Table of Contents
HP E1563A and HP E1564A Digitizer

HEWLETT-PACKARD WARRANTY STATEMENT 3
Safety Symbols 4
WARNINGS 4
Declaration of Conformity 5
WARNINGS 6
Reader Comment Sheet 7

Chapter 1
Digitizer Module Set-up 9
Using This Chapter 9
Installing the Digitizer in a Mainframe 15
Cable Connector Assembly Instructions 20

Chapter 2
Digitizer Application Information 25
Using this Chapter 25
Digitizer Block Diagrams 25
Power-on and Reset State 27
Triggering the Digitizer 29
Input Overload Condition 34
HP E1563A and E1564A Digitizer Application Examples 35

Chapter 3
Digitizer Command Reference 39
ABORt 42
CALCulate 43
CALibration 47
DIAGnostic 55
FORMat 63
INITiate 64
INPut 66
OUTPut 68
[SENSe:] 70
SAMPle 78
STATus 83
SYSTem 87
TEST 88
TRIGger 94
IEEE 488.2 Common Command Quick Reference 100

Appendix A
HP E1563A and E1564A Digitizer
Specifications 111

Appendix B
HP E1563A and E1564A
Register-Based Programming 113
About This Appendix............ 113
Register Addressing............ 113
Register Descriptions............ 116
Program Timing and Execution............ 130

Appendix C
HP E1563A and E1564A Digitizer
Error Messages 139
Execution Errors 139
Self-Test Errors............ 144
Calibration Errors 144

Appendix D
HP E1563A and E1564A
Verification Tests 145
Introduction............ 145
Verification Tests............ 146
Performance Verification Test Programs 146
Functional Test 147
Performance Verification............ 148
Performance Test Record............ 152

Appendix E
HP E1563A and E1564A
Adjustments 157
Introduction............ 157
Adjustment Procedures............ 158

HEWLETT-PACKARD WARRANTY STATEMENT

HP PRODUCT: HP E1563A 2-Channel/E1564A 4-Channel Digitizer **DURATION OF WARRANTY:** 3 years

1. HP warrants HP hardware, accessories and supplies against defects in materials and workmanship for the period specified above. If HP receives notice of such defects during the warranty period, HP will, at its option, either repair or replace products which prove to be defective. Replacement products may be either new or like-new.

2. HP warrants that HP software will not fail to execute its programming instructions, for the period specified above, due to defects in material and workmanship when properly installed and used. If HP receives notice of such defects during the warranty period, HP will replace software media which does not execute its programming instructions due to such defects.

3. HP does not warrant that the operation of HP products will be interrupted or error free. If HP is unable, within a reasonable time, to repair or replace any product to a condition as warranted, customer will be entitled to a refund of the purchase price upon prompt return of the product.

4. HP products may contain remanufactured parts equivalent to new in performance or may have been subject to incidental use.

5. The warranty period begins on the date of delivery or on the date of installation if installed by HP. If customer schedules or delays HP installation more than 30 days after delivery, warranty begins on the 31st day from delivery.

6. Warranty does not apply to defects resulting from (a) improper or inadequate maintenance or calibration, (b) software, interfacing, parts or supplies not supplied by HP, (c) unauthorized modification or misuse, (d) operation outside of the published environmental specifications for the product, or (e) improper site preparation or maintenance.

7. TO THE EXTENT ALLOWED BY LOCAL LAW, THE ABOVE WARRANTIES ARE EXCLUSIVE AND NO OTHER WARRANTY OR CONDITION, WHETHER WRITTEN OR ORAL, IS EXPRESSED OR IMPLIED AND HP SPECIFICALLY DISCLAIMS ANY IMPLIED WARRANTY OR CONDITIONS OF MERCHANTABILITY, SATISFACTORY QUALITY, AND FITNESS FOR A PARTICULAR PURPOSE.

8. HP will be liable for damage to tangible property per incident up to the greater of $300,000 or the actual amount paid for the product that is the subject of the claim, and for damages for bodily injury or death, to the extent that all such damages are determined by a court of competent jurisdiction to have been directly caused by a defective HP product.

9. TO THE EXTENT ALLOWED BY LOCAL LAW, THE REMEDIES IN THIS WARRANTY STATEMENT ARE CUSTOMER’S SOLE AND EXLUSIVE REMEDIES. EXCEPT AS INDICATED ABOVE, IN NO EVENT WILL HP OR ITS SUPPLIERS BE LIABLE FOR LOSS OF DATA OR FOR DIRECT, SPECIAL, INCIDENTAL, CONSEQUENTIAL (INCLUDING LOST PROFIT OR DATA), OR OTHER DAMAGE, WHETHER BASED IN CONTRACT, TORT, OR OTHERWISE.

FOR CONSUMER TRANSACTIONS IN AUSTRALIA AND NEW ZEALAND: THE WARRANTY TERMS CONTAINED IN THIS STATEMENT, EXCEPT TO THE EXTENT LAWFULLY PERMITTED, DO NOT EXCLUDE, RESTRICT OR MODIFY AND ARE IN ADDITION TO THE MANDATORY STATUTORY RIGHTS APPLICABLE TO THE SALE OF THIS PRODUCT TO YOU.

U.S. Government Restricted Rights

The Software and Documentation have been developed entirely at private expense. They are delivered and licensed as "commercial computer software" as defined in DFARS 252.227- 7013 (Oct 1988), DFARS 252.211-7015 (May 1991) or DFARS 252.227-7014 (Jun 1995), as a "commercial item" as defined in FAR 2.101(a), or as "Restricted computer software" as defined in FAR 52.227-19 (Jun 1987)(or any equivalent agency regulation or contract clause), whichever is applicable. You have only those rights provided for such Software and Documentation by the applicable FAR or DFARS clause or the HP standard software agreement for the product involved.

HP E1563A 2-Channel/E1564A 4-Channel Digitizer User's SCPI Programming Manual
Edition 2

HEWLETT® PACKARD

Documentation History

All Editions and Updates of this manual and their creation date are listed below. The first Edition of the manual is Edition 1. The Edition number increments by 1 whenever the manual is revised. Updates, which are issued between Editions, contain replacement pages to correct or add additional information to the current Edition of the manual. Whenever a new Edition is created, it will contain all of the Update information for the previous Edition. Each new Edition or Update also includes a revised copy of this documentation history page.

Edition 1 . . . . October 1997

Edition 2 . . . . April 1998

Safety Symbols

Instruction manual symbol affixed to product. Indicates that the user must refer to the manual for specific WARNING or CAUTION information to avoid personal injury or damage to the product.

Indicates the field wiring terminal that must be connected to earth ground before operating the equipmentÅprotects against electrical shock in case of fault.

or Frame or chassis ground terminal—typically connects to the equipment's metal frame.

Alternating current (AC)

Direct current (DC).

Indicates hazardous voltages.

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

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

WARNINGS

The following general safety precautions must be observed during all phases of operation, service, and repair of this product. 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 product. Hewlett-Packard Company assumes no liability for the customer's failure to comply with these requirements.

Ground the equipment: For Safety Class 1 equipment (equipment having a protective earth terminal), an uninterruptible safety earth ground must be provided from the mains power source to the product input wiring terminals or supplied power cable.

DO NOT operate the product in an explosive atmosphere or in the presence of flammable gases or fumes.

For continued protection against fire, replace the line fuse(s) only with fuse(s) of the same voltage and current rating and type. DO NOT use repaired fuses or short-circuited fuse holders.

Keep away from live circuits: Operating personnel must not remove equipment covers or shields. Procedures involving the removal of covers or shields are for use by service-trained personnel only. Under certain conditions, dangerous voltages may exist even with the equipment switched off. To avoid dangerous electrical shock, DO NOT perform procedures involving cover or shield removal unless you are qualified to do so.

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

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

Declaration of Conformity

according to ISO/IEC Guide 22 and EN 45014

Manufacturer's Name: Hewlett-Packard Company
Loveland Manufacturing Center

Manufacturer's Address: 815 14th street S.W.
Loveland, Colorado 80537

declares, that the product:

Product Name: 2-Channel and 4-Channel Digitizer

Model Number: HP E1563A and HP E1564A

Product Options: All

conforms to the following Product Specifications:

Safety: IEC 61010-1 (1990) Incl. Amend 2 (1996)/EN61010-1 (1993)
CSA C22.2 #1010.1 (1992)
UL 3111-1 (1994)

EMC: CISPR 11:1990/EN55011 (1991): Group 1, Class A
EN61000-3-2:1995 Class A
EN61000-3-3:1995
EN50082-1:1992
IEC 1000-4-2:1995 4kV CD, 8kV AD
IEC 1000-4-3:1995 3 V/m
IEC 1000-4-4:1995 1kV Power Line, 0.5kV Signal Lines
ENV50141:1993/prEN50082-1 (1995): 3 Vrms
EN61000-4-5:1995 1kV CM, 0.5kV DM
EN61000-4-8: 1993/prEN50082-1 (1995): 3 A/m
EN61000-4-11:1994/prEN50082-1 (1995): 30%,10ms 60%,100ms

Supplementary Information: The product herewith complies with the requirements of the Low Voltage Directive 73/23/EEC and the EMC Directive 89/336/EEC (inclusive 93/68/EEC) and carries the "CE" mark accordingly.

Tested in a typical configuration in an HP C-Size VXI mainframe.

November 15, 1997

Jim White

Jim White, QA Manager

European contact: Your local Hewlett-Packard Sales and Service Office or Hewlett-Packard GmbH, Department HQ-TRE, Herrenberger Straße 130, D-71034 Böblingen, Germany (FAX +49-7031-14-3143)

WARNINGS

The HP E1563A and E1564A Digitizers are capable of measuring voltages up to 256V maximum. Voltage levels above the levels specified for accessible connectors or cable ends could cause bodily injury or death to an operator. Special precautions must be adhered to (discussed below) when applying voltages in excess of 60 Vdc, 30 Vac rms or 42.4 Vac peak for a continuous, complex waveform.

Module connectors and test signal cables connected to them cannot be operator accessible. Cables and connectors are considered inaccessible if a tool (e.g., screwdriver, wrench, socket, etc.) or a key (equipment in a locked cabinet) is required to gain access to them. Additionally, the operator cannot have access to a conductive surface connected to any cable conductor (High, Low or Guard).

Assure the equipment under test has adequate insulation between the cable connections and any operator-accessible parts (doors, covers, panels, shields, cases, cabinets, etc.). Verify there are multiple and sufficient protective means (rated for the voltages you are applying) to assure the operator will NOT come into contact with any energized conductor even if one of the protective means fails to work as intended. For example, the inner side of a case, cabinet, door, cover or panel can be covered with an insulating material as well as routing the test cables to the module's front panel connectors through non-conductive, flexible conduit such as that used in electrical power distribution.

Please fold and tape for mailing

Reader Comment Sheet

HP E1563A 2-Channel and HP E1564A 4-Channel Digitizer User's and SCPI Programming Manual
Edition 2

You can help us improve our manuals by sharing your comments and suggestions. **In appreciation of your time, we will enter you in a quarterly drawing for a Hewlett-Packard Palmtop Personal Computer** (U.S. government employees are not eligible for the drawing).

Your Name	City, State/Province
Company Name	Country
Job Title	Zip/Postal Code
Address	Telephone Number with Area Code

Please list the system controller, operating system, programming language, and plug-in modules you are using.

fold here

cut along this line

NO POSTAGE NECESSARY IF MAILED IN THE UNITED STATES

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO. 37 LOVELAND, CO

POSTAGE WILL BE PAID BY ADDRESSEE

HEWLETT-PACKARD COMPANY
Measurement Systems Division
Learning Products Department
P.O. Box 301
Loveland, CO 80539-9984

fold here

Please pencil-in one circle for each statement below:

	Disagree				Agree
• The documentation is well organized.	O	O	O	O	O
• Instructions are easy to understand.	O	O	O	O	O
• The documentation is clearly written.	O	O	O	O	O
• Examples are clear and useful.	O	O	O	O	O
• Illustrations are clear and helpful.	O	O	O	O	O
• The documentation meets my overall expectations.	O	O	O	O	O

Please write any comments or suggestions below–be specific.

Chapter 1

Digitizer Module Set-up

Note Your Digitizer may have experienced temperature extremes during shipment that can affect it's calibration. It is recommened you perform a zero offset calibration upon receipt using the CAL:ZERO<*channel*>:ALL? command for each channel to realize the accuracy specifications in Appendix A. See Appendix E for the zero adjustment procedure.

Using This Chapter

This chapter provides one page of general module information followed by the tasks you must perform to set up your module and verify your installation was successful. Chapter contents are:

- Adding DRAM (PC Memory SIMM) to the Module
- Setting the Module Address Switch
- Interrupt Line
- Input Terminals and Front Panel Indicators
- 3-Wire and 2-Wire Input Cabling Considerations
- Initial Operation

General Information

The HP E1563A (2-channel) and HP E1564A (4-channel) are 800 kSample/second (14-bit resolution) digitizers capable of handling both continuous and transient voltages up to 256V. They are ideal for measurements in electomechanical design characterization, particularily in environments with high levels of electrical noise. They are also ideal for characterizing electronic and mechanical transient waveforms. You *cannot* upgrade an E1563A 2-Channel to an E1564A 4-Channel Digitizer.

Both digitizers are designed to use PC SIMM memory. Memory sizes that are supported are 4, 8, 16, 32, 64 and 128 Mbytes. The large memory can easily capture transients or act as FIFO to allow continuous digitizing while unloading data with block mode transfers.

All channels sample simultaneously. The sample can be from an internal clock derived from the internal time base or it can come from an external source. Triggering can be set up for several sources with programmable pre and post trigger reading counts. External time base, trigger and sample inputs are provided on the front panel "D" subminiature connector.

Both the E1563A and E1564A digitizers are register-based instruments that can be programmed at the register level (register programming information is covered in Appendix C) or at a higher level using SCPI or Plug&Play drivers.

Continuous voltages in a test set-up where the user has access to module connectors and test signal cable ends are restricted to:

- 60 Vdc
- 30 Vac-rms
- 42.4 Vac peak of a continuous, complex waveform

Continuous voltages in test set-ups where the module connectors and the test signal cables connected to them are made non-accessible are:

- 256 Vdc, 240 Vdc floating
- 256 Vac peak

Transient voltages:

- Transient voltages are permitted providing the maximum amount of charge transferred into a human body that contacts the voltage under normal conditions, does not exceed 45 uCoulombs (45 uA-s).

Overload voltages (opens channel input relay):

Range	Voltage Input Condition	Vmax
62 mV to 4V	**H**igh or **L**ow to **G**uard	>20V
16V to 256V	**L**ow to **G**uard	>40V

4-channel and 2-channel Module Differences

- The E1564A 4-Channel Digitizer has four selectable input filters per channel (1.5 kHz, 6 kHz, 25 kHz and 100 kHz) that can be enabled.
- The E1563A 2-Channel Digitizer has a fixed 25 kHz input filter per channel that can be enabled.

- The E1564A 4-Channel Digitizer has a calibration bus output (High, Low and Guard) and a programmable short.
- The E1563A 2-Channel Digitizer does not have a calibration bus output however, a programmable short is provided for each channel. An external calibration source must be provided for calibration.

The HP E1563A and E1564A Digitizers are capable of measuring voltages up to 256V maximum. Voltage levels above the levels specified for accessible connectors or cable ends could cause bodily injury or death to an operator. Special precautions must be adhered to (discussed below) when applying voltages in excess of 60 Vdc, 30 Vac rms or 42.4 Vac peak for a continuous, complex waveform.

Module connectors, and test signal cables connected to them, must be made NON-accessible to an operator who has not been told to access them: It is a supervisor's responsibility to advise an operator that dangerous voltages exist when the operator is instructed to access connectors and cables carrying these

WARNING voltages. Making cables and connectors that carry hazardous voltages inaccessible is a protective measure keeping an operator from inadvertent or unknowing contact with these harmful voltages. Cables and connectors are considered inaccessible if a tool (e.g., screwdriver, wrench, socket, etc.) or a key (equipment in a locked cabinet) is required to gain access to them. Additionally, the operator cannot have access to a conductive surface connected to any cable conductor (High, Low or Guard).

Assure the equipment under test has adequate insulation between the cable connections and any operator-accessible parts (doors, covers, panels, shields, cases, cabinets, etc.): Verify there are multiple and sufficient protective means (rated for the voltages you are applying) to assure the operator will NOT come into contact with any energized conductor even if one of the protective means fails to work as intended. For example, the inner side of a case, cabinet, door, cover or panel can be covered with an insulating material as well as routing the test cables to the module's front panel connectors through non-conductive, flexible conduit such as that used in electrical power distribution.

WARNING Tighten the faceplate mounting screws after installing the module in the mainframe to prevent electric shock in the case of equipment or field wiring failure.

Caution To prevent equipment damage, do not connect this equipment to mains or to any signal directly derived from mains. Short-term temporary overvoltages must be limited to 500V or less.

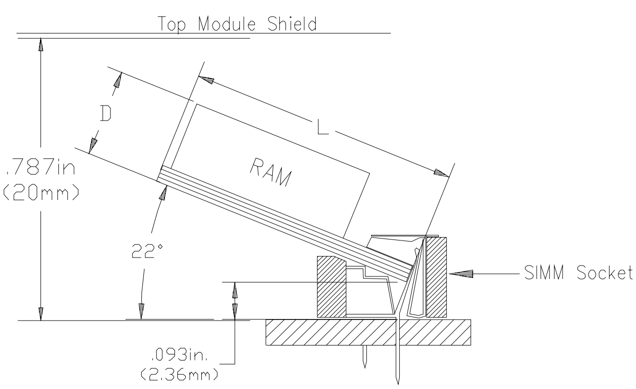
Caution To prevent equipment damage in the case of an overvoltage condition, do not connect this equipment to any voltage source which can deliver greater than 2A at 500V in the case of a fault. If such a fault condition is possible, insert a 2A fuse in the input line.

CLEANING THE FRONT PANEL AND TOP/BOTTOM SHIELDS: Clean the outside surfaces of this module with a cloth slightly dampened with water. Do not attempt to clean the interior of this module.

Adding RAM to the Module

You can increase the size of RAM on your Digitizer module by purchasing PC SIMM memory and installing it on the module after you remove the standard 4 Mbyte SIMM shipped with your digitizer. Both FPM (Fast Page Mode) and EDO (Extended Data Out) are supported.

Note Although most commercially available PC SIMM RAM will work with your Digitizer, there are some that are physically too large and will make contact with the top shield when installed. A standard 72 SIMM specifies the length (L) or keying but does not specify the depth (D). Certain depths are too large and not compatible. The E1563/E1564 has about 17.6 mm of space from the bottom of the SIMM RAM inserted in the socket to the top module shield (see diagram below). You must verify that the SIMM RAM you purchase for replacement on the module has a depth (D) that will clear the top module shield. You can use the 4 Mbyte SIMM RAM you remove as a guide, as well as the dimensions in the diagram below, when purchasing your upgrade RAM .



