the bit pattern is xxxx 1111 xxxx 0000, a pattern interrupt is generated.

The program loops until an SRQ is generated. When the input channel states matches the pattern set, a pattern interrupt occurs which generates the SRQ. The interrupt is serviced by controller subroutine Results. When the program completes, the time of the interrupt is displayed.

When the interrupt occurs, the HP-IB SRQ line is set TRUE and an SRQ is sent to the controller. Also, the INTR bit (bit 9) and the service request bit (bit 6) in the status register are set. Because the interrupt is handled by the controller, both bits must be cleared (STA? clears bit 9, SPOLL clears bit 6) before the controller can respond to the next interrupt that occurs.

```
10   ON INTR 7 GOTO Results              !Call sub on interrupt
20   ENABLE INTR 7;2                     !Enable controller intr on SRQ
30   OUTPUT 709;"USE 290"                !Set for pattern interrupt
40   OUTPUT 709;"RQS ON;RQS INTR"        !Enable interrupt on SRQ
50   OUTPUT 709;"STA?"                   !Clear FPS,LCL,INTR,LMT,ALRM bits
60   OUTPUT 709;"CLROUT"                 !Clear STA? data from output buffer
70   OUTPUT 709;"PATTERN EQU,3840,3855"  !Set pattern for interrupt
80   OUTPUT 709;"ENABLE INTR"            !Enable accessory to interrupt
90   OUTPUT 709;"ENABLE INTR SYS"        !Enable mainframe to sense intr
100  GOTO 100                            !Loop until interrupt occurs
110  Results:  !                         !Start controller subroutine
120  OUTPUT 709;"TIME"                   !Query time of day
130  ENTER 709;T                         !Enter time of day
140  PRINT "Slot 2 intr @ ";TIME$(T)     !Display interrupt time/message
150  A=SPOLL(709)                        !Read/clear SRQ bit
160  STOP                                !End controller subroutine
170  END
```

When the interrupt occurs (input channel states match the pattern/mask), an interrupt is generated and the controller queries the time of day and enters the time. A typical return is:

```
Slot 2 intr @ 02:46:50
```

• **Mainframe**

Description

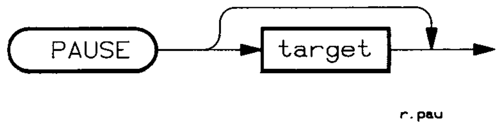
Pauses HP 3852A subroutine execution.

Prerequisites

The *target* parameter is used to pause run task subroutines in the multitasking mode. This parameter and multitasking are only available with mainframe firmware revision 3.0 or greater.

Syntax

PAUSE [*target*]



Parameters

target Number of the run task containing the subroutine to be paused. The range for *target* is 0 to 7.

If *target* is not specified, the subroutine in which the PAUSE command was executed is paused.

Remarks

Executing the PAUSE Command

When the *target* parameter is specified, the PAUSE command can be executed from the front panel and HP-IB as well as within a subroutine. When *target* is not specified, PAUSE must be executed within the subroutine to be paused.

Continuing a Paused Subroutine

Once a subroutine has been paused by the PAUSE command or as a target of the PAUSE command, the subroutine is subsequently continued by the CONT command. If the subroutine was paused without the *target* parameter, you can "step" through the remainder of the program with the STEP command. You cannot step through a subroutine paused by the *target* parameter.

Using PAUSE in Nested Subroutines

In the power-on mode or in the front panel, HP-IB, and interrupt tasks of the multitasking mode, PAUSE without *target* cannot be located in a nested subroutine or in a subroutine called more than once. PAUSE without *target* can be located in a nested run task subroutine. PAUSE *target* can be in a nested subroutine in the front panel, HP-IB, and interrupt environments so long as the subroutine does not target itself. Run task subroutines can target themselves.

Paused Subroutines Allow Time-Slicing to Continue

If the HP 3852A is in the multitasking mode and a subroutine is executing within a task (front panel, HP-IB, interrupt, run task) with a priority equal to other tasks, time-slicing will continue while the subroutine is paused. When the subroutine is continued, it is re-scheduled and resumes time-slicing with the other tasks.

Data Returned

None

Related Commands

CONT, STEP, CALL, SUB, SUBEND, RUN

Examples

Example: Pausing a Subroutine

This program stores subroutine PAUSESQR in the HP 3852A memory, then calls the subroutine with the CALL command. The first time the subroutine is called, the value 0.000000E+00 is displayed on the front panel. The first time the subroutine is continued (CONT in line 210), the square root of 1 (1.000000E+00) is displayed. The second time the subroutine is continued (CONT in line 220), the square root of 2 (1.414214E+00) is displayed.

```
10 OUTPUT 709;"RST"               !Reset the HP 3852A
20 OUTPUT 709;"REAL R"            !Define REAL variable R
30 OUTPUT 709;"SUB PAUSESQR"      !Start subroutine
40 OUTPUT 709;"FOR R = 0 TO 9"    !Loop 10 times
50 OUTPUT 709;"DISP SQR(R)"       !Display sq root of R
60 OUTPUT 709;"PAUSE"             !Pause subroutine
70 OUTPUT 709;"NEXT R"            !Repeat loop
80 OUTPUT 709;"SUBEND"            !End subroutine
          .
          .
200 OUTPUT 709;"CALL PAUSESQR"    !Displays 0 (sq root of 0)
210 OUTPUT 709;"CONT"             !Displays 1 (sq root of 1)
220 OUTPUT 709;"CONT"             !Displays 1.414.. (sq root of 2)
230 END
```

Example: Targeting a Subroutine to be Paused

This example shows how the PAUSE command is directed to a run task subroutine.

In the program, the HP 3852A is placed in the multitasking mode and two subroutines are downloaded and directed to run tasks. Subroutine A (run task 1) counts from 0 to 199 and subroutine B (run task 2) repeatedly displays A B C.