RAM Install Procedure

1. Disconnect any field wiring from the module and remove power from the mainframe before proceeding.
2. Remove the module from the mainframe and remove the top shield from the module.
3. Remove the 4 Mbyte SIMM from the PC board by first spreading the tabs at the ends of the SIMM connector. Store this SIMM in an anti-static bag and save this part.

Note It is important that you retain the 4 Mbyte SIMM you remove from the Digitizer. If you return your Digitizer to Hewlett-Packard for repair or exchange, you must return it in the same configuration as it was shipped to you. You must remove your large memory SIMM and replace it with the standard 4 Mbyte SIMM shipped with the product.

---

4. Add your replacement SIMM to the module's RAM socket.
5. Reinstall the module's top shield.

6. Note the new memory configuration by checking the appropriate

box on the module's top shield.

7. Set both the "CALIBRATION CONSTANTS" switch and the "FLASH" switch to the "Write Enable" position.
8. Install the module in your mainframe and apply power.
9. Set the new RAM memory size by sending the command:
   DIAGnostic:MEMory:SIZE <*size*>
10. Query the memory size to verify the setting by sending:
    DIAGnostic:MEMory:SIZE?
11. Remove mainframe power, remove the module and set the "CALIBRATION CONSTANTS" and "FLASH" switches back to the "Read Only" position.
12. Reinstall your module in your mainframe.

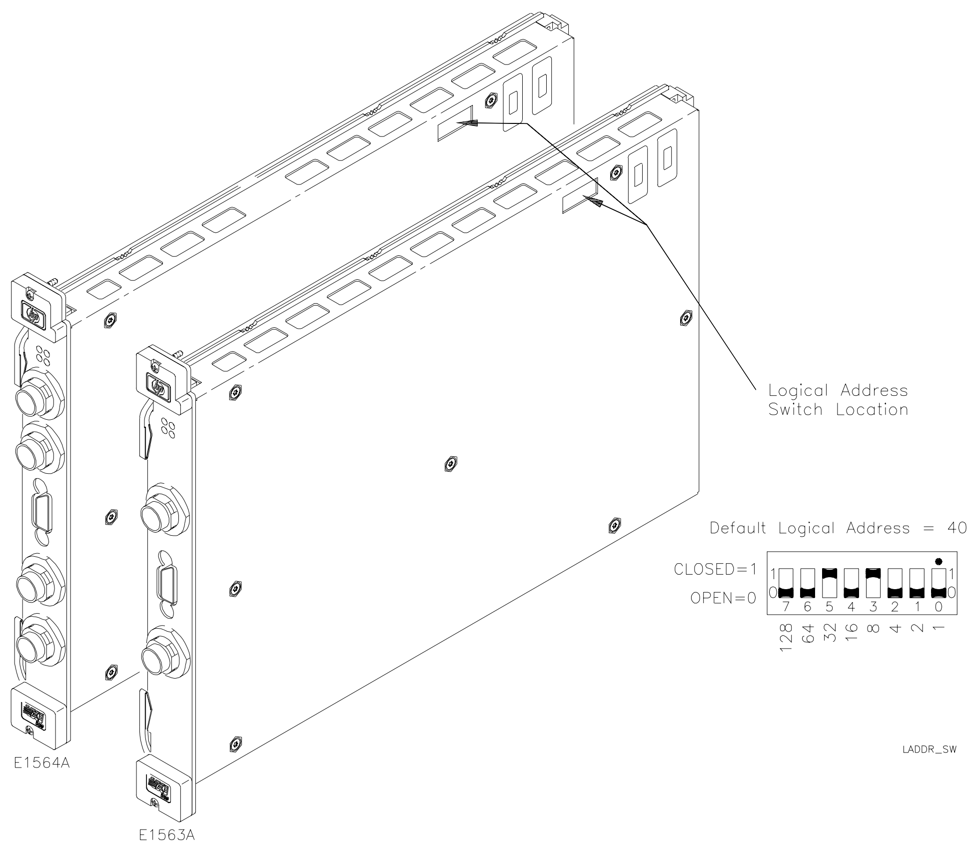
WARNING **Tighten the faceplate mounting screws to prevent electric shock in the case of equipment or field wiring failure.**

Setting the Module Address Switch

The logical address switch factory setting is 40. Valid address are from 1 to 254 for static configuration (the address you set on the switch) and address 255 for dynamic configuration. The HP E1563A and HP E1564A do not support dynamic configuration of the address.

If you install more than one digitizer, each module must have a different logical address. If you use a VXIbus command module, the logical address must be a multiple of eight (e.g., 32, 40, 48, 56, etc.). Each instrument must have a unique secondary address which is the logical address divided by eight. The Digitizer is shipped from the factory with logical address 40.

Note When using an HP E1405A/B or E1406A as the VXIbus resource manager with SCPI commands, the digitizer's address switch value must be a multiple of 8.



Interrupt Line

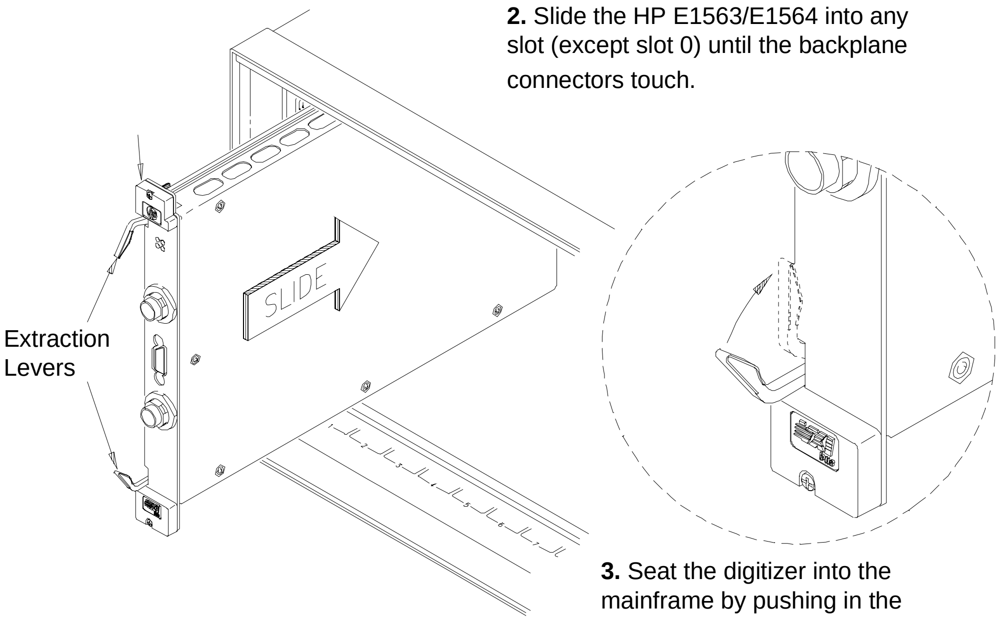
The HP E1563A and E1564A Digitizers are VXIbus interrupters. You can specify which interrupt line (1 through 7) the interrupt is transmitted. The interrupt line is specified using the DIAGnostic:INTerrupt:LINE command. You can query the active interrupt line using the DIAGnostic:INTerrupt :LINE? command. The default is no interrupt line enabled at power-up. You specify "0" if you do not want an interrupt. Resetting the module does change the interrupt line setting and you must reset your interrupt setting.

Installing the Digitizer in a Mainframe

The HP E1563A or E1564A can be installed in any slot (except slot 0) in a C-size VXIbus mainframe. Refer to following diagram for the procedure to install the Digitizer in a mainframe.

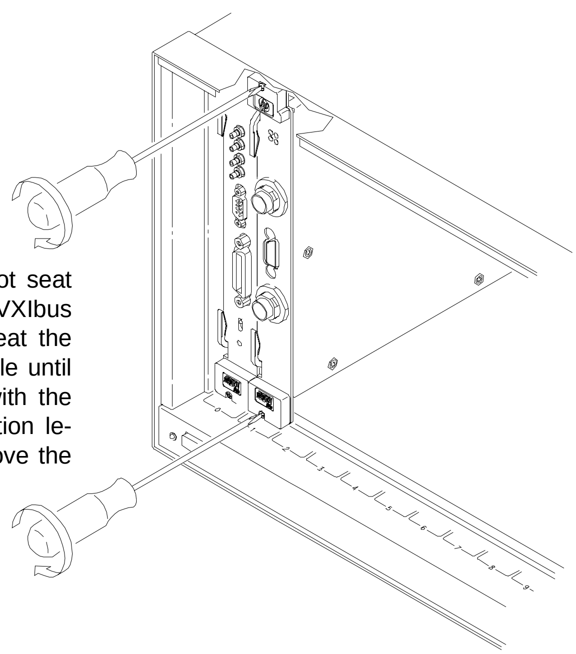
1. Set the extraction levers out.

2. Slide the HP E1563/E1564 into any slot (except slot 0) until the backplane connectors touch.



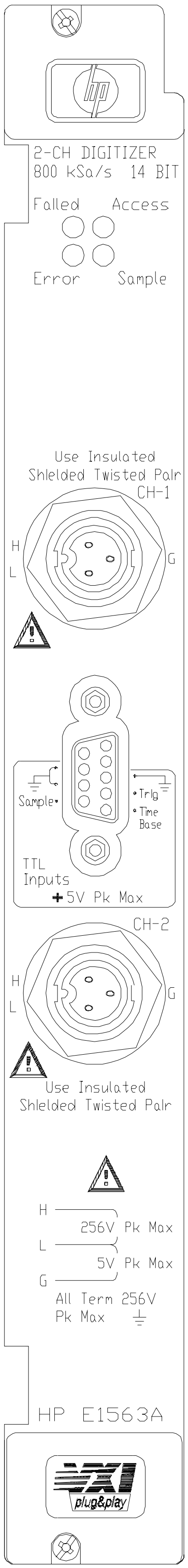
3. Seat the digitizer into the mainframe by pushing in the extraction levers.

4. Tighten the top and bottom screws to secure the digitizer module to the mainframe.



NOTE: The extraction levers will not seat the backplane connectors on older VXIbus mainframes. You must manually seat the connectors by pushing in the module until the module's front panel is flush with the front of the mainframe. The extraction levers may be used to guide or remove the digitizer.

Reverse the procedure to remove the digitizer from the mainframe.



HP E1563A Front Panel Indicators

The “Failed” LED illuminates momentarily during the digitizer’s power-on boot.

The “Access” LED illuminates only when the backplane is communicating with the digitizer.

The “Errors” LED illuminates only when an error is present in the digitizer’s driver error queue. The error can result from improperly executing a command or the digitizer being unable to pass self-test or calibration. Use the SYST:ERR? command repeatedly to read and clear the error queue (or use *CLS to clear the error queue without reading errors). A response of +0,”No error” indicates the error queue is empty. See Appendix B, HP E1563A and E1564A Digitizer Error Messages, for a list of all errors.

The “Sample” LED illuminates while the digitizer samples the input for a measurement. The “Sample” indicator typically will blink for slow sample rates and is on steady-state for high sample rates.

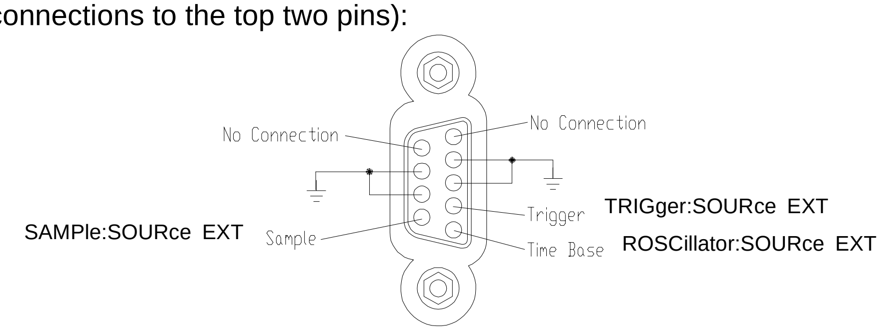
HP E1563A Input Terminals

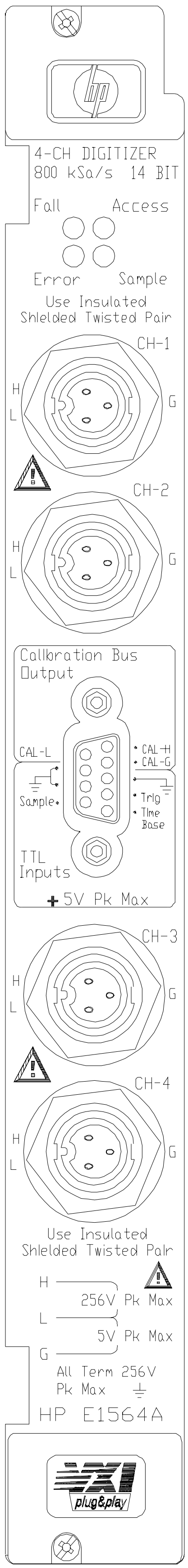
The HP E1563A Digitizer’s front panel contains two Switchcraft® EN3™ Mini Weathertight Connectors (female). Mating Switchcraft® Cord Connectors (male) are supplied with the module and the user must provide the cable and assemble the connector to the cable end. Shielded twisted pair cable is recommended with some recommendations shown in the following table that have an outside dimension compatible with the cord connector.

Wire gauge	Belden® cable P/N	Alpha® cable P/N
20 AWG (7x28)	8762	none
22 AWG (7x30)	9462	5481C
24 AWG (7x32)	8641	5491C

HP E1563A D-subminiature Connector Pins

The front panel contains a 9-pin D-subminiature connector with the following pin-out and associated SCPI commands (do not make any connections to the top two pins):





HP E1564A Front Panel Indicators

The “Failed” LED illuminates momentarily during the digitizer’s power-on boot.

The “Access” LED illuminates only when the backplane is communicating with the digitizer.

The “Errors” LED illuminates only when an error is present in the digitizer’s driver error queue. The error can result from improperly executing a command or the digitizer being unable to pass self-test or calibration. Use the SYST:ERR? command repeatedly to read and clear the error queue (or use *CLS to clear the error queue without reading errors). A response of +0,”No error” indicates the error queue is empty. See Appendix B, HP E1563A and E1564A Digitizer Error Messages, for a list of all errors.

The “Sample” LED illuminates while the digitizer samples the input for a measurement. The “Sample” indicator typically will blink for slow sample rates and is on steady-state for high sample rates.

HP E1564A Input Terminals

The HP E1564A Digitizer’s front panel contains four Switchcraft® EN3™ Mini Weathertight Connectors (female). Mating Switchcraft® Cord Connectors (male) are supplied with the module and the user must provide the cable and assemble the connector to the cable end. Shielded twisted pair cable is recommended with some recommendations shown in the following table that have an outside dimension compatible with the cord connector.

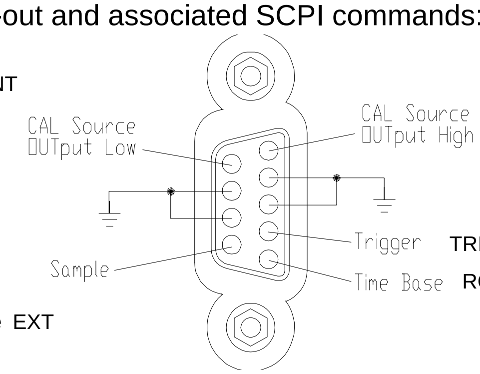
Wire gauge	Belden® cable P/N	Alpha® cable P/N
20 AWG (7x28)	8762	none
22 AWG (7x30)	9462	5481C
24 AWG (7x32)	8641	5491C

HP E1564A D-subminiature Connector Pins

The front panel contains a 9-pin D-subminiature connector with the following pin-out and associated SCPI commands:

CAL:SOURce INT

CAL:SOURce INT



TRIGger:SOURce EXT

ROSCillator:SOURce EXT

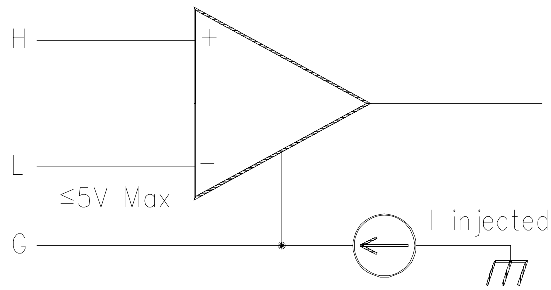
SAMPle:SOURce EXT

3-Wire and 2-Wire Input Cabling Considerations

The HP E1563A and E1564A Digitizers provide a three-terminal input system (High, Low and Guard). An unavoidable and undesirable current is injected from chassis ground to the Guard terminal. Dependent on whether you measure on a low-voltage range or a high-voltage range, the way you connect the Guard terminal may or may not introduce a measurement error due to this current. This section describes some considerations you must take to use the Guard terminal properly to minimize measurement error.

Digitizer Input Model

The input model for the digitizer is shown below. Maximum voltage between Low and Guard is 5V.

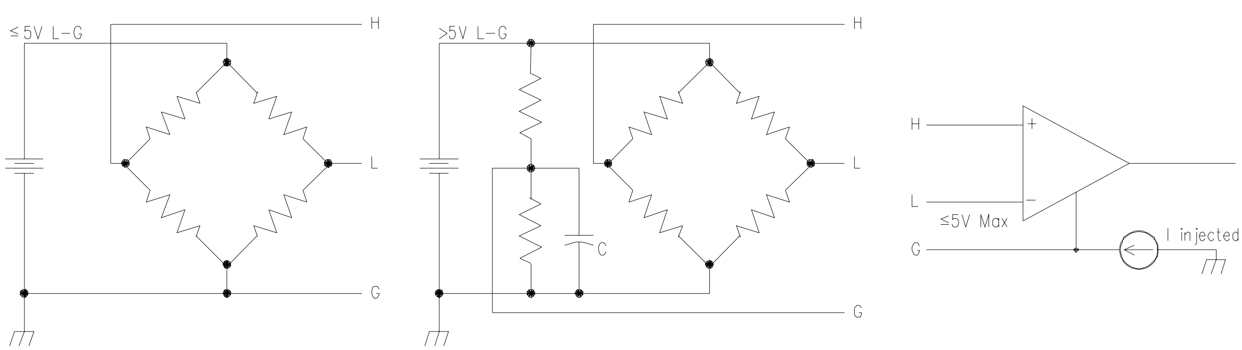


Note Maximum voltage between Low and Guard is 5V. Exceeding this limitation will not damage your digitizer but will generate invalid data for any measurement taken.

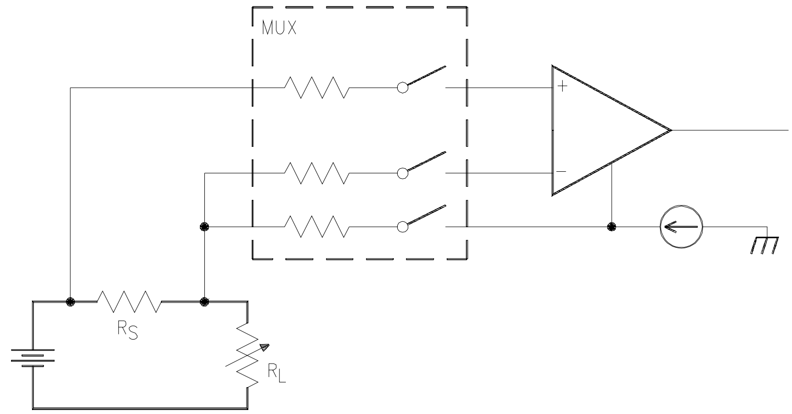
Three-Wire Connection

This section shows two examples of connecting the input using a three-wire connection. Both examples can be constructed using shielded twisted pair. The first example shows making connections for a bridge measurement where the L-to-G voltage is ≤ 5V and where the L-to-G voltage exceeds 5V. A “Wagner ground” is used to satisfy the L-to-G restriction of ≤ 5V and to make a Guard connection point that minimizes measurement error due to the digitizer’s injected current. A capacitor is added to the Wagner ground to provide a signal path to ground to minimize common mode voltages. The second example shows measuring the voltage across a small current sensing resistor where the input to the digitizer is switched through a multiplexer switch module.

3-Wire Cable Connection Example 1



3-Wire Cable Connection Example 2

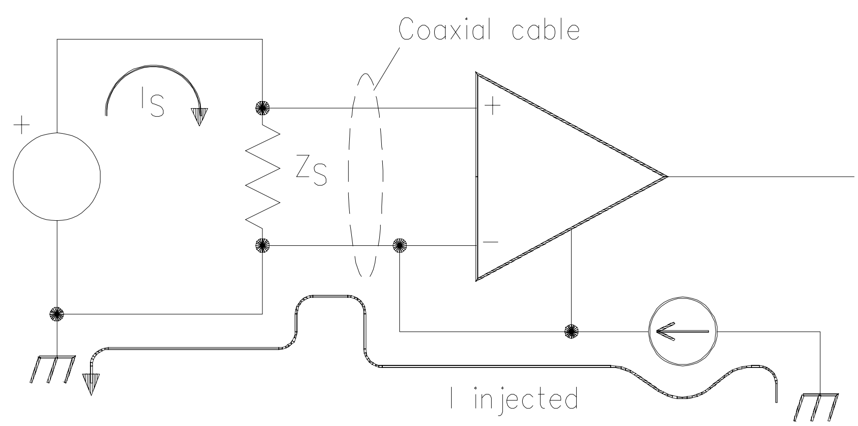


Two-Wire Connection

When Low and Guard are simply connected together at the digitizer's input on a low-voltage range (4V and below), the injected current is directed to flow through the source impedance (in a floating source) and the resultant voltage drop will introduce a measurement error. The resultant voltage drop through the source impedance can be a significant error on low-voltage ranges where the voltage of interest is small. It is not as significant an error on high-voltage ranges simply because the error introduced is not a significant part of a larger voltage; the percent of error is less significant.

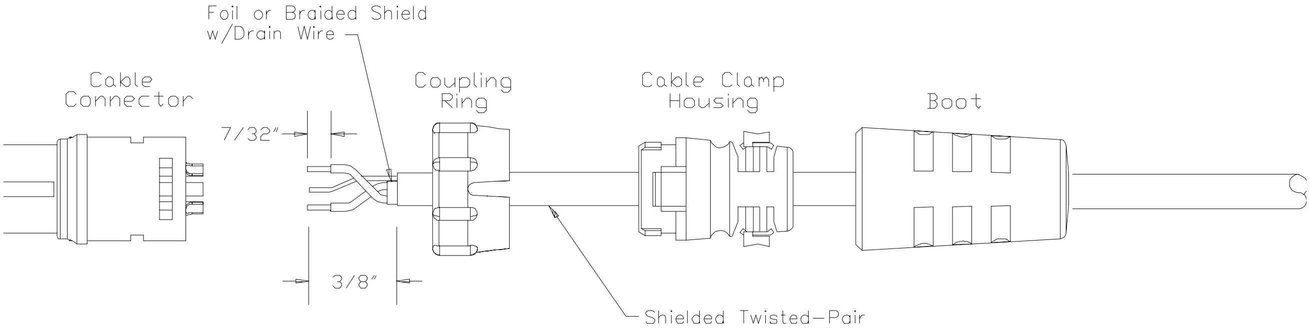
Measurement error can increase significantly when you connect Low to Guard at the digitizer's input AND use switches to switch input signals to the digitizer. Some switches have input protection resistors (usually 100Ω) in series with the switch. The digitizer's injected current now generates a voltage drop across this resistor in addition to the voltage drop generated across the source impedance. Even with a grounded source, an error voltage is generated across the switches current limiting resistor.

2-Wire Cable Connection Example

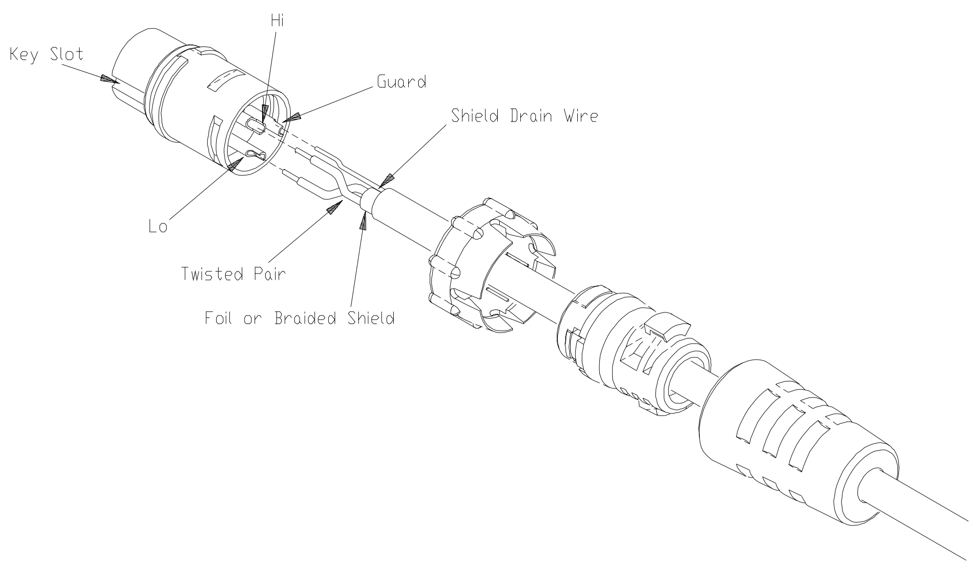


Cable Connector Assembly Instructions

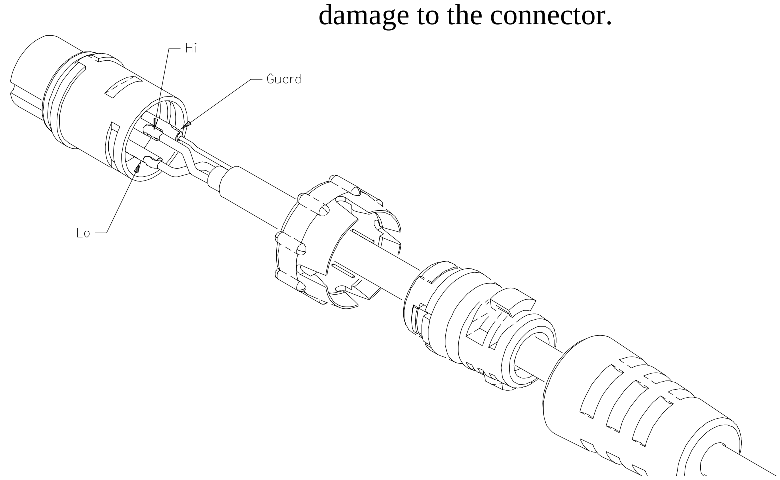
Step 1. Strip cable as shown and feed the end of the cable through the boot, cable clamp housing, and coupling ring in the order and position shown. NOTE: The coupling ring can also be inserted onto the cable connector from the front.



Step 2. Orient the HI, LO and Guard conductors with the corresponding pins.



Step 3. Solder conductors to pins. **CAUTION -** Excessive heat on the connector terminals can cause damage to the connector.

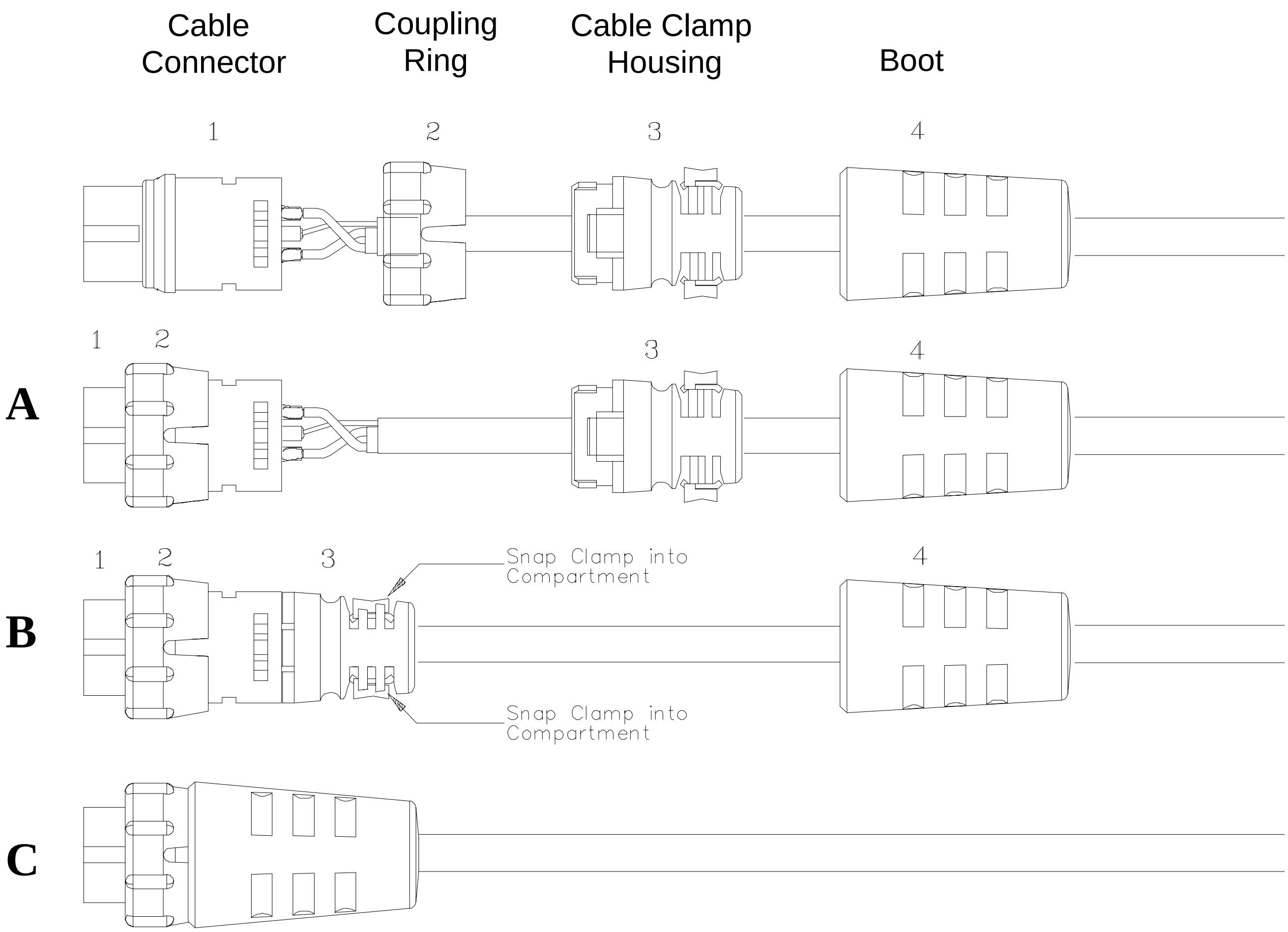


Step 4. Assemble the connector.

A. Align coupling ring's tabs with cable connector's side notches and push the coupling ring onto the cable connector.

B. Push the cable clamp housing forward until it locks into the connector body and snap the two clamps into their compartments to secure the cable.

C. Push the boot all the way forward to seat tightly onto the cable clamp housing.



To mate the cable connector to the instrument's front panel connector...

1. Hold the cable connector by the rubber boot and align the notched key slot with the key on the left side of the instrument's front panel connector. Insert the cable connector just enough to encounter insertion resistance and stay in place.

2. Grasp the coupling ring and slowly rotate it clockwise, while you gently push the connector toward the panel mount, until the notches on the coupling ring drop into the front panel connector detents.

3. Continue rotating until you feel the coupling ring ride over the locking "bump" which secures the connector to the instrument's front panel connector.

Initial Operation

To program the Digitizer using SCPI, you must select the interface address and SCPI commands to be used. General information about using SCPI commands is presented at the beginning of Chapter 3. See the HP 75000 Series C Installation and Getting Started Guide for interface addressing.

Note This discussion applies only to SCPI (Standard Commands for Programmable Instruments) programming. The program is written using VISA (Virtual Instrument Software Architecture) function calls. VISA allows you to execute on VXI *Plug&Play* system frameworks that have the VISA I/O layer installed (visa.h "include" file).

Programming the Digitizer

Example: Query the Digitizer for its ID and for system errors.

Programming the digitizer using Standard Commands for Programmable Instruments (SCPI) requires that you select the controller language (e.g., C, C++, Basic, Visual Basic, etc.), interface address and SCPI commands to be used. See the "C-Size Installation and Getting Started Guide" (or equivalent) for interfacing, addressing and controller information.

The following C program verifies communication between the controller, mainframe and digitizer. It resets the module (*RST), queries the identity of the module (*IDN?) and queries the module for system errors.

```
#include <stdio.h>
#include <visa.h>

/*** FUNCTION PROTOTYPE ***/
void err_handler (ViSession vi, ViStatus x);

void main(void)
{
    char buf[512] = {0};

    #if defined(_BORLANDC_) && !defined(_WIN32_)
    _InitEasyWin();
    #endif

    ViStatus err;
    ViSession defaultRM;
    ViSession digitizer;

/* Open resource manager and digitizer sessions */
    viOpenDefaultRM (&defaultRM);
    viOpen(defaultRM, "GPIB-VXI0::9::40",VI_NULL,VI_NULL, &digitizer);

/* Set the timeout value to 10 seconds. */
    viSetAttribute (digitizer, VI_ATTR_TMO_VALUE, 10000);

/* Reset the module. */
    err = viPrintf(digitizer, "*RST\n");
        if (err<VI_SUCCESS) err_handler (digitizer, err);
```

```
/* Query for the module's identification string. */
    err = viPrintf(digitizer, "*IDN?\n");
        if (err<VI_SUCCESS) err_handler (digitizer, err);
    err = viScanf(digitizer, "%t", buf);
        if (err<VI_SUCCESS) err_handler (digitizer, err);
    printf ("Module ID = %s\n\n", buf);

/* Check the module for system errors. */
    err = viPrintf(digitizer, "*SYST:ERR?\n");
        if (err<VI_SUCCESS) err_handler (digitizer, err);
    err = viScanf(digitizer, "%t", buf);
        if (err<VI_SUCCESS) err_handler (digitizer, err);
    printf ("System error response = %s\n\n", buf);

    viClose (digitizer);    /* close the digitizer session */

}   /* end of main */

/*** Error handling function ***/

void err_handler (ViSession digitizer, ViStatus err)
{
    char buf[1024] = {0};

    viStatusDesc (digitizer, err, buf);    /* retrieve error description */
    printf ("ERROR = %s\n", buf);
    return;
}
```

Chapter 2
Digitizer Application Information

Using this Chapter

This chapter provides digitizer application information in four parts.

- Digitizer Block Diagrams.
- Triggering the Digitizer.
- Master-Slave Operation.
- HP E1563A and E1564A Digitizer Application Examples.

Digitizer Block Diagrams

A block diagram of the HP E1564A 4-Channel Digitizer is shown in Figure 2-1. The HP E1563A 2-Channel Digitizer has the same internal structure without channels 3 and 4. Note the TRIG:LEVel<channel> signals drive the internal trigger inputs, LEVel1 drives INT1, LEVel2 drives INT2, etc.

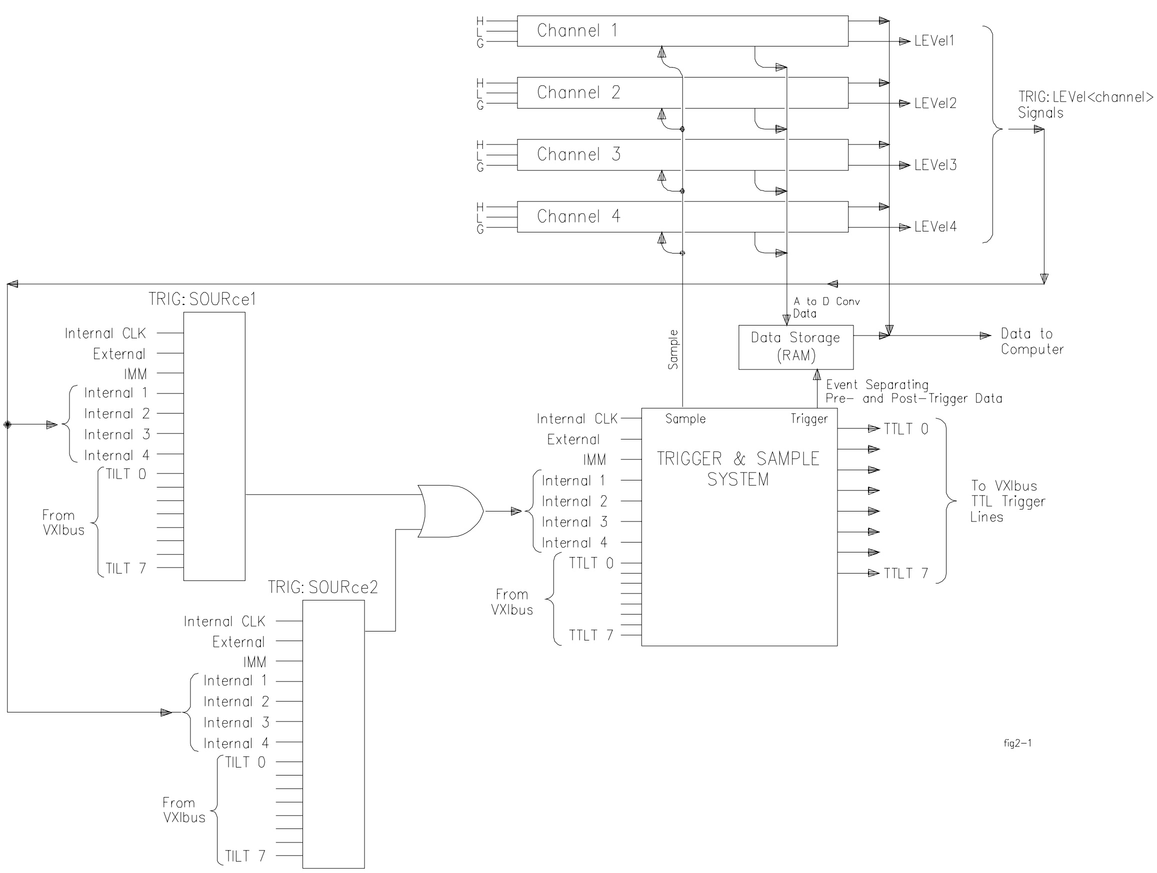
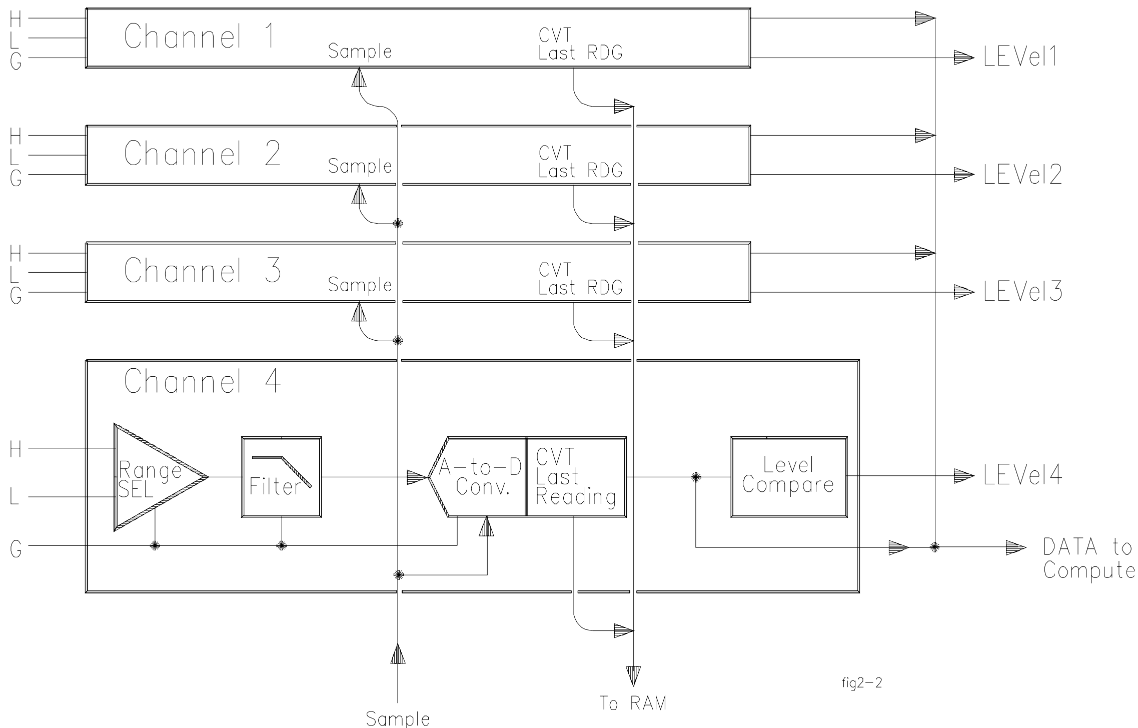