When the mainframe first swaps to run task 2, the PAUSE command is directed to run task 1 (line 280). This pauses subroutine A. When the mainframe swaps back to run task 1, it detects that subroutine A has been paused and immediately swaps back to run task 2. Subroutine B resumes execution and prior to finishing, directs the CONTinue command (line 340) to run task 1 which enables subroutine A to continue.

```
10    !Set a 1 second time-slice and set the number of run tasks and
20    !queue size the system will allow. Place the HP 3852A in the
30    !multitasking mode. The reset caused by ENABLE MULTI will load
40    !the TSLICE and NTASKS setting into the operating system.
50    !
60    OUTPUT 709;"TSLICE 1"
70    OUTPUT 709;"NTASKS 2,2"
80    OUTPUT 709;"ENABLE MULTI"
90    !
100   !Write and download the subroutines to be directed to run tasks.
110   !Subroutine A when active, counts from 0 to 199.
120   !
130   OUTPUT 709;"SUB A"
140   OUTPUT 709;"  INTEGER I"
150   OUTPUT 709;"  FOR I=0 TO 199"
160   OUTPUT 709;"    DISP I"
170   OUTPUT 709;"  NEXT I"
180   OUTPUT 709;"SUBEND"
190   !
200   !Download subroutine B. When active, the subroutine directs the
210   !PAUSE command to run task 1, thus pausing subroutine A.
220   !Subroutine B then displays the sequence A B C 300 times. When
230   !finished, the subroutine directs the CONT command to run task
240   !1 which enables subroutine A to continue.
250   !
260   OUTPUT 709;"SUB B"
270   OUTPUT 709;"  INTEGER J"
280   OUTPUT 709;"  PAUSE 1"
290   OUTPUT 709;"  FOR J=0 TO 299"
300   OUTPUT 709;"    DISP 'A'"
310   OUTPUT 709;"    DISP 'B'"
320   OUTPUT 709;"    DISP 'C'"
330   OUTPUT 709;"  NEXT J"
340   OUTPUT 709;"  CONT 1"
350   OUTPUT 709;"SUBEND"
360   !
370   !Direct the subroutines to run tasks. Set each run task to execute
380   !one time.
390   !
400   OUTPUT 709;"RUN 1 A"
410   OUTPUT 709;"RUN 2 B"
420   END
```

As this program executes, subroutine A counts to approximately 70 before the mainframe swaps to run task 2. Since subroutine A is then paused by subroutine B (run task 2), the mainframe displays A B C the programmed number of times. As subroutine B finishes, subroutine A is continued and the counting resumes.

PDELAY

- **Mainframe**

Description

Sets the delay time between the pacer trigger (see the PTRIG command) and the first pacer pulse output from the PACER OUT BNC connector.

Prerequisites

None

Syntax

PDELAY *trigger_delay*

RR_69

Parameters

trigger_delay — Time in seconds between the trigger and the output of the first pacer pulse. The *trigger_delay* range is 500 nsec to 4.19430375 seconds. The values entered are rounded to 250 nsec steps. At power-on, *trigger_delay* is 500 nsec.

Remarks

Executing PDELAY During an Ongoing Pacer Output

Sending PDELAY halts any ongoing pacer output, requiring another pacer trigger (PTRIG) to restart the pulse train.

Related Commands

PTRIG, PACER

Examples

Example: Setting Delay for Pacer Output Pulses

This program segment sets the HP 3852A so that five pacer pulses are output from the PACER OUT BNC at 4 µsec intervals. There is a 10 µsec delay from the time of an external trigger (into the PACER TRIGGER IN BNC) to the first pacer pulse output.

```
External                    Pacer Pulses
Trigger

   |                 |      |      |      |      |
 ____________________________________________________

        10 usec delay     4 us    4 us   4 us    4us
```

```
        .
        .
50 OUTPUT 709;"PACER 4.0E-6,5"    !Output 5 pulses 4 usec apart
60 OUTPUT 709;"PDELAY 10.0E-6"    !Delay first pulse 10 usec after trigger
70 OUTPUT 709;"PTRIG EXT"         !Set HP 3852A to trig on ext trigger
```

• HP 44702A/B High-Speed Voltmeter (Scanner or System Mode)

Description

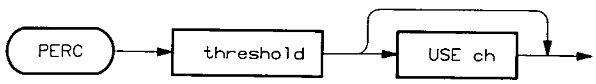
Sets the trigger threshold level to the specified percentage of the voltmeter's full-scale range.

Prerequisites

The threshold trigger mode must be enabled (see SCTRIG, STTRIG, and TRIG commands). If the voltmeter is in Scanner Mode, SCTRIG must be set to HOLD before setting PERC.

Syntax

PERC *threshold* [USE *ch*]



Parameters

threshold Input threshold value as a percentage of the voltmeter full-scale range. *threshold* must be a number between -128 and +127. Resolution is 1%. At power-on, *threshold* = 0.

USE *ch* Voltmeter slot number. See Glossary.

Remarks

Using Autoranging

If autoranging is used, the absolute trigger threshold value will change, based on the current range. Although the same percentage is used, there is a different full scale value for each range, so a different absolute value results.

Threshold Values Without Autoranging

The following table lists the minimum (-128%) and maximum (+127%) trigger thresholds allowable for the four HP 44702A/B voltage ranges when autoranging is NOT used.

full-scale range	min trig threshold	max trig threshold
0.04 V	-0.0512 V	+0.0508 V
0.32 V	-0.41 V	+0.406 V
2.56 V	-3.28 V	+3.25 V
10.24 V	-13.1 V	+13.0 V

Data Returned

None

Related Commands

RANGE, SLOPE, SCSLOPE, STSLOPE, TRIG, SCTRIG, STTRIG, USE

Examples

Example: Using Threshold Triggering - System Mode

This program uses the PERC command in order to "threshold" trigger the HP 44702A/B voltmeter. Since the voltmeter is set to the 10.24V range, PERC 50 sets a triggering threshold of 0.50 x 10.24V = 5.12V. Once the voltmeter is triggered, the readings are taken then transferred to the controller and displayed.

```
 10 OUTPUT 709;"RST"                    !Reset the HP 3852A and HP 44702A/B
 20 REAL Sysrdgs(0:14)                  !Declare controller array
 30 OUTPUT 709;"USE 500"                !Use the voltmeter in slot 5
 40 OUTPUT 709;"CONF DCV"               !Configure for DC voltage
 50 OUTPUT 709;"NRDGS 15"               !Take 15 readings when triggered
 60 OUTPUT 709;"RANGE 10"               !Set 10.24V range
 70 OUTPUT 709;"PERC 50"                !Set trigger threshold level
 80 OUTPUT 709;"SLOPE LH"               !Set triggering edge
 90 OUTPUT 709;"TRIG MEAS"              !Enable threshold triggering
100 OUTPUT 709;"MEAS DCV, 400"          !Take measurements when triggered
110 ENTER 709;Sysrdgs(*)                !Enter readings into controller
120 FOR I=0 TO 14
130 PRINT Sysrdgs(I)                    !Print readings
140 NEXT I
150 END
```

A typical output based on this program follows:

```
5.38
5.38
5.38
5.38
5.3825
5.3825
5.3825
5.3825
5.3825
5.3825
5.2825
5.385
5.385
5.385
5.385
```