Figure 2-1. Digitizer Block Diagram.

Channel Block Diagram

Figure 2-2 is a block diagram of an individual channel and the interconnections between channels. Note that the sample signal goes to all channels. The stair-stepped commands beneath the diagram show the SCPI commands that are used to program each section of a channel. In this case, all the commands are written for channel 4. See Chapter 3, SCPI Command Reference, for a full description of the commands illustrated here.



RANGE SELECTION:
INPut4:STATe ON | 1 | OFF | 0
VOLTage4:DC:RANGe <range>

FILTER SETTING:
INPut4:FILTer:LPASs:FREQ <freq>
INPut4:FILTer:LPASs:STATe ON | 1 | OFF | 0

QUERY LAST READING (current value):
SENSe:DATA:CVTable? (@4)

LIMIT and LEVEL COMPARISON:
CALCulate4:LIMit:LOWer:DATA <value>
CALCulate4:LIMit:LOWer:STATe ON | 1 | OFF | 0
or
CALCulate4:LIMit:UPPer:DATA <value>
CALCulate4:LIMit:UPPer:STATe ON | 1 | OFF | 0
or
TRIGger:SOURce INTernal4
TRIGger:LEVel4 <voltage>
TRIGger:SLOPe4 POS | 1 | NEG | 0

Figure 2-2. Digitizer Channel Block Diagram.

Pre-Trigger/ Post-Trigger Block Diagram

Figure 2-3 illustrates relationship of pre-trigger readings and post-trigger readings with the trigger event. See Chapter 3, SCPI Command Reference, for a full description of the commands illustrated here.

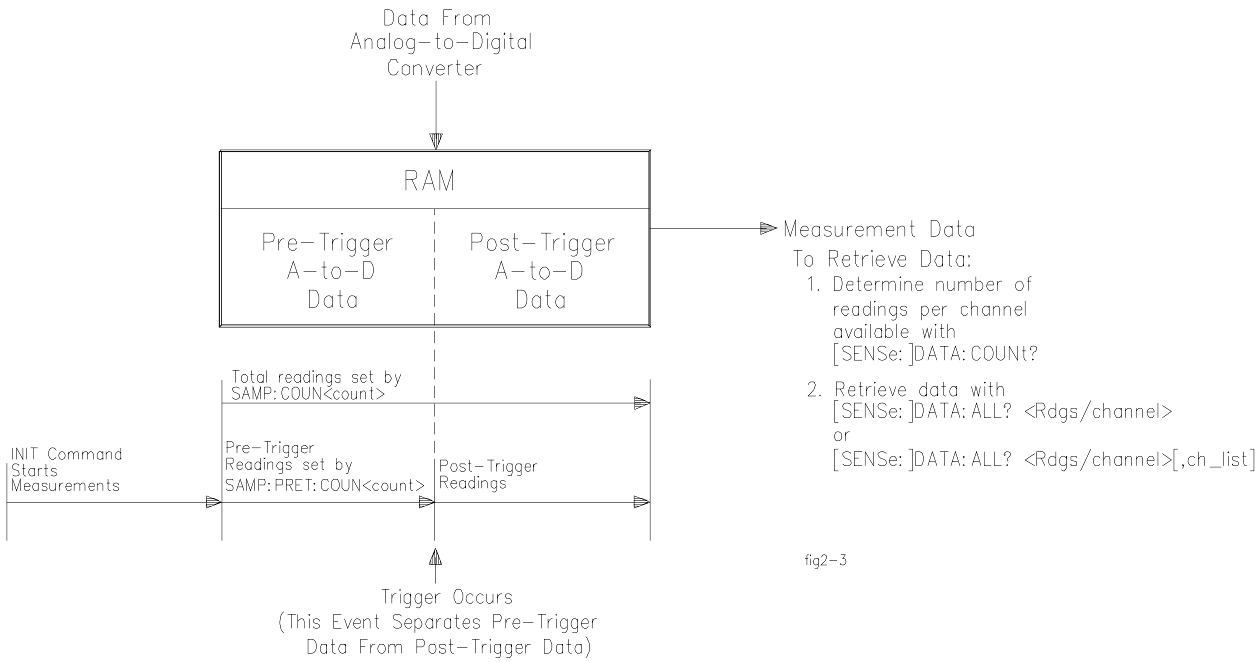


Figure 2-3. Pre-Trigger and Post-Trigger Block Diagram.

Power-on and Reset State

Table 2-1 describes all power-on and reset states for the digitizer. The reset state obtained after executing a *RST command is the same as the power-on state.

Table 2-1. Power-on and Reset States.

Parameter	Power-on/Reset State
DIAG:INTerrupt:LINE	interrupt line #1
FORMat:DATA	ASCii
INPut1:FILTer:FREQ	0 (no filter on channel 1)
INPut2:FILTer:FREQ	0 (no filter on channel 2)
INPut3:FILTer:FREQ	0 (no filter on channel 3)
INPut4:FILT:FREQ	0 (no filter on channel 4)

Table 2-1. Power-on and Reset States.

Parameter	Power-on/Reset State
INPut1:STATe	ON (channel 1 input state)
INPut2:STATe	ON (channel 2 input state)
INPut3:STATe	ON (channel 3 input state)
INPut4:STATe	ON (channel 4 input state)
OUTPut:TTLT0-7:SOURce	TRIGger (all TTLTrigger lines)
OUTPut:TTLT0-7:STATe	OFF (all TTLTrigger lines)
ROSCillator:SOURce	INTernal
SWEep:POINts	1 (one sample)
SWEep:OFFSet:POINts	0 (no pretrigger samples)
VOLT1:RANGe	256V (channel 1 range)
VOLT2:RANGe	256V (channel 2 range)
VOLT3:RANGe	256V (channel 3 range)
VOLT4:RANGe	256V (channel 4 range)
VOLT1:RESolution	7.8125 mV (channel 1 res)
VOLT2:RESolution	7.8125 mV (channel 2 res)
VOLT3:RESolution	7.8125 mV (channel 3 res)
VOLT4:RESolution	7.8125 mV (channel 4 res)
SAMPle:COUNt	1 (one sample)
SAMPle:PRETrigger:COUNt	0 (no pretrigger samples)
SAMPle:SLOPe	POSitive
SAMPle:SOURce	TIMer (internal time base)
SAMPle:TIMer	1.3 µS
TRIGger:LEVel1	-256V (channel 1 level)
TRIGger:LEVel2	-256V (channel 2 level)
TRIGger:LEVel3	-256V (channel 3 level)
TRIGger:LEVel4	-256V (channel 4 level)
TRIGger:SOURce1	IMMediate (source 1 not ch 1)
TRIGger:SOURce2	HOLD (source 2 not ch 2)
TRIGger:SLOPe1	POSitive (slope 1 not ch 1)
TRIGger:SLOPe2	POSitive (slope 2 not ch 2)

Triggering the Digitizer

Two Common Trigger Sources per Channel

Triggering digitizer readings across all input channels is accomplished with one or both of the two available trigger sources (TRIGger:SOURce1 and TRIGger:SOURce2). The trigger event can be different for each source e.g., SOURce1 can be EXT and SOURce2 can be TTLT0. You use the TRIG:SOURce<n> command to set the trigger source event options which can be OFF | BUS | EXT | HOLD | IMMEDIATE | INTernal1-4 | TTLT0-7. You must execute the TRIG:SOURce<n> command two times to set both trigger sources (TRIG:SOUR1 and TRIG:SOUR2). At power-up and after resetting the module with *RST, TRIG:SOUR1 defaults to IMM and TRIG:SOUR2 defaults to HOLD. The number of readings set by the SAMPle:COUNt command are taken after the trigger event occurs.

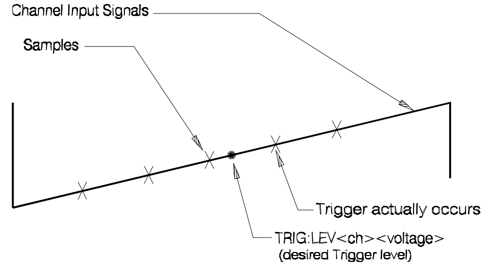
Note Do not confuse TRIG:SOUR1 as being associated with only channel 1 (as well as TRIG:SOUR2 with only channel 2). Both sources are common to ALL channels and the "1" and "2" are not channel designators but "source" designators.

Internal Triggers

Using SCPI or Plug&Play, you can trigger internally off of a voltage level from any channel. Trigger level is set using the TRIG:LEVel<channel> <*voltage*> command for the particular channel you want to generate the trigger event. You then set the trigger source to trigger internally from that channel using the TRIG:SOURce<n> INT<channel> command. For example, if you want to trigger from a 11.5V level on channel 2, you send the following commands: VOLT2:RANG 16; TRIG:LEV2 11.5; TRIG:SOUR INT2. Figure 2-1 illustrates the relationship of the trigger level to the internal trigger source.

Internal Trigger Level Detection

Each channel has a level compare circuit that compares the input signal to the value set by the TRIG:LEVel<*channel*> command. This level initiates a trigger when the input signal equals **OR EXCEEDS** the value set by the TRIG:LEVel command. This means the trigger can occur at a value other than the value set by the TRIG:LEVel command. For example, assume a trigger level of 0V on a ramp from -1V to +1V. The first samples may be negative values close to zero. These values will not cause a trigger because they do not equal or exceed the trigger level value yet. The next sample may be a positive value greater than the trigger level. The trigger compare circuit (see Figure 2-2) detects this level is equal to or greater than the trigger level value set and a trigger is generated. It was not however, generated at the exact trigger level value set by the TRIG:LEVel command.



External Trigger

You can provide an external trigger common to all channels. The external trigger connection is on the digitizer's front panel D-subminiature connector "Trig" pin. You set this input as the trigger source for all channels using the TRIGger:SOURce<n> EXT command. You use the TRIGger:SLOPe<n> POSitive | NEGative command to set which signal edge will trigger.

Master-Slave Operation

The HP E1563A and HP E1564A Digitizers can be configured in a master-slave configuration. This configuration allows a master module and one or more slave modules to have their measurements synchronized. Synchronization occurs by having all channels trigger off of the same trigger event as well as all channels sampling from one sample signal.

- The sample synchronization signal is always generated by the master.
- The TTL trigger event can be generated by either the master module or any of the slave modules. This allows a slave module (as well as the master module) to use one of the four internal trigger sources or their external trigger source to trigger a measurement.

Both the trigger signal and the sample signal are put on the VXI backplane TTL trigger (TTLT) lines where the master module and all slave modules receive the signals simultaneously. TTL trigger lines are used in pairs between the master and slave(s) where one TTL trigger line carries the sample signal and the other carries the trigger signal. The next section describes how these TTL trigger lines are paired.

Trigger Mode

The TRIGger:MODE command is used to configure Digitizers for master-slave operation. The mode can be NORMal, MASTer or SLAVe.

NORMal Mode

The default setting for trigger mode is TRIGger:MODE NORMal which configures the module as an individual instrument.

MASTer Mode

Use the TRIGger:MODE MASTer<n> command to configure a module as a master. The eight TTL trigger lines (TTLT0-TTLT7) on the VXI backplane allow four different pairings as shown in Table 2-1 (MASTer0-SLAVe0, MASTer2-SLAVe2, MASTer4-SLAVe4 and MASTer6-SLAVe6). You must select an unused set of TTL trigger lines for the master-slave coupling when determining which master mode to set. Do not use a TTLT line already used by SAMPle:SOURce or TRIGger:SOURce.

SLAVe Mode

Use TRIGger:MODE SLAVe0 to configure a module as a slave to a MASTer0 module. MASTer0 and SLAVe0 modules share TTL trigger lines TTLT0 and TTLT1. TTLT0 carries the sample signal and TTLT1 carries the trigger signal. The following table shows all pairs of TTL trigger lines for each master-slave mode.

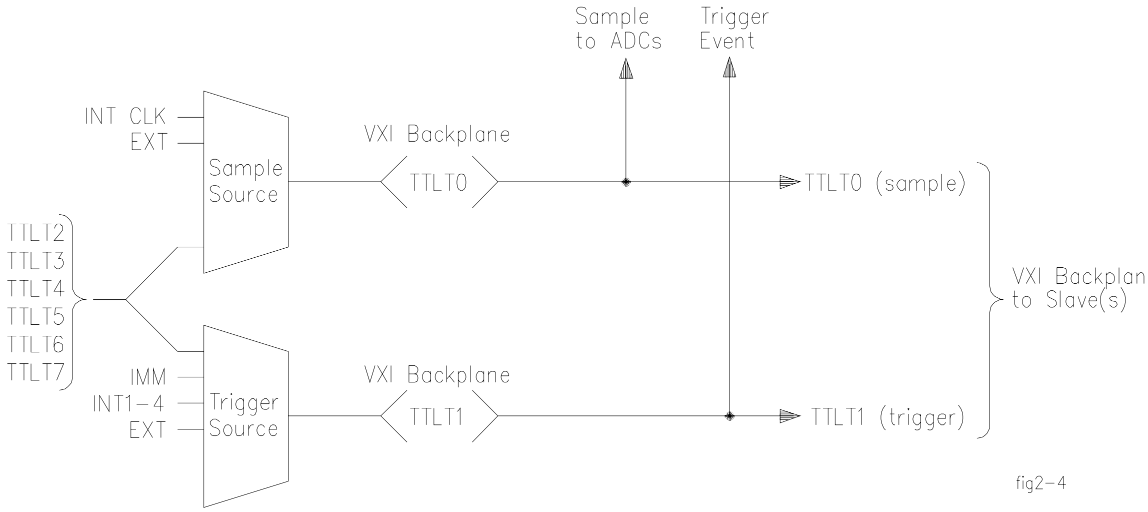
Table 2-2. Trigger Sources for Master-Slave Modes.

		MASTer-SLAVe Trigger Sources	
MASTer MODE	**SLAVe MODE**	**TRIG:SOUR1**	**TRIG:SOUR2**
MASTer0	SLAVe0	TTLT1	Any source (except TTLT0 & TTLT1)
MASTer2	SLAVe2	TTLT3	Any source (except TTLT2 & TTLT3)
MASTer4	SLAVe4	TTLT5	Any source (except TTLT4 & TTLT5)
MASTer6	SLAVe6	TTLT7	Any source (except TTLT6 & TTLT7)

Master-Slave Diagrams

Figures 2-4 illustrates a module configured as a master module. Figure 2-5 illustrates a module configured as a slave module.

TRIG:MODE MASTer0 pairs TTLT0 (sample) with TTLT1 (trigger) The MASTer0 module will function with all SLAVe0 modules.



1) The trigger source from the master can be set with TRIG:SOURce1,2 IMM | INT1-4 | EXT | TTLT<n>.

2) TRIG:MODE MASTer0 drives the TTL lines as if OUTPut:TTLT0:SOURce SAMPle and OUTPut:TTLT1:SOURce TRIGger have been set.

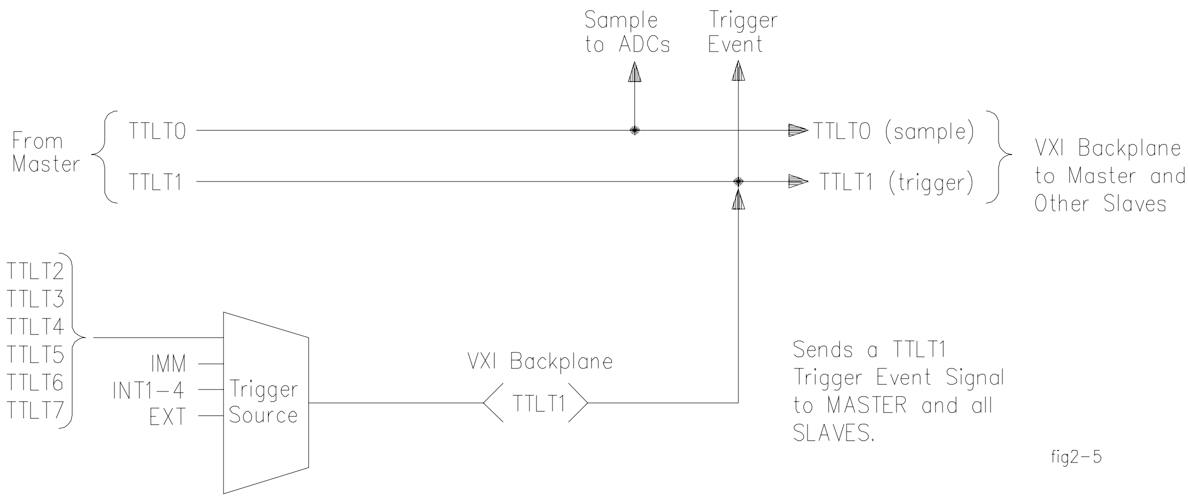
3) The master module generates the sample signal that all modules (master and slaves) initiate a measurement from.

MODE	MASTer Sample Signal
MASTer0	TTLT2-7 | INT1-4 | EXT
MASTer2	TTLT0,1,4-7 | INT1-4 | EXT
MASTer4	TTLT0-3,6-7 | INT1-4 | EXT
MASTer6	TTLT0-5 | INT1-4 | EXT

4) MASTer0 sets the TTLT1 line as if it were TRIG:SOUR1 TTLT1. However, the query TRIG:SOUR? will not return this setting. This line is simply dedicated for synchronization between the two modules in the master- slave mode. You should not use this line for any other purpose with the OUTPut, SAMPle or TRIGger commands.

Figure 2-4. Master Module Configuration Block Diagram.

TRIG:MODE SLAVe0 pairs TTLT0 (sample) with TTLT1 (trigger) A SLAVe0 module will function with other SLAVe0 modules and the MASTer0 module.



1) The trigger source from the slave can be set with TRIG:SOURce2 IMM | INT1-4 | EXT | TTLT<n>.

MODE	SLAVe Sample signal
SLAVe0	TTLT0
SLAVe2	TTLT2
SLAVe4	TTLT4
SLAVe6	TTLT6

2) SLAVe0 sets the TTLT0 line as if it were SAMP:SOUR TTLT0 and sets the TTLT1 line as if it were TRIG:SOUR1 TTLT1. However, the queries SAMP:SOUR? or TRIG:SOUR? will not return these settings. These lines are simply dedicated for synchronization between the modules in the master- slave mode. You should not use these lines for any other purpose with the OUTPut, SAMPle or TRIGger commands.

Figure 2-5. Slave Module Configuration Block Diagram.

Input Overload Condition

Overload voltages may occur which will open the channel input relay disconnecting the input signal from the channel. Overload voltage by range is shown in the following table.

Range	Voltage Input Condition	Vmax
62 mV to 4V	**H**igh or **L**ow to **G**uard	>20V
16V to 256V	**L**ow to **G**uard	>40V

Overload Reporting

The overload is reported both when the readings are retrieved and when the next measurement is initiated. If an overload occurred:

1. An error message is returned when data is retrieved informing you that the data is questionable (Overload detected - data questionable).

2. An error message is also returned when you initiate the next measurement (Overload detected - attempting re-connect of input relays).

HP E1563A and E1564A Digitizer Application Examples

This section contains example programs that demonstrate several applications of the HP E1563A or HP E1564A Digitizer. The examples described in this section list only the SCPI commands (see Chapter 3, HP E1563A and HP E1564A Command Reference) required to perform the application. The programming language is not included in print but C programs are included on the VXI*plug&play* driver media under the subdirectory "examples". You can use these examples to help you learn the capabilities of the HP E1563A/E1564A and then to help you develop programs for your specific application

C Language Example Programs

Example programs are provided on the VXI*plug&play* media. These programs have been compiled and tested using Microsoft® Visual C++™ Version 1.51 for the C programs. All of the C Language example programs in this section are written for the HP 82341A HP-IB Interface Card using the HP VISA I/O Library.

C Programs

All projects written in C programming language require the following Microsoft® Visual C++™ Version 1.51 settings to work properly:

Project Type:QuickWin application (.EXE)

Project Files:<*source code file name*>.C

[drive:]\VXIPNP\WIN\LIB\MSC\VISA.LIB (Microsoft® compiler)

[drive:]\VXIPNP\WIN\LIB\BC\VISA.LIB (Borland® compiler)

Memory Model:Options | Project | Compiler | Memory Model ⇒ Large

Directory Paths:Options | Directories

Include File Paths: [drive:]\VXIPNP\WIN\INCLUDE

Library File Paths: [drive:]\VXIPNP\WIN\LIB\MSC (Microsoft®)

[drive:]\VXIPNP\WIN\LIB\BC (Borland®)

Example programs: located on the Universal Instrument Drivers CD

Hardware Used

486 IBM compatible computer running Windows 3.1. The computer has an HP 82341 HP-IB interface and HP SICL/Windows 3.1 & Windows/NT for HP-IB software. The VXI modules were loaded in a VXI C-size mainframe using an HP E1406A Command Module as resource manager connected to the computer via the HP 82341 HP-IB card.

Compiling and Linking a C Program

You can find specific instructions for compiling C language programs for the PC in the HP VISA User’s Guide. See the section "Compiling and Linking an HP VISA Program.

Making Digitizer Measurements

This section provides three programs that demonstrate how to make digitizer measurements and retrieve data. SCPI command sequences for each program are contained in the boxes. The three programs are:

1. Use an IMMediate trigger to begin the sampling measurements on two channels and retrieve the interleaved readings from FIFO memory.

2. Use the internal level trigger to trigger off of an input ramp signal as it crosses zero. This program takes pre-trigger readings as well as post trigger readings.

3. Use an external trigger input at the D-connector “Trig” input to trigger readings.

READINGS.C

Use an IMMediate trigger to begin the sampling measurements on two channels and retrieve the interleaved readings from FIFO memory.

SCPI COMMANDS IN THIS PROGRAM:

*RST	*reset the digitizer*
*CLS	*clear the status system*
VOLT1:RANG 4	*set ch 1 to 4V range*
VOLT2:RANG 4	*set ch 2 to 4V range*
SAMP:COUN 20	*set sample count to 20 (common to all channels)*
SAMP:PRET:COUN 10	*set pre-trigger count to 10 (common to all channels)*
INIT	*initiate measurements*
DATA? 20,(@1,2)	*read 20 readings from chs 1 & 2*
Enter statement	*enter readings into the computer*

Separate the interleaved readings and display them.

Comments

- Resetting the module sets the data format to ASCII, sample source to timer and trigger source to immediate.

RAMP.C

Use the internal level trigger to trigger off of an input ramp signal as it crosses zero. This program takes pre-trigger readings as well as post trigger readings.

SCPI COMMANDS IN THIS PROGRAM:

*RST	*reset the digitizer*
*CLS	*clear the status system*
VOLT1:RANG 4	*set ch 1 to 4V range*
SAMP:COUN 7	*set sample count to 7 (common to all channels)*
SAMP:PRET:COUN 3	*set pre-trigger count to 3 (common to all channels)*
SAMP:TIM 50e-6	*set sample interval to 50 µS*
TRIG:SOUR INT1	*set trigger source to a level on channel 1*
TRIG:LEV1 0	*set the trigger level to 0V*
TRIG:SLOP POS	*set trigger slope to positive*
INIT	*initiate measurements*
DATA? 7,(@1)	*read 7 readings from ch 1*
Enter statement	*enter readings into the computer*

Display the readings.

Comments

- Resetting the module sets the data format to ASCii, sample source to TIMer and trigger source to IMMediate. The sample interval and the trigger source are changed from the reset setting.

- Resetting the module also sets the trigger level to 0V and the trigger slope to positive. Trigger level and slope commands are resent to reiterate the level and slope of the trigger. In this case, these commands are redundant.

EXT_TRIG.C

Use an external trigger input at the D-connector "Trig" input to trigger readings.

SCPI COMMANDS IN THIS PROGRAM:

*RST	*reset the digitizer*
*CLS	*clear the status system*
VOLT1:RANG 4	*set ch 1 to 4V range*
SAMP:COUN 7	*set sample count to 7 (common to all channels)*
SAMP:PRET:COUN 3	*set pre-trigger count to 3 (common to all channels)*
SAMP:TIM 100e-6	*set sample interval to 100 µS*
TRIG:SOUR EXT	*set trigger source to EXTernal (requires an external input to the "Trig" pin on the front panel D-connector)*
TRIG:LEV1 0.5	*set the trigger level to 0.5V*
TRIG:SLOP POS	*set trigger slope to positive*
INIT	*initiate measurements*
DATA? 7,(@1)	*read 7 readings from ch 1*
Enter statement	*enter readings into the computer*

Display the readings.

Comments

- Resetting the module sets the data format to ASCii, sample source to TIMer and trigger source to IMMediate. The sample interval and the trigger source are changed from the reset setting.

- Resetting the module also sets the trigger level to 0V and the trigger slope to positive. Trigger level and slope commands are resent to reiterate the level and slope of the trigger. In this case, the slope command is redundant.

Chapter 3
Digitizer Command Reference

Using This Chapter

This chapter describes the Standard Commands for Programmable Instruments (SCPI) and IEEE 488.2 Common (*) commands applicable to the HP E1563A and HP E1564A Digitizers.

Command Types

Commands are separated into two types: IEEE 488.2 Common Commands and SCPI Commands.

Common Command Format

The IEEE 488.2 standard defines the Common commands that perform functions like reset, self-test, status byte query, etc. Common commands are four or five characters in length, always begin with the asterisk character (*), and may include one or more parameters. The command keyword is separated from the first parameter by a space character. Some examples of common commands are shown below:

```
*RST    *ESR 32    *STB?
```

SCPI Command Format

The SCPI commands perform functions such as making measurements, querying instrument states, or retrieving data. The SCPI commands are grouped into command "subsystem structures". A command subsystem structure is a hierarchical structure that usually consists of a top level (or root) command, one or more low-level commands, and their parameters. The following example shows the root command CALibration and its lower-level subsystem commands:

```
CALCulate
    :LIMit:FAIL?
    :LIMit:LOWer[:STATe]  ON | 1 | OFF | 0
    :LIMit:LOWer[:STATe]?
    :LIMit:LOWer:DATA  < value>
    :LIMit:LOWer:DATA?
    :LIMit:UPPer[:STATe] ON | 1 | OFF | 0
    :LIMit:UPPer[:STATe]?
    :LIMit:UPPer:DATA <value>
    :LIMit:UPPer:DATA?
```

CALCulate is the root command, LIMit is a second level command, FAIL?, LOWer and UPPer are third level commands and DATA, DATA?, STATe and STATe? are fourth level commands.

Command Separator

A colon (**:**) always separates one command from the next lower level command as shown below:

```
CALCulate:LIMit:FAIL?
```

Colons separate the root command from the second level command

(CALCulate:LIMit) and the second level from the third level (LIMit:FAIL?).

Abbreviated Commands

The command syntax shows most commands as a mixture of upper and lower case letters. The upper case letters indicate the abbreviated spelling for the command. For shorter program lines, send the abbreviated form. For better program readability, you may send the entire command. The instrument will accept either the abbreviated form or the entire command.

For example, if the command syntax shows CALCulate, then CALC and CALCULATE are both acceptable forms. Other forms of CALCulate, such as CALCU or CALCUL will generate an error. Additionally, SCPI commands are case insensitive. Therefore, you may use upper or lower case letters and commands of the form CALCULATE, calculate, and CaLcUlAtE are all acceptable.

Implied Commands

Implied commands are those which appear in square brackets (**[ ]**) in the command syntax. (Note that the brackets are not part of the command; do not send them to the instrument.) Suppose you send a second level command but do not send the preceding implied command. In this case, the instrument assumes you intend to use the implied command and it responds as if you had sent it. Examine the partial SENSe subsystem shown below:

```
[SENSe:]
    VOLTage[:DC]:RANGe  <range>|MIN|MAX
    VOLTage[:DC]:RANGe?  [MIN|MAX]
```

The root command SENSe is an implied command; so is the third level command DC. For example, to set the digitizer's DC voltage range to MAX, you can send one of the following three command statements:

SENS:VOLT:DC:RANG MAX

VOLT:DC:RANG MAX

VOLT:RANG MAX

Parameters

Parameter Types. The following table contains explanations and examples of parameter types you might see later in this chapter.

Table 3-1.

Parameter Type	Explanations and Examples
Numeric	Accepts all commonly used decimal representations of number including optional signs, decimal points, and scientific notation. 123, 123E2, -123, -1.23E2, .123, 1.23E-2, 1.23000E-01. Special cases include MINimum, MAXimum, and DEFault.
Boolean	Represents a single binary condition that is either true or false. ON, OFF, 1, 0
Discrete	Selects from a finite number of values. These parameters use mnemonics to represent each valid setting. An example is the TRIGger:SOURce <*source*> command where source can be OFF, BUS, EXT1-2, HOLD, IMM, INT1-4 or TTLT0-7.

Optional Parameters. Parameters shown within square brackets (**[ ]**) are optional parameters. (Note that the brackets are not part of the command; do not send them to the instrument.) If you do not specify a value for an optional parameter, the instrument chooses a default value. For example, consider the TRIGger:LEVel<*chan*>? [MIN | MAX] command. If you send the command without specifying a MINimum or MAXimum parameter, the present TRIGger:LEVel value is returned for the specified channel. If you send the MIN parameter, the command returns the minimum trigger level allowable. If you send the MAX parameter, the command returns the maximum trigger level allowable. Be sure to place a space between the command and the parameter.

Linking Commands

Linking IEEE 488.2 Common Commands with SCPI Commands. Use only a semicolon between the commands. For example:

*RST;OUTP:TTLT4 ON *or* SAMP:COUNt 25;*WAI

Linking Multiple SCPI Commands From the Same Subsystem. Use only a semicolon between commands within the same subsystem. For example, to set trigger level, trigger slope and the trigger source which are all set using the TRIGger subsystem, send the following SCPI string:

TRIG:LEVel 1.5;SLOPe NEG;SOURce EXT

Linking Multiple SCPI Commands of Different Subsystems. Use both a semicolon and a colon between commands of different subsystems. For example, a SAMPle and OUTPut command can be sent in the same SCPI string linked with a semicolon and colon (;:) as follows:

SAMP:COUNt 10;:OUTP:TTLT4 ON

ABORt

This command aborts a measurement in progress or stops a measurement being made continuously. The command is ignored without error if a measurement is not in progress. This command also aborts a calibration in progress and will set the CAL:STATe to OFF.

Subsystem Syntax ABORt

Comments

- Use the DATA:COUNt? query to determine how many readings were taken before the ABORt was received.
- ABORt does not affect any instrument settings.
- Executable when initiated: YES
- Coupled command: No
- Reset (*RST) Condition: None

The CALCulate subsystem enables the limit checking of measured data.

Subsystem Syntax

```
CALCulate[<channel>]
    :LIMit:FAIL?
    :LIMit:LOWer[:STATe]   ON | 1 | OFF | 0
    :LIMit:LOWer[:STATe]?
    :LIMit:LOWer:DATA   <value> | MIN | MAX
    :LIMit:LOWer:DATA?   [MIN | MAX]
    :LIMit:UPPer[:STATe]   ON | 1 | OFF | 0
    :LIMit:UPPer[:STATe]?   [MIN | MAX]
    :LIMit:UPPer:DATA   <value> | MIN | MAX
    :LIMit:UPPer:DATA?   [MIN | MAX]
```

Comments

- Only one limit can be enabled at a time e.g., either LOWer or UPPer can be enabled but not LOWer and UPPer. If you enable the LOWer limit and later enable the UPPer limit, the LOWer limit is disabled.
- The :LIMit:FAIL? command reports the limit was exceeded. You must be aware of which limit you have enabled (LOWer or UPPer) to know which limit was exceeded.
- Lower and upper limit failures can be monitored by unmasking bits 9 and 10 in the Questionable Data Register of the status system using the STATus command.

:LIMit:FAIL?

CALCulate[<channel>]:LIMit:FAIL? queries the present status of the limit checking on the specified channel. The returned value of “0” indicates the limit was not exceeded (test passed). The returned value of “1” indicates the limit was exceeded (test failed).

Note Limit detection is reset with each new measurement, therefore, this command does not give a cumulative record of limit failures only that the last measurement either passed or failed.

:LIMit:LOWer[:STATe]

CALCulate[<channel>]:LIMit:LOWer[:STATe] OFF | 0 | ON | 1 enables the lower limit checking for the specified channel. Use :LIMit:LOWer:DATA <*value*> to set the actual limit value to be tested against. This command returns the voltage level measured and the detection mode. A returned value of “0” indicates the specified channel is disabled for lower limit checking. “1” returned indicates the specified channel is enabled and will detect signals below the specified lower limit.

Comments

- Executable when initiated: YES
- Coupled command: YES; setting the lower state ON will cause LIMit:UPPer[:STATe] to be set OFF (if it is ON).

Note An error will be generated if you have TRIG:SOURce set to INT1-4 and the internal input is the same as the channel you are attempting to enable the lower limit testing. For example, if TRIG:SOUR INT2 is set. The trigger level from channel 2 is the trigger event that is the internal trigger input. CALC:LIMit:LOWer:STATe ON is attempting to use this signal for limit testing and creates a settings conflict. Either the trigger level can be used as an internal trigger or the level can be used in limit testing, but not both.

- Reset (*RST) Condition: OFF

:LIMit:LOWer[:STATe]?

CALCulate[<channel>]:LIMit:LOWer[:STATe]? queries the lower limit checking state to see if it is enabled or disabled for the specified channel. "1" returned indicates the specified channel is enabled for lower limit checking. "0" returned indicates the specified channel is disabled for lower limit checking.

:LIMit:LOWer:DATA

CALCulate[<channel>]:LIMit:LOWer:DATA <*value*> | MIN | MAX sets the lower limit value you want to test against. The CALC<channel>:LIMit:FAIL? command will return a "1" following the measurement (and prior to the next measurement) if the input signal fell below the specified lower limit value and if LIM:LOW:STATe is ON. A "0" is returned if the limit was not exceeded.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
value	numeric	-254 to +252	volts

Comments

- Allowable maximum values for the lower limit by range and the associated resolution is given below:

Range	Maximum value	Resolution
0.0625	±0.061523438	0.000488281
0.250	±0.246093750	0.001953125
1.00	±0.984375000	0.00781250
4.00	±3.937500	0.031250
16.00	±15.750	0.1250
64.00	±63.00	0.500
256.00	±252.00	2.0

- Executable when initiated: No
- Coupled Command: YES; Range changes will change the value; the percent of full scale of the range will be kept constant. For example, on the 4 volt range, with a 2V limit, a range change to 16V will set a new limit of 8V.
- Related Command: [SENSe:]VOLTage[<channel>][:DC]:RANGe <range>
- Reset (*RST) Condition: -254 volts

:LIMit:LOWer:DATA?

CALCulate[<channel>]:LIMit:LOWer:DATA? [MIN | MAX] queries the lower limit value set for the specified channel.

:LIMit:UPPer[:STATe]

CALCulate[<channel>]:LIMit:UPPer[:STATe] OFF | 0 | ON | 1 enables the upper limit checking for the specified channel. Use :LIMit:UPPer:DATA <*value*> to set the actual limit value to be tested against.

Comments

- Executable when initiated: YES
- Coupled command: YES, setting the upper state ON will cause LIMit:LOWer[:STATe] to be set OFF (if it is ON).