Example: Threshold Triggering - Scanner Mode

This program gives an example of how threshold triggering can be used with the voltmeter in the scanner mode. The voltmeter is set to the 10.24V range and PERC is specified such that the triggering threshold is 5.12V. Once the voltmeter is triggered, the readings are stored in a PACKED array in the mainframe.

```
 10 OUTPUT 709;"RST"                        !Reset HP 3852A and HP 44702A/B
 20 OUTPUT 709;"USE 500"                    !Use the voltmeter in slot 5
 30 OUTPUT 709;"PACKED HSRDGS(199)"         !Declare packed array (100 rdgs)
 40 OUTPUT 709;"SCANMODE ON"                !Turn scanner mode on
 50 OUTPUT 709;"CONF DCV"                   !Configure for DC voltage
 60 OUTPUT 709;"NRDGS 100"                  !Take 100 readings when triggered
 70 OUTPUT 709;"RANGE 10"                   !Set 10.24V range
 80 OUTPUT 709;"PERC 50"                    !Set trigger threshold level
 90 OUTPUT 709;"SPER 10E-6"                 !Set 10 us sample period
100 OUTPUT 709;"SLOPE LH"                   !Set triggering edge
110 OUTPUT 709;"CLWRITE SENSE 400"          !Specify measurement channel
120 OUTPUT 709;"SCTRIG MEAS"                !Enable threshold triggering
130 OUTPUT 709;"XRDGS 500 INTO HSRDGS"      !Take readings when triggered,
140                                         !Store in memory
150 END
```

POS

- **HP 44714A 3-Channel Stepper Motor Controller/Pulse Output**

Description

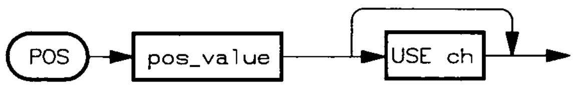
Loads a position value, in units specified by the PSCALE command, in the internal position counter. This allows the user to set the counter to any value corresponding with current position.

Prerequisites

Requires mainframe firmware revision 3.0 or greater.

Syntax

POS *pos_value* [USE *ch*]



Parameters

pos_value — The *pos_value* parameter is the position value to be loaded into the internal counter. This value can be the current motor location.

USE *ch* — Channel into which the position value is to be loaded. Channel range can be ES00 to ES02.

Remarks

Data Returned

None

Related Commands

POS?

Examples

Example: Set the Internal Counter to 25

```
10 OUTPUT 709;"POS 25 USE 201"    !Set position counter to 25
20                                !for channel 1 slot 2
30 END
```

- **HP 44714A 3-Channel Stepper Motor Controller/Pulse Output**

Description

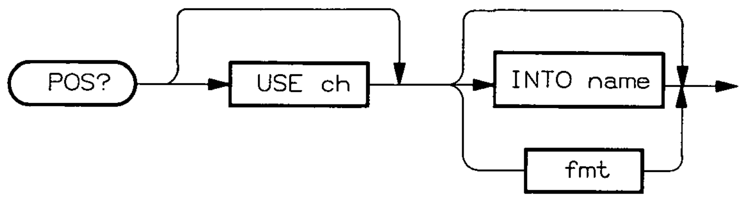
Requests the current position value present in the internal position counter. The value in the counter is updated by counting the number of pulses generated and applying the scale factor.

Prerequisites

Requires mainframe firmware revision 3.0 or greater.

Syntax

POS? [USE *ch*] [INTO *name*] or [*fmt*]



RR.QPOS

Parameters

USE *ch* — Channel queried. Channel range is ES00 to ES02.

INTO *name* — See Glossary.

fmt — See Glossary. Default format is RASC.

Remarks

Data Returned

Internal counter value (current position)

Related Commands

POS

Examples

Example: Query the Internal Counter

```
10 OUTPUT 709;"POS? USE 201"  !Query the counter of ch 1 slot 2
20 ENTER 709; VALUE           !Enter value into VALUE
30 PRINT VALUE                !Print VALUE
40 END
```

POSTSCAN

- **HP 44702A/B High-Speed Voltmeter (Scanner Mode Only)**

Description

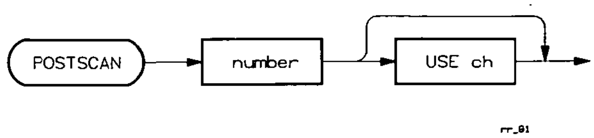
Sets the number of passes the voltmeter will make through its scan list after a stop trigger is accepted.

Prerequisites

The voltmeter must be in Scanner Mode (SCANMODE ON) and SCTRIG must be set to HOLD before POSTSCAN is executed.

Syntax

POSTSCAN *number* [USE *ch*]



Parameters

number Number of passes through the voltmeter scan list that will be made after a stop trigger is received. *number* range is 0 to 65535. At power-on, *number* = 0.

USE *ch* Voltmeter slot number. See Glossary.

Remarks

Data Returned

None

Related Commands

SCANMODE, STTRIG, PRESCAN, USE

Examples

Example: Using the POSTSCAN Command

This program sets the POSTSCAN command such that three postscans will be made after the voltmeter acknowledges its internal stop trigger. Note that the CONF command used in the program sets PRESCAN to 1 so that a total of four passes will be made through the scan list in which 4-wire ohms measurements are made.

```
 10 REAL Hsrdgs(0:19)          !Declare controller array
 20 OUTPUT 709;"RST"           !Reset the HP 3852A and HP 44702A/B
 30 OUTPUT 709;"USE 500"       !Use the volmeter in slot 5
 40 OUTPUT 709;"SCANMODE ON"   !Turn scanner mode on
 50 OUTPUT 709;"CONF OHMF100K" !Configure for 4-wire ohms
 60 OUTPUT 709;"ASCAN ON"      !Turn autoscan on (1 scan trigger needed)
 70 OUTPUT 709;"CLWRITE SEP 400-404"
 80                            !Specify ch list and ribbon cable conn.
 90 OUTPUT 709;"POSTSCAN 3"    !Make 3 postscans after stop trigger
100 OUTPUT 709;"SCTRIG SGL"    !Issue a single scan trigger
110 OUTPUT 709;"XRDGS 500"     !Transfer readings from vm to output buffer
120 ENTER 709;Hsrdgs(*)        !Enter readings into controller
130 FOR I=0 TO 19
140 PRINT Hsrdgs(I)            !Print readings
150 NEXT I
160 END
```

At typical listing of 4-wire ohms measurements based on this program is shown below:

```
1000
14912.5
11912.5
13381.25
10550
999.2188
10668.75
10931.25
11781.25
11481.25
999.2188
9812.5
9987.5
10450
10525
1000
9675
9693.75
9850
9862.5
```

POWEROFF

- **Mainframe**

Description

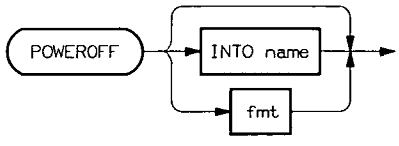
Returns the Julian date and time of day (in seconds) of the most recent HP 3852A system power loss.