Note An error will be generated if you have TRIG:SOURce set to INT1-4 and the internal input is the same as the channel you are attempting to enable the upper limit testing. For example, if TRIG:SOUR INT2 is set. The trigger level from channel 2 is the trigger event that is the internal trigger input. CALC:LIMit:UPPer:STATe ON is attempting to use this signal for limit testing and creates a settings conflict. Either the trigger level can be used as an internal trigger or the level can be used in limit testing, but not both.

- Reset (*RST) Condition: OFF

:LIMit:UPPer[:STATe]?

CALCulate[<channel>]:LIMit:UPPer[:STATe]? queries the upper limit checking state to see if it is enabled or disabled for the specified channel. This command returns the voltage level measured and the detection mode. A returned value of “0” indicates the specified channel is disabled for upper limit checking. “1” returned indicates the specified channel is enabled and will detect signals above the specified upper limit.

:LIMit:UPPer:DATA

CALCulate[<channel>]:LIMit:UPPer:DATA <*value*> | MIN | MAX sets the upper limit value you want to test against. The :LIMit:FAIL? command will return a "1" following the measurement (and prior to the next measurement) if the input signal rose above the specified upper limit value and LIM:UPP:STATe is ON. A "0" is returned if the limit was not exceeded.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
value	numeric	-254 to +252	volts

Comments

- The maximum allowed <*value*> depends on the range setting. An error will occur if you try to set a level that exceeds the range setting.

- Changing the range after setting the limit value will change the limit value. The percent of full scale is kept constant.

- Allowable maximum values for the upper limit by range and the associated resolution is given below:

Range	Maximum value	Resolution
0.0625	0.062011719	0.000488281
0.250	0.248046875	0.001953125
1.00	0.992187500	0.00781250
4.00	3.968750	0.031250
16.00	15.8750	0.1250
64.00	63.50	0.500
256.00	254.00	2.0

- Executable when initiated: No

- Coupled Command: YES; Range changes will change the value; the percent of full scale of the range will be kept constant. For example, on the 4 volt range, with a 2V limit, a range change to 16V will set a new limit of 8V.

- Reset (*RST) Condition: +252 volts

:LIMit:UPPer:DATA?

CALCulate[<channel>]:LIMit:UPPer:DATA? [MIN | MAX] queries the upper limit value set for the specified channel.

The CALibration subsystem allows you to calibrate your digitizer.

Subsystem Syntax

```
CALibration
      :DAC:VOLTage  <voltage> | MIN | MAX
      :DAC:VOLTage?  MIN | MAX
      :DATA?
      :GAIN[<channel>]  [<readings> | DEF][,<rate> | DEF][,ON | 1 | OFF | 0]
      :SOURce  INTernal | EXTernal
      :SOURce?
      :STATe  ON | 1 | OFF | 0
      :STATe?
      :STORe
      :VALue  <voltage>]
      :VALue?
      :ZERO[<channel>]  [<readings>][,<rate>]
      :ZERO[<channel>]:ALL?  [<readings>][,<rate>]
```

:DAC:VOLTage

CALibration:DAC:VOLTage <*voltage*> | MIN | MAX is only active if the CALibration:SOURce is set to INTernal. The voltage specified is output by the internal DAC to the calibration bus (HP E1564A 4-Channel Digitizer only). You can measure this voltage on the top two pins of the front panel D-sub connector (CAL-H and CAL-L) and is used for calibrating the digitizer's gain as the CAL:VALue.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
voltage	numeric	±0.061256409 - ±15.00	volts

Comments

- The maximum output levels are limited to the levels shown in the table below. These are the HP E1564A DAC voltages recommended for calibrating each range. The values are approximately 98% of full scale.

Voltage Range	Max DC Voltage (absolute value)
0.0625	0.061256409
0.2500	0.245025635
1.0000	0.980102539
4.0000	3.920410156
16.0000	15.00
64.0000	not used
256.0000	not used

- There is no calibration DAC output for the 64 volt and 256 volt ranges. See the CALibration:GAIN command for more information about the calibration of these two ranges.

- An error will occur if the voltage value specified is greater than that allowed for the present range setting. You must set the desired range prior to setting the calibration DAC voltage.

- Executable when initiated: No

- Coupled Command: No

- Reset (*RST) Condition: 0.0 volts

:DAC:VOLTage?

CALibration:DAC:VOLTage? MIN | MAX queries the setting of the calibration DAC (HP E1564A 4-Channel Digitizer only). The DAC voltage is output to the calibration bus and accessible at the front panel D-connector only if the CALibration:SOURce is set to INTernal. The MIN parameter returns the minimum voltage available from the DAC and MAX returns the maximum voltage available from the DAC.

:DATA?

CALibration:DATA? returns the calibration constants currently stored in non-volatile calibration memory.

:GAIN

CALibration:GAIN[<*channel*>] [<*readings*>|DEF][,<*rate*>|DEF][,ON|1|OFF|0] initiates a gain calibration on the channel specified. The ON parameter will cause the 64V and 256V ranges to be indirectly calibrated from the 16V range gain calibration. The ON/OFF parameter is ignored except for a gain calibration of the 16V range. The following steps must be completed prior to executing a gain calibration:

- Set the digitizer to the desired range and filter on the channel you want to calibrate with the VOLTage[<channel>]:RANGe <range> and INPut[<channel>]:FILTer:FREQ <freq> and :FILTer:STATe ON|OFF commands.

- Enable calibration with the CALibration:STATe ON command.

- Specify the calibration source with the CALibration:SOURce command.

- Specify a calibration value for the channel you are calibrating. The value must be between 85% and 98% of either a positive full scale reading or negative full scale reading. The ideal calibration value is 98% of positive or negative full scale (see CALibration:DAC:VOLTage command for the 4-Channel HP E1564A internal calibration DAC outputs when CALibration:SOURce INTernal is used).

- The calibration voltage must be applied to the input connector if CALibration:SOURce EXTernal is used. You must enter the external calibrator voltage value with the CAL:VALue command when an external calibration source is used. The HP E1564A 4-Channel Digitizer automatically applies the DAC voltage to the internal calibration bus when CALibration:SOURce INTernal is used. You must measure the DAC voltage at the front panel pins (for CAL:SOURce INTernal) and enter that value with the CAL:VALue command.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
readings	numeric	25 to 4000 | DEFault	none
rate	numeric	1.25E-6 to reference period * 8,388,607 | DEFault	Seconds

Comments

- The number of readings and sampling rate will default to 100 readings and 0.001 second sampling rate respectively, to provide averaging over an integral number of either 50 Hz or 60 Hz power line cycles. This allows calibration to cancel out any noise that is periodic with the power supply.

- The 64V and 256V ranges are calibrated indirectly when the 16V range is calibrated and the ON (1) parameter is set. If the OFF (0) parameter is active, only the 16V range is calibrated and the 64V and 256V ranges retain their old cal constant. This boolean ON/OFF parameter is checked and used only when calibrating the 16V range. It is ignored when calibrating any other range.

- All lower ranges (0.0625 through 4.0000) must be calibrated before calibrating the 16V range and calculating new calibration constants for the 64V and 256V ranges. The effects of the attenuators and amplifiers on the gain calibrations for the lower ranges are extrapolated to derive a gain constant for the 64V range and another for the 256V range.

- The absolute maximum voltages for each range are shown in the next table. The values are approximately 98% of full scale.

Voltage Range	Max DC Voltage (absolute value)
0.0625	0.061256409
0.2500	0.245025635
1.0000	0.980102539
4.0000	3.920410156

Voltage Range	Max DC Voltage (absolute value)
16.0000	15.68164062
64.0000	not used
256.0000	not used

- Optional parameters that are left blank are filled from left to right. Therefore, it is necessary to use the syntax DEFault to note that a particular parameter is to use the default value. For example, to specify a sample rate other than the default, you must declare DEFault for the <readings> parameter or the <rate> parameter value you intended will be used to fill in the <readings> parameter. The command for channel 1 would appear as: CAL:GAIN1 DEF,.002. If you are calibrating the 16V range and you want to recalculate the 64V and 256V calibration constants, the command is: CAL:GAIN1 DEF,.002,ON.

- Executable when initiated: No

- Coupled Command: No

- Reset (*RST) Condition: None

SOURce

CALibration:SOURce INTernal | EXTernal specifies the calibration source to be used for any subsequent gain calibrations. "EXTernal" is the default source, a voltage must be provided from an external source to the channel being calibrated.

Note The "INTernal" source is available only on the HP E1564A 4-Channel Digitizer. CAL:SOURce INTernal outputs the specified DAC voltage set by CAL:DAC:VOLT <*voltage*> onto the calibration bus where it is applied internally to the channels. It is also available on the top two pins of the front panel D-subminiature connector where you must measure the voltage with a transfer standard (accurate voltmeter) and enter the measured value using the CAL:VALue command. The calibration gain command then sets calibration constants for the value you input assuming it is the value on the calibration bus.

- Executable when initiated: No

- Coupled Command: No

- Reset (*RST) Condition: EXTernal

:SOURce?

CALibration:SOURce? queries which calibration source is set. This setting is shared by all channels. Returns either "INT" for INTernal or "EXT" for EXTernal.

:STATe

CALibration:STATe ON | 1 | OFF | 0 enables the calibration of the instrument. Many instrument operations are not allowed when this state is ON and will result in an error “Illegal while calibrating”. You must remember to set the calibration state to OFF when calibration is finished.

Note Sending CAL:STAT OFF, without storing any modified cal constants with the CAL:STORe command, will generate an error. Send the ABORt or *RST command to abort a calibration without storing cal constants.

- Executable when initiated: No
- Coupled Command: No
- Reset (*RST) Condition: OFF

:STATe?

CALibration:STATe? queries the present calibration state of the instrument. A return value of “1” indicates the instrument is enabled and will accept calibration commands and perform calibrations. A return value of “0” indicates the instrument is not calibration enabled and attempting to execute a calibration process command such as CAL:GAIN or CAL:ZERO, will return the error “Calibration not enabled”.

:STORe

CALibration:STORe writes the calibration constants to non-volatile RAM after calibration has been completed.

Note You must have the FLASH and CAL CONSTANTS switches set to the “Write Enable” positions before calibration constants are stored in RAM.

- Executable when initiated: No
- Coupled Command: No
- Reset (*RST) Condition: None

:VALue

CALibration:VALue <*voltage*> specifies the voltage value actually applied to the channel for calibration. This value informs the digitizer what voltage is either being placed on the front panel input connector (CAL:SOURce EXTernal) or the value being generated by the internal DAC (HP E1564A 4-Channel Digitizer only) and being output onto the calibration bus.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
voltage	numeric	±0.061256409 - ±15.6800	volts

Comments

The maximum voltage from an external source used to calibrate the 16V range is 15.68V or 98% of full scale. The maximum voltage attainable from the E1564A internal DAC is 15V. See the following paragraphs.

HP E1564A 4-Channel Digitizer Internal DAC

The internal DAC on the HP E1564A can be used for the calibration source when CAL:SOURce INTernal is specified. The output level of this DAC is specified with the CAL:DAC:VOLTage command. The actual output level must be measured with a voltmeter by the person doing the calibration and that measured value is the input for the **<*voltage*>** parameter of this CAL:VALue command. The voltage can be measured across pins 5 (high) and 9 (low) of the D-subminiature calibration bus connector.

The maximum output levels are limited to the levels shown in the table below. These are the HP E1564A DAC voltages recommended for calibrating each range. The values are approximately 98% of full scale (except for the 16V range which the internal E1564A's DAC has a maximum output of ±15V.

Voltage Range	Max DC Voltage (absolute value)
0.0625	0.061256409
0.2500	0.245025635
1.0000	0.980102539
4.0000	3.920410156
16.0000	15.00
64.0000	not used
256.0000	not used

- Executable when initiated: No
- Coupled Command: No
- Reset (*RST) Condition: All channels set to 0.0 volts

:VALue?

CALibration:VALue? queries the present setting of the calibration voltage.

:ZERO

CALibration:ZERO[<*channel*>] [<*samples*>][,<*rate*>] initiates an offset calibration *for the current range* on the specified channel using an internal short. The following steps must be completed prior to executing a zero calibration:

- Set the CAL:STATe ON to allow calibration to occur.
- Set the digitizer to the desired range and filter on the channel you want to calibrate with the VOLTage[<channel>]:RANGe <range> and INPut[<channel>]:FILTer:FREQ <freq> and :FILTer:STATe ON|OFF commands.
- Errors will result if the above steps are not performed before CAL:ZERO.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
samples	numeric	25 to 4000 | DEFault	none
rate	numeric	1.25E-6 to reference period * 8,388,607 | DEFault	Seconds

Comments

- Optional parameters that are left blank are filled from left to right. Therefore, it is necessary to use the syntax DEFault to note that a particular parameter is to use the default value. For example, to specify a sample rate other than the default, you must declare DEFault for the <readings> parameter or the <rate> parameter value you intended will be used to fill in the <readings> parameter. The command for channel 1 would appear as: CAL:ZERO1 DEF,.002.
- The number of samples and the sample rate would normally be set to DEFault values to provide averaging over an integral number of either 50 Hertz or 60 Hertz power line cycles. This allows the calibration to cancel out any noise that is periodic with the power supply. Specifying a value other than DEF for <*samples*> and/or <*rate*> will result in those values being used for the zero offset calibration.
- Executable when initiated: No
- Coupled Command: No
- Reset (*RST) Condition: None

CALibration:ZERO[<*channel*>]:ALL? [<*samples*>][,<*rate*>] initiates a zero offset calibration *for all ranges* on the specified channel using an internal short. The command returns “0” if the calibration was successful. It returns a non-zero value if an error occurred while calibrating one of the ranges. The non-zero return value contains the failed ranges as high bits in the lower word. For example, a return value of 0000000000100001 has a lower word of 00100001 which indicates range 0 (bit 0 = 0.0625V) and range 5 (bit 5 = 64V) failed. The error string in SYST:ERR? contains information about the failure on the highest range that failed (range 5, 64V). The following steps must be completed prior to executing a zero calibration:

- Set the CAL:STATe ON to allow calibration to occur.
- Set the digitizer to the desired filter on the channel you want to calibrate with the INPut[<channel>]:FILTer:FREQ <freq> and :FILTer:STATe ON|OFF commands.
- Errors result if the above steps are not performed before CAL:ZERO:ALL?.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
samples	numeric	25 to 4000 | DEFault	none
rate	numeric	1.25E-6 to reference period * 8,388,607 | DEFault	Seconds

Comments

- Optional parameters that are left blank are filled from left to right. Therefore, it is necessary to use the syntax DEFault to note that a particular parameter is to use the default value. For example, to specify a sample rate other than the default, you must declare DEFault for the <readings> parameter or the <rate> parameter value you intended will be used to fill in the <readings> parameter. The command for channel 1 would appear as: CAL:ZERO1 DEF,.002.
- The number of samples and the sample rate would normally be set to DEFault values to provide averaging over an integral number of either 50 Hertz or 60 Hertz power line cycles. This allows the calibration to cancel out any noise that is periodic with the power supply. Specifying a value other than DEF for <*samples*> and/or <*rate*> will result in those values being used.
- If an error occurs on any range, calibration proceeds on to the next range, and the bad range is noted.
- Executable when initiated: No
- Coupled Command: No
- Reset (*RST) Condition: None

DIAGnostic

The DIAGnostic subsystem contains several commands which were developed to test the instrument at the factory. Some of these commands may prove useful for isolating problems or for use in special applications.

Subsystem Syntax

```
DIAGnostic
    :DAC:OFFSet[<channel>]  <voltage>
    :DAC:OFFSet[<channel>]:RAMP  <count>
    :DAC:GAIN[<channel>]  <value>
    :DAC:SOURce  <voltage>
    :DAC:SOURce:RAMP  <count>
    :INTerrupt:LINE  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7
    :INTerrupt:LINE?
    :MEMory:SIZE  <size>
    :MEMory:SIZE?
    :PEEK?  <reg_number>
    :POKE?  <reg_number>,<data>
    :SHORt[<channel>]  ON | 1 | OFF | 0
    :SHORt[<channel>]?
    :STATus?
```

:DAC:OFFSet

DIAGnostic:DAC:OFFSet[<*channel*>] <*voltage*> writes the specified voltage value to the calibration offset DAC of the specified channel when the DAC:GAIN command is sent. This offset voltage value is not used unless a DAC:GAIN <*value*> is sent to the calibration gain DAC. This command is a factory diagnostic routine.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
voltage	numeric	-2.5 to +2.5	none

Comments

- Executable when initiated: No
- Coupled Command: No
- Reset (*RST) Condition: None

:DAC:OFFSet:RAMP

DIAGnostic:DAC:OFFSet[<*channel*>]:RAMP <*count*> outputs to the specified channel, a ramp of DAC values from 0 to 255 with the DAC code changing approximately every 100 µS. This command is a factory diagnostic routine.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
count	numeric	1 to 32767	none

Comments

- The “count” parameter defines the number of ramps to output. Approximately 37.35 full ramps are output each second.
- A count of 2240 will output ramps for approximately 60 seconds.
- Executable when initiated: No
- Coupled Command: No
- Reset (*RST) Condition: None

:DAC:GAIN

DIAGnostic:GAIN[<*channel*>] <*value*> writes the specified value to the calibration gain DAC of the specified channel. There *must* be a signal on the input for this to work properly. Any offset value set by the DAC:OFFSet <*voltage*> is used by the DAC when the DAC:GAIN command is sent. The gain is set on the specified channel. This command is a factory diagnostic routine.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
value	numeric	0 to 255	none

Comments

- A positive full scale input combined with a DAC gain value of 255 will result in a +2.5V output from the DAC.
- A negative full scale input combined with a DAC gain value of 255 will result in a -2.5V output from the DAC.
- A DAC gain value of 0 will result in 0V output in both cases.
- Executable when initiated: No
- Coupled Command: No
- Reset (*RST) Condition: None

:DAC:SOURce

DIAGnostic:DAC:SOURce <*voltage*> outputs the specified voltage from the internal calibration source DAC onto the cal pins of the front panel connector. This command is a factory diagnostic routine.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
voltage	numeric	-15.0 to +15.0	none

Comments

- The channel's input relay remains open until it is closed by an INPut:STATe ON command or by a reset of the instrument.
- Executable when initiated: No
- Coupled Command: No
- Reset (*RST) Condition: DAC output is set to 0V

:DAC:SOURce:RAMP

DIAGnostic:DAC:SOURce:RAMP <*count*> outputs a ramp of DAC values from 0 to 4095 with the DAC code changing about every 100 µS. This command is a factory diagnostic routine.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
count	numeric	1 through 255	none

Comments

- The "count" variable specifies how many ramps to output. The timing is such that about 2.3257 full ramps are output each second.
- A count of 139 will output ramps for just under 60 seconds.
- The signal will be output onto the cal pins on the front panel D-subminiature connector.
- Executable when initiated: No
- Coupled Command: No
- Reset (*RST) Condition: DAC output is set to 0V

:INTerrupt:LINE

DIAGnostic:INTerrupt:LINE 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 sets the interrupt line to be used. Specifying the "0" parameter disables interrupts.