Prerequisites

To return the correct date and time of the last (next) HP 3852A powerdown, the date and time functions of the real time clock should be set to the present date and current time.

Syntax

POWEROFF [INTO *name*] or [*fmt*]



Parameters

INTO *name* See Glossary.

fmt See Glossary. Default format is DASC.

Remarks

Time Set When Memory Lost

The clock time is stored in a battery backed-up memory. If memory is lost, the power-off time is set to 2.086 629 12 E+11, which represents midnight (2400 hours) on March 1, 1900.

Data Returned

Value of the real-time clock when power was most recently lost. If the clock is properly set, POWEROFF modulo 86400 gives the time (in seconds) since midnight when power was most recently lost. Resolution is 1 msec.

Related Commands

SET TIME, SET TIMEDATE, TIMEDATE

Examples

Example: Determining Time of Last Power Failure

The following program uses the HP Series 200/300 commands DATE$ and TIME$ to determine the time of the last power failure of an HP 3852A.

```
10 OUTPUT 709;"POWEROFF"                          !Query HP 3852A
20 ENTER 709;A                                    !Return fail time
30 PRINT "Last power failure: ";DATE$(A),TIME$(A) !Disp conv time
40 END
```

Assuming the real-time clock was properly set, if the last power failure occurred Aug 11, 1986 at 03:10:43 hours, a typical display is:

```
Last power failure: 11 Aug 1986      03:10:43
```

PRESCAN

- **HP 44702A/B High-Speed Voltmeter (Scanner Mode Only)**

Description

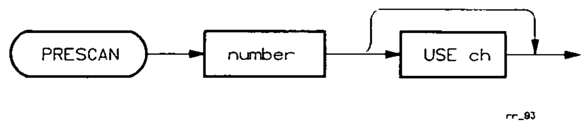
Sets the minimum number of passes the voltmeter will make through its scan list before a stop trigger is accepted.

Prerequisites

The voltmeter must be in Scanner Mode (SCANMODE ON) and SCTRIG must be set to HOLD before PRESCAN is executed.

Syntax

PRESCAN *number* [USE *ch*]



Parameters

number Minimum number of passes through the voltmeter scan list that are made before a stop trigger is accepted. *number* range is 0 to 65535. At power-on, *number* = 1.

USE *ch* Voltmeter slot number. See Glossary.

Remarks

Interaction With Stop Trigger

If both PRESCAN and POSTSCAN are set to 0, the scan sequence may return no results. If STTRIG INT is set, one pass of the scan list must always be done (even if PRESCAN 0 is set) because STTRIG INT is defined as "generate stop trigger at the end of each pass". However, if STTRIG is set to any other source, zero passes of the scan list may occur.

Data Returned

None

Related Commands

SCANMODE, STTRIG, POSTSCAN, USE

Examples

Example: Using the PRESCAN Command

This program enables an HP 44702A/B High Speed Voltmeter in slot 5 of the mainframe to make 3 passes through the voltmeter scan list before the stop trigger is received.

```
 10 REAL Rdgs(0:11)              !Declare controller array
 20 OUTPUT 709;"RST"             !Reset the HP 3852A and HP 44702A/B
 30 OUTPUT 709;"USE 500"         !Use the voltmeter in slot 5
 40 OUTPUT 709;"SCANMODE ON"     !Turn scanner mode on
 50 OUTPUT 709;"CONF DCV"        !Configure voltmeter for DC voltage
 60 OUTPUT 709;"ASCAN ON"        !Turn autoscan on (1 scan triggered rq'd)
 70 OUTPUT 709;"PRESCAN 3"       !Make 3 passes through scan list
 80 OUTPUT 709;"CLWRITE SENSE, 400-403"
 90                              !Define channel list
100 OUTPUT 709;"SCTRIG SGL"      !Issue a single scan trigger
110 OUTPUT 709;"XRDGS 500"       !Transfer readings to output buffer
120 ENTER 709;Rdgs(*)            !Enter readings into controller
130 FOR I=0 TO 11
140 PRINT Rdgs(I)                !Print readings
150 NEXT I
160 END
```

A typical return based on this program follows:

```
8.0925
1.953125E-5
-.105379
.265625
8.095
.
.
.
```

- **HP 44788A HP-IB Controller**

Description

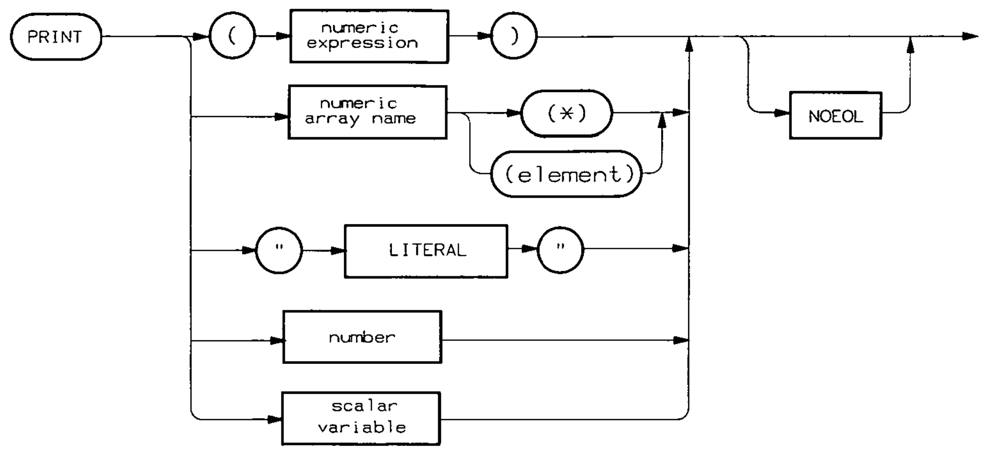
This statement sends items to the PRINTER IS device.

Prerequisites

Requires firmware revision 3.5 or greater.

Syntax

PRINT *output items* **[,NOEOL]**



Parameters

output items The string or variable to be printed.

array An array name.

NOEOL No end of line.

Remarks

Standard Numeric Format

IASC is used for integers, RASC for real numbers if they will fit otherwise, DASC is used. If the output is ASCII, packed data will use IASC, LASC, RASC, or DASC as is appropriate.

Automatic End-Of-Line Sequence

After the print list is exhausted, an End-Of-Line (EOL) sequence is sent to the PRINTER IS device, unless it is suppressed by the NOEOL parameter in the PRINT command.

Printing Arrays

Entire arrays may be printed using the asterisk specifier. Each element in an array is treated as a separate item, as if the elements were all listed.