Note The STATus subsystem will not work if interrupts are disabled (STATus: OPERation and STATus:QUEStionable commands). Use the DIAG:STATus? command when you disable interrupts.

Comments

- The "0" parameter disables all interrupts.
- Power-on default setting is interrupt line "1".
- Executable when initiated: No
- Coupled Command: No
- Reset (*RST) Condition: Interrupt line setting is unchanged.

:INTerrupt:LINE?

DIAGnostic:INTerrupt:LINE? queries the interrupt line setting. Returns a number "0" through "7" to indicate interrupt line 1 through 7. A "0" returned indicates interrupts are disabled. See Note in preceding DIAG:INTerrupt:LINE.

:MEMory:SIZE

DIAGnostic:MEMory:SIZE <*size*> sets the memory size value in calibration memory. Your module comes standard with 4 Mbytes of RAM. You can replace this with PC SIMM modules of up to 128 Mbytes. See Chapter 1 for the procedure for adding RAM to your module.

Note This command is required and used only when you change the size of RAM on the module. You then use this command to set the new memory size value in calibration memory.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
size	numeric	4E6, 8E6, 16E6, 32E6, 64E6 and 128E6	none

Comments The <*size*> parameter will accept a value in excess of the industry notation value of

4M, 8M, 16M, etc. (e.g., 4E6, 8E6, 16E6, etc.) up to the actual size as noted in the following DIAGnostic:MEMory:SIZE? command.

:MEMory:SIZE?

DIAGnostic:MEMory:SIZE? queries the RAM size value in calibration memory. The value returned is the actual amount of memory, not the abbreviated industry notation for memory size. The value returned for each size is shown below:

RAM industry notation	actual size value
4M	4,194,304
8M	8,388,608
16M	16,777,216
32M	33,554,432
64M	67,108,864
128M	134,217,728

:PEEK?

DIAGnostic:PEEK? <reg_number> queries the specified register and returns the contents of the register.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
reg_number	numeric	0 to 31	none

Comments

- See Appendix B for register bit definitions.
- You can read the following digitizer registers using the register number:

reg_number	Register Description (base + register offset)
0	Manufacturer ID Register (base + 00_{16})
1	Device Type Register (base + 02_{16})
2	Status/Control Register (base + 04_{16})
3	Offset Register (base + 06_{16})
4**	FIFO High Word Register (base + 08_{16})
5**	FIFO Low Word Register (base + $0A_{16}$)
6	Interrupt Control Register (base + $0C_{16}$)
7	Interrupt Sources Register (base + $0E_{16}$)
8	CVTable Channel 1 Register (base + 10_{16})
9	CVTable Channel 2 Register (base + 12_{16})
10	CVTable Channel 3 Register (base + 14_{16})
11	CVTable Channel 4 Register (base + 16_{16})

12 Samples Taken High Word Register (base + 18_{16})
13 Samples Taken Low Word Register (base + $1A_{16}$)
14 Calibration Flash ROM Address Register (base + $1C_{16}$)
15 Calibration Flash ROM Data Register (base + $1E_{16}$)
16 Calibration Source Register (base + 20_{16})
17 Cache Count Register (base + 22_{16})
18 Range, Filter, Connect Channels 1 and 2 Register (base + 24_{16})
19 Range, Filter, Connect Channels 3 and 4 Register (base + 26_{16})
20 Trigger/Interrupt Level Channel 1 Register (base + 28_{16})
21 Trigger/Interrupt Level Channel 2 Register (base + $2A_{16}$)
22 Trigger/Interrupt Level Channel 3 Register (base + $2C_{16}$)
23 Trigger/Interrupt Level Channel 4 Register (base + $2E_{16}$)
24 Sample Period High Word Register (base + 30_{16})
25 Sample Period Low Word Register (base + 32_{16})
26 Pre-Trigger Count High Register (base + 34_{16})
27 Pre-Trigger Count Low Register (base + 36_{16})
28 Post-Trigger Count High Register (base + 38_{16})
29 Post-Trigger Count Low Register (base + $3A_{16}$)
30 Trigger Control/Source Register (base + $3C_{16}$)
31 Sample Control/Source Register (base + $3E_{16}$)

- DIAG:PEEK? 4 or DIAG:PEEK? 5 may cause an error if they are read before data has been taken.

Example Read the Manufacturers ID register:

DIAG:PEEK? 0

returns -12289 (decimal) or FFFFFCFFF (hexadecimal). The three least significant characters (FFF) indicates a Hewlett-Packard A16 register-based module.

:POKE

DIAGnostic:POKE ***<reg_number>,<data>*** places the specified value in the specified register.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
reg_number	numeric	2-4, 14-16, 18-31	none
data	numeric	-32768 to +32767 signed integer 0 to 65535 unsigned integer	none

Comments

- See Appendix B for register bit definitions.
- You can write to the following digitizer registers using the register number:

reg_number	Register Description (base + register offset)
2	Status/Control Register (base + 04_{16})
3	Offset Register (base + 06_{16})
6	Interrupt Control Register (base + $0C_{16}$)
14	Calibration Flash ROM Address Register (base + $1C_{16}$)
15	Calibration Flash ROM Data Register (base + $1E_{16}$)
16	Calibration Source Register (base + 20_{16})
18	Range, Filter, Connect Channels 1 and 2 Register (base + 24_{16})
19	Range, Filter, Connect Channels 3 and 4 Register (base + 26_{16})
20	Trigger/Interrupt Level Channel 1 Register (base + 28_{16})
21	Trigger/Interrupt Level Channel 2 Register (base + $2A_{16}$)
22	Trigger/Interrupt Level Channel 3 Register (base + $2C_{16}$)
23	Trigger/Interrupt Level Channel 4 Register (base + $2E_{16}$)
24	Sample Period High Word Register (base + 30_{16})
25	Sample Period Low Word Register (base + 32_{16})
26	Pre-Trigger Count High Register (base + 34_{16})
27	Pre-Trigger Count Low Register (base + 36_{16})
28	Post-Trigger Count High Register (base + 38_{16})
29	Post-Trigger Count Low Register (base + $3A_{16}$)
30	Trigger Control/Source Register (base + $3C_{16}$)
31	Sample Control/Source Register (base + $3E_{16}$)

- Executable when initiated: No
- Coupled Command: No
- Reset (*RST) Condition: None

Example Write to the Range, Filter, Connect Channels 1 and 2 register to set channel 1 and 2 ranges to 64V and set the filters to 100 kHz:

DIAG:POKE 18,13621

The binary bit pattern for +13621 is 0011010100110101

:SHORt

DIAGnostic:SHORt[<*channel*>] ON | 1 | OFF | 0 connects an internal short across the input of the specified channel when the "ON" or "1" parameter is used. The internal short is enabled by "ON" or "1" and disabled by "OFF" or "0".

Comments

- The short remains in effect until a reset or until it is disabled with a DIAG:SHORt[<*channel*>] OFF command.
- Executable when initiated: No
- Coupled Command: No
- Reset (*RST) Condition: Short OFF

:SHORt?

DIAGnostic:SHORt[<*channel*>]? queries the specified channel to determine if the internal short is connected. This command returns “1” if the short is present or returns “0” if it is not present.

:STATus?

DIAGnostic:STATus? returns the status of bits in the instrument's interrupt sources register (offset 08h; see Appendix B). A high value in a bit location indicates a particular event has occurred. The bit positions and their meanings are as follows:

Bit	Event Represented When Bit is High
0	Channel 1 limit was exceeded or channel 1 trigger level was exceeded.
1	Channel 2 limit was exceeded or channel 2 trigger level was exceeded.
2	Channel 3 limit was exceeded or channel 3 trigger level was exceeded.
3	Channel 4 limit was exceeded or channel 4 trigger level was exceeded.
4	An input overload occurred and the input relay opened.
5	The pre-trigger count has been met.
6	The measurement has completed normally, or available memory has been filled and the measurement was halted.
7	A valid trigger event was received after the pretrigger acquisition (if any) was completed.

- This command returns a binary-weighted number representing the bit pattern of the register and therefore, the status of the above instrument events.
- Executable when initiated: No
- Coupled Command: No
- Reset (*RST) Condition: None

The FORMat command subsystem is used to specify the output format of the readings from the HP E1563A/E1564A Digitizer.

Subsystem Syntax

```
FORMat
    [:DATA]   ASCii | PACKed | REAL
    [:DATA]?
```

[:DATA]

FORMat[:DATA] ASCii | PACKed | REAL specifies the output format for measurement data.

Comments

- PACKed,16 format is signed 16 bits (16-bit integers). Data is returned as raw data and must be converted to voltage by one of the following methods:

  voltage = reading * range/32768

  voltage = reading * resolution
  NOTE: Use the [SENSe:]VOLTage[:DC]:RESolution? command to obtain the resolution value.

- REAL,64 format sends data back as IEEE-754 64-bit real numbers.

- Both PACKed,16 and REAL,64 formats return data preceded by the IEEE-488.2 definite length arbitrary block header. The header is as follows:

  # <num_digits> <num_bytes>
  # signifies a block transfer
  <num_digits> is a single digit (1 through 9) which specifies how many digits (ASCII characters) are in the <num_bytes> descriptor which follows.
  <num_bytes> is the number of data bytes which immediately follow the <num_bytes> field.

- Reset (*RST) Condition: FORMat:DATA ASCii

[:DATA]?

FORMat[:DATA]? queries the type of output format set for measurement data.

Comments

The command returns “ASC,+7”, “PACK,+16”, or “REAL,+64”.

- ASC,+7 indicates ASCII data with seven significant digits.

- PACKed,+16 indicates the format is signed 16 bits.

- REAL,+64 indicates data is IEEE-754 64-bit real numbers.

The INITiate subsystem controls the initiation of the trigger system and prepares the Digitizer to take voltage measurements. Once a trigger is received from the programmed source (TRIGger:SOURce command), measurements begin on all channels. Normally, all measurement setup (setting measurement ranges, sample count and trigger sources, etc.) should be done before this command is sent. Sending this command will cause the Digitizer to begin the measurement process.

Subsystem Syntax

```
INITiate
    [:IMMediate]
    :CONTinuous   ON | 1 | OFF | 0
    :CONTinuous?
```

[:IMMediate]

INITiate[:IMMediate] initiates the trigger system and prepares the Digitizer to take voltage measurements. After initiation, the Digitizer enters the wait-for-trigger state and begins taking pretrigger readings until the pretrigger count is met (if there is a pretrigger count set). All incoming triggers are ignored until the pretrigger count is met. Pretrigger readings continue until a trigger arrives. The first trigger received after the pretrigger readings have been acquired is the one accepted and it advances the digitizer to the wait-for-sample state which is where readings are actually taken. When the number of readings specified by the TRIGger:COUNt and SAMPle:COUNt have been taken, the trigger system returns to the idle state and digitizer stops measuring.

Comments

- Executable when initiated: No
- Coupled Command: No
- Reset (*RST) Condition: Idle state

:CONTinuous

INITiate:CONTinuous ON | 1 | OFF | 0 is used to either start or stop a continuous measurement. The ON (1) setting will start a measurement with an infinite sample count. After initiation, the Digitizer enters the wait-for-trigger state and begins taking pretrigger readings until the pretrigger count is met (if there is a pretrigger count set). All incoming triggers are ignored until the pretrigger count is met. Pretrigger readings continue until a trigger arrives. The first trigger received after the pretrigger readings have been acquired is the one accepted and it advances the digitizer to the wait-for-sample state which is where readings are actually taken. The incoming trigger advances the Digitizer to the wait-for-sample state which is where readings are actually taken. The instrument will continuously sample until one of the following three things occurs:

1. The measurement is stopped by the ABORt command.

2. The measurement is stopped by executing INITiate:CONTinuous OFF.

3. The instrument's FIFO memory is filled. This can be prevented by fetching the data from memory in blocks faster than the sample rate can fill memory.

Note The INIT[:IMMediate] and INIT:CONTinuous commands return "1" to *OPC? when the instrument begins measurement not when measurements complete. To determine when a non-continuous measurement is complete, use the DIAG:STATus? command and monitor bit 6. You can also detect when measurements are complete by monitoring the "measurement complete" bit (bit 9) of the STATus:OPERation:CONDition register in the STATus system. See the STATus command. *WAI, *OPC and *OPC? will all be fulfilled immediately after the INIT command is processed not when the measurements are complete.

Comments

- Executable when initiated: No
- Coupled Command: No
- Reset (*RST) Condition: Idle state

:CONTinuous?

INITiate:CONTinuous? queries the instrument to determine if the INITiate:CONTinuous is enabled or disabled.

The INPut command subsystem controls characteristics of the input signal, including ON/OFF state and low-pass filtering. The command defaults to channel 1 if you do not specify a channel in the command syntax (e.g., INP ON is same as INP1 ON).

Subsystem Syntax

```
INPut[<channel>]
    :FILTer[:LPASs]:FREQ   1.5E3 | 6E3 | 25E3 | 100E3 (valid for E1564A only)
    :FILTer[:LPASs]:FREQ?
    :FILTer[:LPASs][:STATe]   ON | 1 | OFF | 0
    :FILTer[:LPASs][:STATe]?
    [:STATe]   ON | 1 | OFF | 0
    [:STATe]?
```

:FILTer[:LPASs]:FREQ

INPut[<*channel*>]:FILTer[:LPASs]:FREQ 1.5E3 | 6E3 | 25E3 | 100E3 command sets the filter frequency for the 4-channel E1564A Digitizer. The filters are 2-pole Bessel filters. <*channel*> is 1 through 4. The 2-channel E1563A Digitizer only has a fixed 25 kHz filter but will accept this command without error.

Note The 2-channel E1563A Digitizer has a fixed 25 kHz filter. The E1563A will accept this command but cannot change the filter and will not generate an error.

Comments

- In the E1564A 4-channel digitizer, the filter will be set to the nearest value that can be achieved by the value specified in the command e.g., if you specify 10E3, the filter is set to 6K, if you specify 20E3, the filter is set to 25K. In the E1563A 2-channel digitizer, the filter will be 25 kHz regardless of what value you input (see above note).
- Executable when initiated: No
- Coupled Command: No
- Reset (*RST) Condition: filter state OFF

:FILTer[:LPASs]:FREQ?

INPut[<*channel*>]:FILTer[:LPASs]:FREQ? command queries the present filter frequency setting on the specified channel.

:FILTer[:LPASs][:STATe]

INPut[<*channel*>]:FILTer[:LPASs][:STATe]   ON | 1 | OFF | 0 command is used to enable or disable the low-pass filter on the specified channel.

Comments Executable Command: No

Couple Command: No

Reset (*RST) Condition: filter OFF

:FILTer[:LPASs][:STATe]?

INPut[<*channel*>]:FILTer[:LPASs][:STATe]? command is used to query the specified channel to determine if the low-pass filter is enabled or disabled. A return value of “0” indicates the filter is OFF, “1” indicates the filter is ON.

[:STATe]

INPut[<*channel*>][:STATe]   ON | 1 | OFF | 0 is used to connect or disconnect the input signal to the Digitizer’s measurement circuitry.

Note **OFF State Connections:**

HP E1563A 2-Channel Digitizer:
INPut<*channel*>:STATe OFF connects the specified channel to ground.

HP E1564A 4-Channel Digitizer:
INPut<*channel*>:STATe OFF connects the specified channel to the internal calibration bus (calibration DAC).

Comments Executable when intitiated: No

Coupled command: No

Reset (*RST) Condition: all channels ON (connected)

[:STATe]?

INPut[<*channel*>][:STATe]? command queries the specified channel to determine if the input signal is connected to, or disconnected from, the Digitizer’s measurement circuitry. If connected, a “1” is returned; if disconnected, a “0” is returned.

The OUTPut command subsystem sets the source of output pulses for the specified TTL Trigger line (TTLT0-TTLT7) and enables or disables the output.

Subsystem Syntax

```
OUTPut
    :TTLT<n>:SOURce    TRIGger | SAMPle | BOTH
    :TTLT<n>:SOURce?
    :TTLT<n>[:STATe]    ON | 1 | OFF | 0
    :TTLT<n>[:STATe]?
```

:TTLT<n>:SOURce

OUTPut:TTLT<n>:SOURce TRIG | SAMP | BOTH sets the source of output pulses for the specified TTL Trigger line.

Comments

- <n> can have the value 0 through 7 (TTLT0 - TTLT7).
- The Digitizer allows separate control of the trigger signal and the sample signal output to the TTL trigger lines. Each can output to only a single line; however, they can both output onto the same line when the BOTH parameter is used. When BOTH is used, no other lines can be enabled.
- Output pulses will not be sent until the TTL trigger line state is set to ON.
- Resource conflicts will occur if either the trigger or sample source is already using a TTL line you attempt to enable. The trigger source will be set to IMMediate if it is the conflict. The sample source will be set to TIMer if it is the conflict. A “Settings Conflict” error will occur.
- Setting the trigger or sample source to a TTL trigger line that has its output state ON will result in a “Settings Conflict” error and the output state will be changed to OFF. The specified trigger line will be assigned to the sample or trigger source.
- Executable when initiated: No
- Coupled Command: Yes
- Reset (*RST) Condition: Source is SAMPle for all TTL lines

:TTLT<n>:SOURce?

OUTPut:TTLT<n>:SOURce? queries the specified TTL Trigger line (TTLT0-TTLT7) to identify the source of output pulses. A response of “TRIG” indicates the source is a trigger event, a “SAMP” indicates the source is a sample event and a response of “BOTH” indicates the source is both a trigger event and a sample event.

:TTLT<n>[:STATe]

OUTPut:TTLT<n>[:STATe] ON | 1 | OFF | 0 enables or disables the specified TTL Trigger line for outputting the source set by the OUTPut:TTLT<n>:SOURce command.

Comments

- <n> can have the value 0 through 7 (TTLT0 - TTLT7).
- Resource conflicts will occur if either the trigger or sample source is already using a TTL line you attempt to enable as an OUTPut line. The OUTPut TTLT line will not be enabled and a "Settings Conflict" error will occur.
- Setting the trigger or sample source to a TTL trigger line that has its output state ON will result in a settings conflict error and the output state will be changed to OFF. The specified trigger line will be assigned to the sample or trigger source.
- TRIG:MODE MASTer<n> | SLAVe<n> will disable all other OUTPut:TTLT<n>:STATe settings. The only outputs that will occur are those defined in the MASTer-SLAVe relationship.
- Executable when initiated: No
- Coupled Command: Yes
- Reset (*RST) Condition: All lines set to OFF.

:TTLT<n>[:STATe]?

OUTPut:TTLT<n>[:STATe]? queries the specified TTL Trigger line (TTLT0-TTLT7) to determine if it is enabled (1) or disabled (0).

The SENSe command subsystem is used to change low-level parameters such as voltage range, sweep and sweep offset points and to set the reference oscillator source and frequency. It is also used to obtain measurement data from the module.