Data Returned

None.

Related Commands

PRINTER IS

Examples

Example: Typical PRINT Statements

The following statements show ways to use the PRINT command. The first statement is entered from the HP 3852A front panel, while the second statement is entered from the system controller.

```
PRINT "LINE",Number                   !Print output items
110 OUTPUT 709;"PRINT Array(*)"       !Print array
```

PRINTER IS

• **HP 44788A HP-IB Controller**

Description

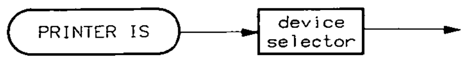
This statement specifies the system printing device.

Prerequisites

Requires firmware revision 3.5 or greater.

Syntax

PRINTER IS *device selector*



Parameters

device selector The designation of the printing device to be used. A numeric expression rounded to an integer.

Remarks

Data Sent by PRINT

The system printing device receives all data sent by the PRINT statement. The default printing device is the display of the HP 3852A (select code 1) at power-on and after reset.

Data Returned

None.

Related Commands

PRINT

Examples

Example: Establish PRINTER IS Device

These statements show ways to use the PRINTER IS command. The first statement establishes the display of the HP 3852A as the PRINTER IS device and is entered from the HP 3852A front panel. The second statement establishes device 01 as the PRINTER IS device when the HP 44788A is in slot 5 and is entered from the system controller.

```
PRINTER IS 1                     !Set HP 3852A display as PRINTER IS
110 OUTPUT 709;"PRINTER IS 501"  !Set device 501 as PRINTER IS
```

- **Mainframe**

Description

Traces the swapping of tasks during execution of a program, and reports the status of the task which was active at the time of the swap. PROBE designates the arrays which will store the data returned by the probe. ENABLE PROBE activates the probe.

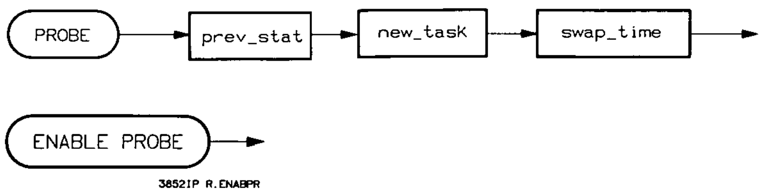
Prerequisites

Requires firmware revision 3.5 or greater. For revisions 3.5 and 3.51, executing PROBE activates the probe, as ENABLE PROBE is not a valid command.

Syntax

PROBE *prev_stat new_task swap_time*

ENABLE PROBE



Parameters

prev_stat Array in which the status of the task from which the swap occurred is stored. The status is represented by the following seven bit code:

BIT	VALUE	STATUS
0	1	The task has been created.
1	2	The task has been activated and is running.
2	4	The task has been suspended by the SUSPEND or SUSPEND UNTIL command.
3	8	An error has occurred within the task.
4	16	The task has been paused by the PAUSE command.
5	32	The task has been suspended by the WAITFOR SIGNAL command.
6	64	The task has been suspended pending the release of a lock.

The *prev_stat* array must be an Integer array and should be the same size as the *new_task* and *swap_time* arrays.

new_task Array which stores the number of the task as it is swapped to. For the probe, the tasks are numbered as follows:

NUMBER	TASK
0 through 7	Run tasks 0 through 7.
-2	Keyboard task.
-3	HPIB task.
-4	Interrupt task.
-5	Instrument ready.
-1	Device clear.

The *new_task* array must also be an Integer array and should be the same size as the *prev_stat* and *swap_time* arrays.

swap_time Array which stores the reading of the mainframe's clock at the time a swap occurs. The *swap_time* array must be a Real array and should be the same size as the *prev_stat* and *new_task* arrays.

Remarks

Determining the Size of the Arrays

Determining the size of the arrays to use with the probe is done usually through trial and error. Arrays that are too small or too large will not affect the multi-tasking program being traced. If the size of the arrays are too small, array elements are overwritten as necessary until the program finishes or until the probe is disabled.

If the program being traced finishes and the arrays have not been filled (i.e. they're too large), zeroes corresponding to the empty array elements will be returned along with the probe data.

Enabling/Disabling the Probe

PROBE, which identifies the arrays to be used, can execute at any point in the program. ENABLE PROBE is executed prior to the section of code to be traced.

The probe remains enabled until disabled by DISABLE PROBE. Note that enabling or disabling the probe does not clear the designated arrays or reset their index pointers. That is accomplished by reading or redeclaring the arrays.

Changing the Size of the Output Buffer

To avoid a deadlock between the controller and the HP 3852A when large arrays are used, it may be necessary to change the size of the output buffer. See the OUTBUF command.

Data Returned

No data from the probe is returned directly to the output buffer. The task, task status, and time are placed in mainframe arrays which can then be read into the output buffer.

Related Commands

DISABLE PROBE

Examples

Example: Tracing a Multitasking Program

The following example demonstrates how the probe is used to trace the execution of a multitasking program. (The example assumes 3.52 firmware is installed.)

Lines 10 through 110 configure this particular multitasking system and declare the arrays used by the controller, the system, and by the probe. The PROBE command in line 130 designates the arrays to be used.

Lines 150 through 200 are a subroutine which is directed to a run task. The subroutine contains a FOR/NEXT loop which suspends the task on each pass through the loop. This activity is reported as the program is traced.

The probe is enabled (line 220) just before the subroutine is directed to a run task (line 230). Line 260 disables the probe after the run task completes.

```
10    DIM P_s(0:8),N_t(0:8),P(0:8)
20    FOR J=.001 TO .065 STEP .064
30      OUTPUT 709;"TSLICE .065,";J
40      OUTPUT 709;"NTASKS 1,1"
50      OUTPUT 709;"ENABLE MULTI"
60      WAIT 1
70      OUTPUT 709;"DISABLE LABELS"
80      OUTPUT 709;"OUTBUF ON"
90      OUTPUT 709;"SET TIME 0"
100     OUTPUT 709;"INTEGER I,PRV_STAT(8),NEW_TASK(8)"
110     OUTPUT 709;"REAL PERIOD(8)"
120   !
130     OUTPUT 709;"PROBE PRV_STAT,NEW_TASK,PERIOD"
140   !
150     OUTPUT 709;"SUB A"
160     OUTPUT 709;"  FOR I =1 TO 3"
170     OUTPUT 709;"    SUSPEND .1"
180     OUTPUT 709;"  NEXT I"
190     OUTPUT 709;"  SIGNAL HPIB"
200     OUTPUT 709;"SUBEND"
210   !
220     OUTPUT 709;"ENABLE PROBE"          !3.52 and above
230     OUTPUT 709;"RUN 0 A"
240   !
250     OUTPUT 709;"WAITFOR SIGNAL"
260     OUTPUT 709;"DISABLE PROBE"
270     OUTPUT 709;"VREAD PRV_STAT"
280     OUTPUT 709;"VREAD NEW_TASK"
290     OUTPUT 709;"VREAD PERIOD"
300   !
310     ENTER 709;P_s(*)
320     ENTER 709;N_t(*)
330     ENTER 709;P(*)
340   !
```

```
350     PRINT "STATUS     NEW        TIME OF"
360     PRINT "OF PREV    TASK       SWAP"
370     PRINT "TASK "
380     PRINT "-------------------------"
390   !
400     FOR I=0 TO 8
410       PRINT P_s(I),N_t(I),P(I)
420     NEXT I
430     PRINT
440   NEXT J
450   END
```

A typical output based on this program is given below:

```
STATUS            NEW            TIME OF
OF PREV           TASK           SWAP
TASK
--------------------------------------
35                 0               .09
7                 -5               .093
1                  0               .193
7                 -5               .195
1                  0               .296
7                 -5               .298
1                  0               .398
3                 -3               .399
0                  0               0


STATUS            NEW            TIME OF
OF PREV           TASK           SWAP
TASK
--------------------------------------
35                 0               .077
7                 -5               .079
1                  0               .231
7                 -5               .232
1                  0               .427
7                 -5               .429
1                  0               .624
3                 -3               .625
0                  0               0
```

The first number in the "status" column is 35. From the code information listed for the *prev_stat* parameter, 35 (= 32+2+1) indicates that the HPIB task has been suspended by the WAITFOR SIGNAL command (32) in line 250, and that the HPIB task has also been created (1) and is running (2). The 0 in the "new task" column means that run task 0 is the next task that is swapped to. When the time-slice period expires, run task 0 has been created (1), is running (2), but has also been suspended (4). This is indicated by the 7 in the status column. Since the HPIB task is waiting to be signaled and there are no other run tasks, the system goes into a "ready" state rather than to another task as indicated by -5 in the new task column.

Notice that the status of a task does not correspond to the new task shown on the same line. Rather, the status applies to the task shown on the previous line.

Continuing through the output, the status of the ready condition is reported (1) and the system swaps back to run task 0 which then suspends. This sequence is repeated until run task 0 finishes and signals the HPIB task (line 190). At this point, the probe is disabled (line 260) and the data in the probe arrays is entered into the controller. The -3 in the new task column indicates that command execution did occur within the HPIB task (i.e. the DISABLE PROBE command). The zeroes at the end of the output indicate the probe arrays were of sufficient size since no data was overwritten and a complete trace of the program was obtained.

The "TIME OF SWAP" column shows the times at which a swap occurred. Looking at the first output, the time from when the clock was set in line 90 (HPIB task) until the swap to run task 0 was 90 ms (.09). Run task 0 was then suspended and the system swapped to the ready state 3 ms later (.093). Once the suspend period (100 ms) expired, the system swapped back to run task 0 (.193). Run task 0 suspends again and a swap is made to the ready state 2 ms later (.195). (Note that the resolution of the internal clock is 1 ms.)

In the second output, the time from when the clock is set until the swap to run task 0 is 77 ms (.077). Run task 0 is then suspended and the system swaps to the ready state 2 ms later (.079). When the suspend period expires, the system swaps back to run task 0 (.231).

Note that the difference between .231 and .079 is .152. The SUSPEND command in line 170, however, suspends the task for 100 ms (.1). The reason the task is suspended a longer period of time is because of the tic interval resolution (65 ms) on the second pass through the FOR J/NEXT J loop. Since the suspend period expired between tic intervals, the system had to wait for the next tic interval before a swap could occur. Similarly, the suspend period expired between tic intervals on the first pass through the loop. However, given the tic interval resolution was 1.024 ms, the tic intervals are much shorter, allowing a swap to occur closer to the suspend period specified.

• **HP 44714A 3-Channel Stepper Motor Controller/Pulse Output**

Description

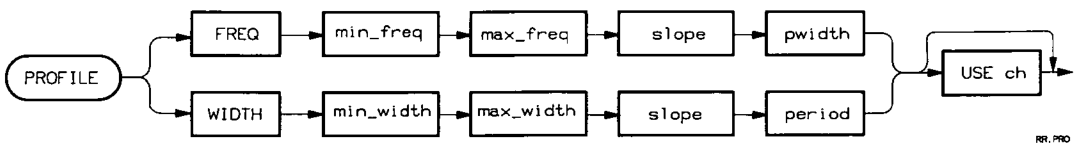
Defines a trapezoidal motion profile that defines the frequency range, acceleration/deceleration rate, and pulse width or pulse width range and period of pulses generated by the MOVE and SUSTAIN commands.

Prerequisites

Requires mainframe firmware revision 3.0 or greater.

Syntax

PROFILE *FREQ min_freq max_freq slope pwidth* [USE *ch*] **or**
WIDTH min_width max_width slope period [USE *ch*]



Parameters

FREQ or *WIDTH* — The *FREQ* or *WIDTH* mode parameter specifies whether the profile is a frequency profile or a pulse width profile. Frequency (FREQ) is the choice for controlling stepper motors while pulse width (WIDTH) is the choice for general purpose pulse output.

mode	Definition
FREQ	FREQ is the mode used for controlling stepper motors and determines the motor's velocity.
WIDTH	WIDTH is the mode used for general purpose pulse output and determines the pulse width and period of the pulses.

min_xxxxx — FREQ mode: *min_freq* specifies the minimum frequency/velocity (Hz).
WIDTH mode: *min_width* specifies the minimum pulse width (Seconds).

max_xxxxx — FREQ mode: *min_freq* specifies the maximum frequency (Hz).
WIDTH mode: *min_width* specifies the maximum pulse width (Seconds).

slope — FREQ mode: specifies the rate of change from one frequency/velocity to another (acceleration/deceleration; Hz/second).
WIDTH mode: specifies the rate of change from one pulse width to another (seconds/second)

pwidth or *period* — FREQ mode: *pwidth* specifies the pulse width for all frequencies (must not be larger than the smallest period; specified in Seconds).
WIDTH mode: *period* specifies the period of pulse output (Seconds).

USE *ch* — Channel for which the motion profile is specified. Channel range can be ES00 to ES02.

Remarks

Power-On State

The power-on state for the PROFILE command is

```
mode = FREQ
min = 0 Hz
max = 250 Hz
slope = 500 Hz/sec
pulse_width = 50 microseconds
```

Data Returned

None

Related Commands

The PROFILE command defines how pulses are transmitted by the MOVE and SUSTAIN commands.

Examples

Example: Set a motion profile for a stepping motor with the following parameters:

set channel 0 slot 2
mode = frequency
minimum frequency = 100 Hz
maximum frequency = 1200 Hz
acceleration/deceleration (slope) = 500 Hz/sec
pulse width = 0.1 mSec

```
OUTPUT 709;"PROFILE FREQ 100, 1200, 500, 1E-4, USE 200"
```

Example: Set a pulse width profile for General Purpose Pulse Output with the following parameters:

set channel 0 slot 2
mode = width
minimum width = 15 microseconds
maximum width = 150 microseconds
acceleration/deceleration (slope) = 50 microseconds/second
period = 0.2 mSec

```
OUTPUT 709;"PROFILE WIDTH, 15E-6, 150E-6, 50E-6, 0.2E-3, USE 200"
```

• **HP 44714A 3-Channel Stepper Motor Controller/Pulse Output**

Description

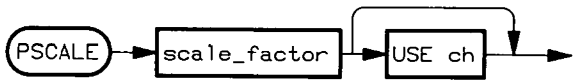
Specifies the distance of motion that one step of the motor causes. The scale factor can be in any units i.e. inches, angle of rotation, etc. Using this scale factor allows you to specify moves in distance instead of number of pulses.

Prerequisites

Requires mainframe firmware revision 3.0 or greater.

Syntax

PSCALE *scale_factor* [USE *ch*]



Parameters

scale_factor The *scale_factor* parameter is a positive or negative number that specifies the distance moved when the motor makes one step. Any unit of measure can be associated with the scale factor but all move commands must be referenced in these units. The scale factor is stored by the local processor and converts move commands specified in distance to the appropriate number of pulses.

USE *ch* Channel that the scale factor applies to. Channel range can be ES00 to ES02.

Remarks

Power-On State

The power-on state for the scale_factor is 1.0.

Data Returned

None

Beware of Round-Off Errors!

Using a scale_factor without enough resolution for individual positions can cause counter errors. If the moves are made with a *REL* parameter, the errors can grossly accumulate. For example, the example that follows show how errors can accumulate when a move is made RELative and when it is made ABSolute.

```
10 OUTPUT 709;"USE 201"                          !Use ch 1 slot 2
20 OUTPUT 709;"STANDBY AUTO LO"                  !
30 OUTPUT 709;"PULSE SD HI HI"                   !
40 OUTPUT 709;"PROFILE FREQ 100 800 500 25E-6"   !
50 OUTPUT 709;"PSCALE 2"                         !Scale factor = 2 deg
60 OUTPUT 709;"POS 0"                            !Set the counter to 0
70 OUTPUT 709:"MOVE 1 REL"                       !1 degree move relative
80 OUTPUT 709;"POS?"                             !Returns 2
```

```
 90 PAUSE                                  !Pause to check display
100 OUTPUT 709;"MOVE 1 REL"                !1 degree move relative
110 OUTPUT 709;"POS?"                      !Returns 4
120 PAUSE                                  !Pause to check display
130 OUTPUT 709;"MOVE 1 REL"                !1 degree move relative
140 OUTPUT 709;"POS?"                      !Returns 6
150 PAUSE                                  !Pause to check display
160 OUTPUT 709;"POS 0"                     !Set counter to 0
170 OUTPUT 709;"MOVE 1 ABS"                !1 degree move absolute
180 OUTPUT 709;"POS?"                      !Returns 2
190 PAUSE                                  !Pause to check display
200 OUTPUT 709;"MOVE 1 ABS"                !1 degree move absolute
210 OUTPUT 709;"POS?"                      !Returns 2
220 PAUSE                                  !Pause to check display
230 OUTPUT 709;"MOVE 1 ABS"                !1 degree move absolute
240 OUTPUT 709;"POS?"                      !Returns 4
250 END
```

Related Commands

The *scale_factor* affects the meaning of the distance and position parameters in the SOFTLIM, MOVE, SUSTAIN, POS, and POS? commands. It also affects the meaning of the velocity and acceleration/deceleration specified in the PROFILE command.

Examples

Example: Set a Scale Factor of 0.5 Inches Per Step

```
OUTPUT 709;"PSCALE 0.5 USE 201"   !Set ch 1 in slot 2
                                  !Scale factor = 0.5
```

- **Mainframe**

Description

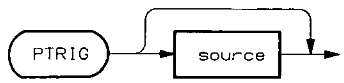
Sets the trigger source which initiates the output of pacer pulses. Any ongoing (continuous) pacer pulse output is halted.

Prerequisites

None

Syntax

PTRIG [*source*]



RR_84

Parameters

source Trigger source which starts one group of pacer output pulses. Power-on *source* = HOLD. Default *source* = SGL.

EXT	Pacer trigger source is PACER TRIGGER IN BNC connector.
SGL	Single trigger is issued at the time of command execution.
HOLD	Holds pacer triggering off (power-on configuration). Disables the PACER TRIGGER IN BNC connector (external triggering) and stops any on-going pacer pulses.

Remarks

Must Reconfigure PTRIG After Trigger

Once triggered in EXT or SGL, *source* automatically resets to HOLD and PTRIG EXT or PTRIG SGL must be set for each new trigger.

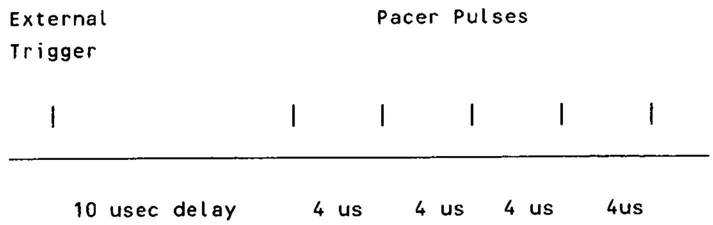
Related Commands

PACER, PDELAY

Examples

Example: Triggering Pacer Output Pulses

This program segment sets the HP 3852A so that five pacer pulses are output from the PACER OUT BNC at 4 μsec intervals. There is a 10 μsec delay from the time of an external trigger (into the PACER TRIGGER IN BNC) to the first pacer pulse output.




```
      .
      .
50 OUTPUT 709;"PACER 4.0E-6,5"    !Output 5 pulses 4 usec apart
60 OUTPUT 709;"PDELAY 10.0E-6"    !Delay first pulse 10 usec after trigger
70 OUTPUT 709;"PTRIG EXT"         !Set HP 3852A to trig on ext trigger
      .
      .
```

• **HP 44714A 3-Channel Stepper Motor Controller/Pulse Output**

Description

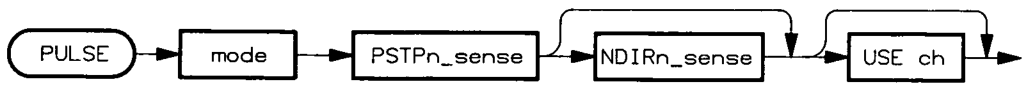
Sets which outputs the pulses appear at and whether the pulses are positive going or negative going.

Prerequisites

Requires mainframe firmware revision 3.0 or greater.

Syntax

PULSE *mode PSTPn_sense* [*NDIRn_sense*] [USE *ch*]



Parameters

mode The *mode* parameter establishes whether pulses are transmitted to the PSTPn or NDIRn outputs.

mode	Definition
SD	**Step/Direction.** Pulses appear on the PSTPn output while the NDIRn output controls direction.
SS (power-on)	**Step/Step.** Pulses for motion in one direction appear on the PSTPn output while pulses for motion in the opposite direction appear on the NDIRn output.

PSTPn_sense The *PSTPn_sense* parameter specifies whether the pulses at the PSTPn output are positive going (low-high-low) or negative going (high-low-high).

PSTPn sense	Definition
HI	In the idle state, the PSTPn output is low. When a pulse is generated, the pulse is low-high-low (the leading edge is low-to-high).
LO (power-on)	In the idle state, the PSTPn output is high. When a pulse is generated, the pulse is high-low-high (the leading edge is high-to-low).
LH	Same as HI parameter. Pulses have a low-to-high leading edge.
HL (power-on)	Same as LO parameter. Pulses have a high-to-low leading edge.

NDIRn_sense (defaults to PSTPn_sense)

The NDIRn_sense has a different meaning in the SS (Step/Step) mode than it does in the SD (Step/Direction) mode.

- In the SS (Step/Step) mode, the *NDIRn_sense* parameter specifies whether the pulses at the NDIRn output are positive going (low-high-low) or negative going (high-low-high). If the *NDIRn_sense* parameter is not specified, the sense is set to that specified for PSTPn. Definitions for the *NDIR_sense* are the same as for the *PSTPn_sense*.

- In the SD (Step/Direction) mode, the *NDIR_sense* parameter specifies direction of motor motion. When the sense is set to HI or LH, a high on the NDIRn output indicates a positive direction. When the sense is set to LO or HL, a low on the NDIRn output indicates a positive direction. If the *NDIRn_sense* parameter is not specified, the sense is set to that specified for PSTPn. Definitions for the *NDIRn_sense* are the same as for the *PSTPn_sense*.

USE *ch* — Channel for which the pulse outputs are set up. Channel range can be ES00 to ES02.

Remarks

Power-On State

The power-on state for the PULSE command is SS, LO, LO. The PSTPn and NDIRn outputs power-on in the high state (pulses are to be negative-going) and in the SS (Step/Step) mode.

Data Returned

None

Related Commands

None

Examples

Example: Set Channel 1 to Step/Step Mode; PSTPn and NDIRn senses HIGH

```
10 OUTPUT 709;"USE 201"                          !Use channel 1 in slot 2
20 OUTPUT 709;"STANDBY AUTO LO"                  !Low to power down motor
30 OUTPUT 709;"PULSE SS HI HI"                   !Step/Step mode
40 OUTPUT 709;"PROFILE FREQ 100 800 500 25E-6"   !Sets a motion profile
50 OUTPUT 709;"PSCALE 2"                         !Scale factor set to 2
60 OUTPUT 709;"MOVE 10000"                  !Move distance of 10000 units
70                                          !scaled by PSCALE (5000 pulses)
80 END
```

Example: Set Channel 1 to Step/Direction Mode; Motor drive pulses: positive-going; Direction sense: low state = positive motion

```
10 OUTPUT 709;"USE 201"                          !Use channel 1 in slot 2
20 OUTPUT 709;"STANDBY AUTO LO"                  !Low to power down motor
30 OUTPUT 709;"PULSE SD LH LO"                   !Step/Direction mode
40 OUTPUT 709;"PROFILE FREQ 100 800 500 25E-6"   !Sets a motion profile
50 OUTPUT 709;"PSCALE 2"                         !Scale factor set to 2
```

```
60 OUTPUT 709;"MOVE 10000"                !Move distance of 10000 units
70                                        !scaled by PSCALE (5000 pulses)
80 END
```

PURGE

- **HP 44788A HP-IB Controller**

Description

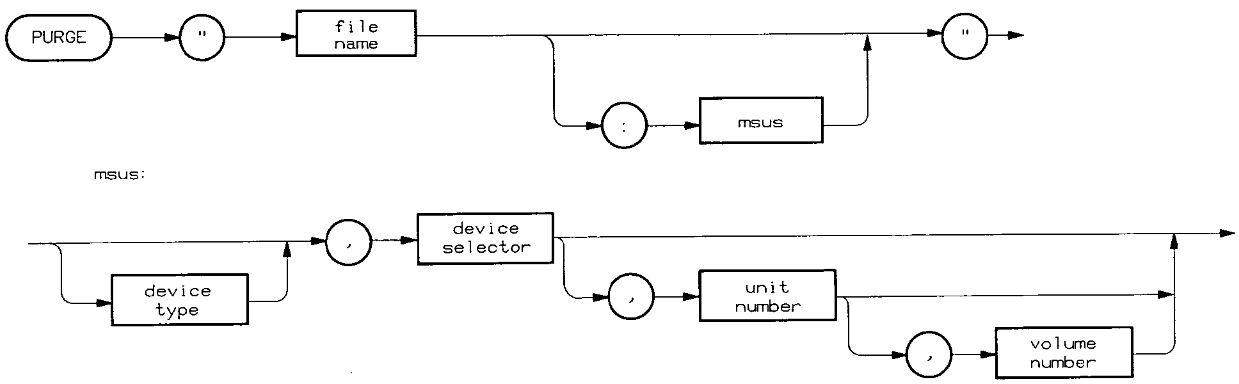
This statement deletes a file entry from the directory of the mass storage media.

Prerequisites

Requires firmware revision 3.5 or greater.

Syntax

PURGE *"file specifier"*



Parameters

file specifier The name of the file (and optional msus) to be purged.

Remarks

Purged File Cannot be Accessed

Once a file is purged, you cannot access the information which was in the file. The records of a purged file are returned to "available space". An open file must be closed before it can be purged. Any file can be closed by ASSIGN @ I/O path TO * (see ASSIGN).

Data Returned

None.

Related Commands

CREATE ASCII, CREATE BDAT

Examples

Example: Typical PURGE Statements

Two PURGE statements follow. The first statement is entered from the HP 3852A front panel, while the second statement is entered from the system controller.

```
PURGE "File1"                        !Purge file File1
100 OUTPUT 709;"PURGE 'VOLTS:,200,1'"   !Purge file VOLTS on device 200
```