Subsystem Syntax

[SENSe:]
DATA? <*rdgs_per_channel*>[,*channel_list*]
DATA:ALL? <*rdgs_per_channel*>
DATA:COUNt?
DATA:CVTable? [*channel_list*]
ROSCillator:EXTernal:FREQuency <*freq*>|
ROSCillator:EXTernal:FREQuency?
ROSCillator:SOURce INTernal | EXTernal
ROSCillator:SOURce?
SWEep:POINts <*neg_value*> | MIN | MAX
SWEep:POINts? MIN | MAX
SWEep:OFFSet:POINts <*neg_value*> | MIN | MAX
SWEep:OFFSet:POINts? MIN | MAX
VOLTage[<channel>][:DC]:RANGe <*range*> | MIN | MAX
VOLTage[<channel>][:DC]:RANGe?
VOLTage[<channel>][:DC]:RESolution?

DATA?

[SENSe:]DATA? <*rdgs_per_channel*>[,*channel_list*] command returns voltage formatted data from all channels (default) or only from the specified channel list.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
rdgs_per_channel	numeric	1 to MAX samples which depends on size of RAM on module (see SAMPle:COUNt command)	none
channel_list	numeric	1-2 (E1563A) 1-4 (E1564A)	N/A

Comments

- *channel_list* has the form (@1) or (@2), (@1,2), (@1:4) or (@1,2,3,4) or for specific channels but not all, the format is (@1,3,4)

- The readings are returned in an array in an interleaved configuration. That is, the array contains the first reading from each specified channel followed by the second reading from each specified channel. The readings are in channel number order starting with the lowest to highest specified channel in the channel list. For example, the channel list (@2,1) returns channel 1 readings followed by channel 2 readings and returns the same as channel list (@1,2).

Note

1. Measurement data on channels not in the specified channel list is thrown away by this command and is not recoverable.
2. This command can read the data from a measurement only once. It is a destructive read and the data cannot be retrieved a second time.

- The number of readings this command will return for each channel is determined by the number of samples set by the SAMPle:COUNt command. The total number of readings returned is the number of samples X the number of specified channels. If a measurement is aborted with the ABORt command, there may be less readings available than indicated by (samples X channels). For ABORted measurements, use DATA:COUNt? to determine how many readings are available.

- Data is returned as raw data (16-bit integers) when the data format is set to PACKed (see the FORMat[:DATA] PACKed command). Use either of the two methods shown below to convert the raw readings to voltages:

  voltage = reading * range/32768

  voltage = reading * resolution
  NOTE: Use the [SENSe:]VOLTage[:DC]:RESolution? command to obtain the resolution value.

- Data is returned as real numbers when the data format is set to REAL (see the FORMat[:DATA] REAL command). The data is returned in voltage units and no scaling conversion is required as with the PACKed format. Readings are in an interleaved configuration as described in the first bullet item.

- Both PACKed and REAL formats return data preceded by the IEEE-488.2 definite length arbitrary block header. The header is as follows:

  # <num_digits> <num_bytes>

  # signifies a block transfer
  <num_digits> is a single digit (1 through 9) which specifies how many digits (ASCII characters) are in the <num_bytes> descriptor which follows
  <num_bytes> is the number of data bytes which immediately follow the <num_bytes> field

- A full scale reading may actually be an overload.

- A deadlock can occur when trigger events are set to BUS or HOLD because a software trigger could not break in after this command is sent.

- Executable when initiated: Yes

- Coupled command: No

- Reset (*RST) condition: none

DATA:ALL?

[SENSe:]DATA:ALL? <*rdgs_per_channel*> command returns voltage formatted data from each active channel.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
rdgs_per_channel	numeric	1 to 32M** (E1563A) 1 to 16M** (E1564A)	none

**(memory size in bytes) / (number of channels * 2) = 128M/4 or 128M/8 (MAX)

Comments

- The readings are returned in an array in an interleaved configuration. That is, the array contains the first reading from channel 1, channel 2, etc. This is followed by the second reading from channel 1, channel 2, etc.

Note This command can read the data from a measurement only once. It is a destructive read and the data cannot be retrieved a second time.

- The number of readings this command will return for each channel is determined by the number of samples set by the SAMPle:COUNt command. The total number of readings returned is the number of samples X the number of channels. If a measurement is aborted with the ABORt command, there may be less readings available than indicated by (samples X channels). For ABORted measurements, use DATA:COUNt? to determine how many readings are available.

- Data is returned as raw data (16-bit integers) when the data format is set to PACKed (see the FORMat[:DATA] PACKed command). Use either of the two methods shown below to convert the raw readings to voltages:

  voltage = reading * range/32768

  voltage = reading * resolution
  NOTE: Use the [SENSe:]VOLTage[:DC]:RESolution? command to obtain the resolution value.

- Data is returned as real numbers when the data format is set to REAL (see the FORMat[:DATA] REAL command). The data is returned in voltage units and no scaling conversion is required as with the PACKed format. Readings are in an interleaved configuration as described in the first bullet item.

- Both PACKed and REAL formats return data preceded by the IEEE-488.2 definite length arbitrary block header. The header is as follows:

  # <num_digits> <num_bytes>

signifies a block transfer
<num_digits> is a single digit (1 through 9) which specifies how many digits (ASCII characters) are in the <num_bytes> descriptor which follows
<num_bytes> is the number of data bytes which immediately follow the <num_bytes> field

- A full scale reading may actually be an overload.
- A deadlock can occur when trigger events are set to BUS or HOLD because a software trigger could not break in after this command is sent.
- Executable when initiated: Yes
- Coupled command: No
- Reset (*RST) condition: none

DATA:COUNt?

[SENSe:]DATA:COUNt? command returns the number of readings available to be read by the DATA? command per channel. This is useful for determining the amount of data taken in an aborted measurement. The data count from a completed measurement is equal to the sample count set by the SAMPle:COUNt command.

DATA:CVTable?

[SENSe:]DATA:CVTable? *@channel_list* command returns the **most recent** reading taken from each specified channel. The last reading, or **C**urrent **V**alue, from each channel is returned in channel number order starting with the first one in the list.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
channel_list	numeric	1-2 (E1563A) 1-4 (E1564A)	N/A

Comments

- *channel_list* has the form (@1) or (@2), (@1,2), (@1:4) or (@1,2,3,4) or for specific channels but not all, the format is (@1,3,4). If you specify channels not in ascending order e.g., (@2,1) or (@3,4,2), they are rearranged as 1,2 or 2,3,4 respectively.
- Data is returned as raw data (16-bit integers) when the data format is set to PACKed (see the FORMat[:DATA] PACKed command). Use either of the two methods shown below to convert the raw readings to voltages:

  voltage = reading * range/32768

  voltage = reading * resolution

NOTE: Use the [SENSe:]VOLTage[:DC]:RESolution? command to obtain the resolution value.

- Data is returned as real numbers when the data format is set to REAL (see the FORMat[:DATA] REAL command). The data is returned in voltage units and no scaling conversion is required as with the PACKed format.

- Both PACKed and REAL formats return data preceded by the IEEE-488.2 definite length arbitrary block header. The header is as follows:

  # <num_digits> <num_bytes>
  # signifies a block transfer
  <num_digits> is a single digit (1 through 9) specifying how many digits (ASCII characters) are in the <num_bytes> descriptor which follows
  <num_bytes> is the number of data bytes which immediately follow the <num_bytes> field

ROSCillator:EXTernal:FREQuency

[SENSe:]ROSCillator:EXTernal:FREQuency ***<freq>*** is used to specify the externally supplied timebase frequency. This command is not required unless ROSCillator:SOURce is EXTernal. The default timebase is the INTernal timebase.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
freq	numeric	9.9E3 to 30E6	Hz

Comments

- The frequency parameter value is used to calculate sample periods when the sample source is set to TIMer.

- The sample period must be at least 1.250E-6 seconds (800 kHz), and must be an integral multiple of the timebase period (this is 1.0E-7 seconds when the timebase source is INTernal). Period values will be rounded to the nearest period the instrument can obtain.

- Executable when initiated: NO

- Coupled command: NO

- Reset (*RST) Condition: frequency = 10.0 MHz

ROSCillator:EXTernal:FREQuency?

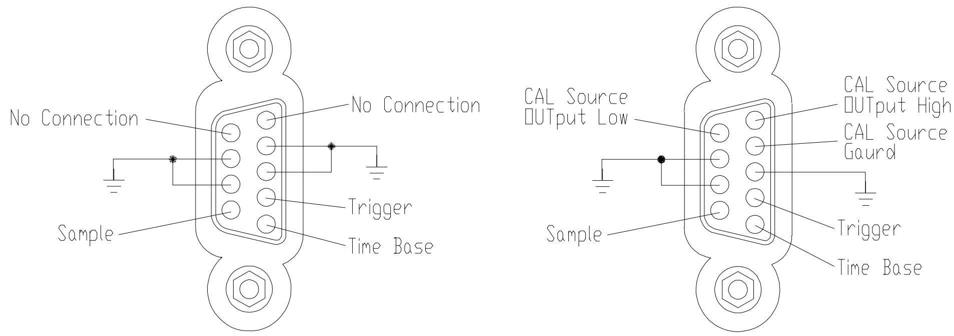
[SENSe:]ROSCillator:EXTernal:FREQuency? queries the external frequency.

ROSCillator:SOURce

[SENSe:]ROSCillator:SOURce INTernal | EXTernal is used to specify the timebase source. The default timebase is the INTernal timebase which uses the VXI CLK10, 10 MHz reference. The EXTernal input is the TTL "Time Base" input pin on the front panel D-subminiature connector (right pin column, bottom pin).

HP E1563A **HP E1564A**

("Time Base" input - bottom right pin)



Note The EXTernal source requires you also send the ROSC:EXT:FREQ <*freq*> command to specify the frequency of the external timebase.

Comments

- The timebase reference set by SAMPle:TIMer <*interval*> is used when the sample source is TIMer (SAMPle:SOURce TIMer).
- Executable when initiated: NO
- Coupled command: Yes, The SAMPle:TIMer <*interval*> is set to a period or interval nearest the old value when source is changed from EXTernal to INTernal or vice versa.
- Reset (*RST) Condition: INTernal source, freq = 10.0 MHz

ROSCillator:SOURce?

[SENSe:]ROSCillator:SOURce? queries to determine the timebase source. Returns either INTernal or EXTernal.

SWEep:POINts

[SENSe:]SWEep:POINts <*count*> | MIN | MAX command sets the number of sweep points. The number of points set is common to all channels. You cannot have two different channels with different a sweep point count.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
<*count*>	numeric	1 to 32M** (E1563A) 1 to 16M** (E1564A)	N/A

**(memory size in bytes) / (number of channels * 2) = 128M/4 or 128M/8 (MAX)

Comments

- This command is the same as SAMPle:COUNt and is included for SCPI compatibility.

SWEep:POINts?

[SENSe:]SWEep:POINts? [MIN | MAX] command returns the sweep points.

SWEep:OFFSet:POINts

[SENSe:]SWEep:OFFSet:POINts <*count*> | MIN | MAX command sets the number sweep offset points.

Parameters

<*count*> must be a negative number.

Note <*count*> must be a **negative** number.

Comments

- This command is the same as SAMPle:PRETrigger:COUNt (except the sign on <*count*> is negative here, whereas, it is positive for pretrigger count) and is included for SCPI compatibility.

SWEep:OFFSet:POINts?

[SENSe:]SWEep:OFFSet:POINts? [MIN | MAX] command returns the sweep offset points.

VOLTage[<*channel*>][:DC]:RANGe

[SENSe:]VOLTage[<channel>][:DC]:RANGe <*range*> command is used to change the range on the specified channel. There are seven different ranges. If the range specified falls between two of the instrument's ranges, the range is set to the next higher range setting. The command defaults to channel 1 if no channel is specified.

Parameters The crossover points for range changes are as follows:

Voltage Range	Resolution
0.0625	.000007629
0.2500	.000030518
1.0000	.000122070
4.0000	.000488281
16.0000	.007812500
64.0000	.007812500
256.0000	.03125

Comments

- Executable when initiated: No
- Coupled command: YES; TRIGger:LEVel may be affected if one of the levels is the trigger event on the channel that had the range change. The level set for CALCulate:LIMit :LOWer (and :UPPer) will be modified to be the same percent of full range. This will generate a different voltage value for the limit level.
- Reset condition: range is set to 256 for all channels

VOLTage[<*channel*>][:DC]:RANGe?

[SENSe:]VOLTage[<channel>][:DC]:RANGe? queries the specified channel for its present range setting. The command defaults to channel 1 if no channel is specified.

VOLTage[<*channel*>][:DC]:RESolution?

[SENSe:]VOLTage[<channel>][:DC]:RESolution? queries the specified channel for its present resolution setting. Resolution versus range setting is shown in the VOLTage[:DC]:RANGe command. The command defaults to channel 1 if no channel is specified.

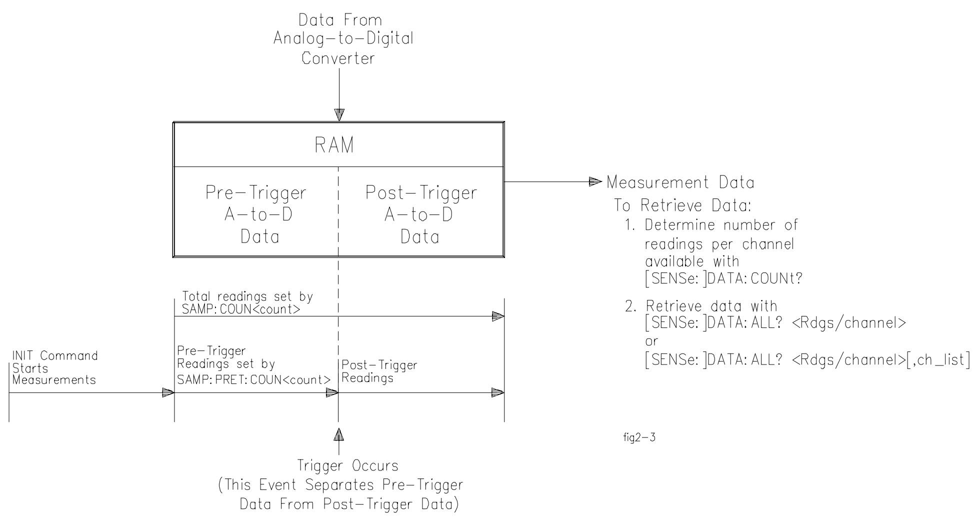
The SAMPle command subsystem sets the number of samples to be taken for each trigger. It also sets the number of samples to be taken prior to the trigger and the source of the sample signal and its slope. When the sample source is TIMer, you can also set the sample interval.

Subsystem Syntax

```
SAMPle
    :COUNt   <count> | MIN | MAX
    :COUNt?   [MIN | MAX]
    [:IMMediate]
    :PRETrigger:COUNt   <count> | MIN | MAX
    :PRETrigger:COUNt?   [MIN | MAX]
    :SLOPe   POS | 1 | NEG | 0
    :SLOPe?
    :SOURce   HOLD | TIMer | TTLT0-7 | EXT
    :SOURce?
    :TIMer   <interval> | MIN | MAX
    :TIMer?   [MIN | MAX]
```

:COUNt

SAMPle:COUNt <*count*> | MIN | MAX command sets the number of total samples which includes the pre-trigger and post-trigger samples. The number of samples set is common to all channels. You cannot have two or more channels with different sample settings.



Comments The total number of readings is limited to at most 16,777,216 for the 4-channel

E1564A Digitizer and 33,554,432 for the 2-channel E1563A Digitizer, depending on the amount of memory on the card. The following describes the limits with the different memory options:

Memory Size	HP E1563A (2-channel) Maximum Samples	HP E1564A (4-channel) Maximum Samples
4 MBytes	1,048,576	524,288
8 MBytes	2,096,152	1,048,576
16 MBytes	4,194,304	2,097,152
32 MBytes	8,388,608	4,194,304
64 MBytes	16,777,216	8,388,608
128 MBytes	33,554,432	16,777,216

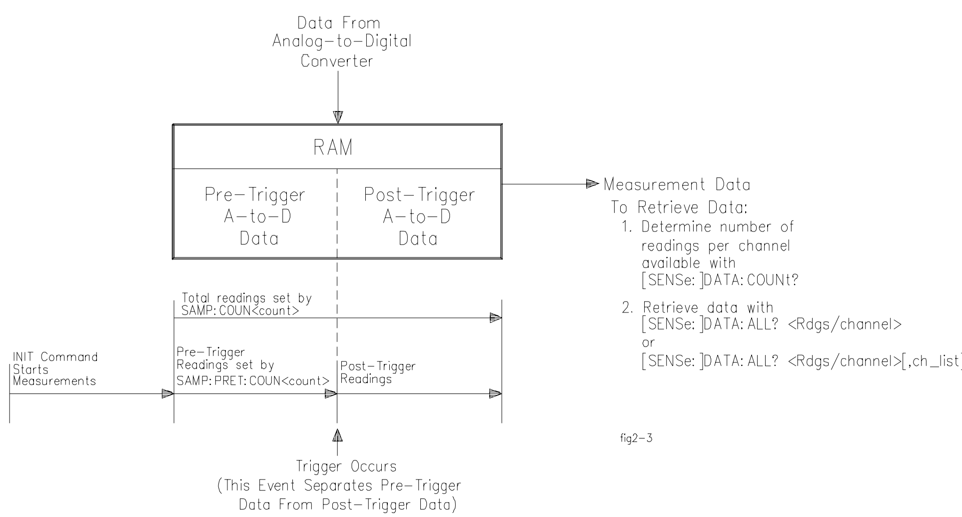
- One pre-trigger sample is required to get the above maximums. The maximum is one less if pre-trigger count is zero.
- Executable when initiated: No
- Coupled command: No
- Reset (*RST) condition: all channels set to 1 sample

:COUNt?

SAMPle:COUNt? [MIN | MAX] command returns the number of samples each channel will make. The number of samples returned is common to all channels.

:PRETrigger:COUNt

SAMPle:PRETrigger:COUNt <*count*> | MIN | MAX command sets the number of pretriggers (number of readings that will occur before the trigger event occurs). The count is common to all channels.



- <*count*> must be a positive number and not greater than the sample count -1.

This count specifies the portion of the total SAMPle:COUNt that will be sampled prior to the trigger.

- A trigger is ignored if it occurs *before* the pretrigger count is met.
- If the specified number of pretrigger samples (<*count*>) have been taken and a trigger has not yet occurred, the digitizer continues to sample the input signal. The digitizer retains the most recent pretrigger samples specified by the number "<*count*>" when the trigger does occur.
- Executable when initiated: No
- Coupled command: No
- Reset (*RST) condition: 0 pretriggers

:PRETrigger:COUNt?

SAMPle:PRETrigger:COUNt? [MIN | MAX] command returns the number of pretrigger samples each channel will make prior to each trigger. The number of pretriggers returned is common to all channels.

[:IMMediate]

SAMPle[:IMMediate] command is generally used only when the sample source is HOLD to take a single reading when the digitizer is in the wait-for-sample state.

:SLOPe

SAMPle:SLOPe POS | 1 | NEG | 0 command sets the slope of the sample signal (the active edge, rising or falling, of the sample signal). The slope setting is common to all channels.

Comments

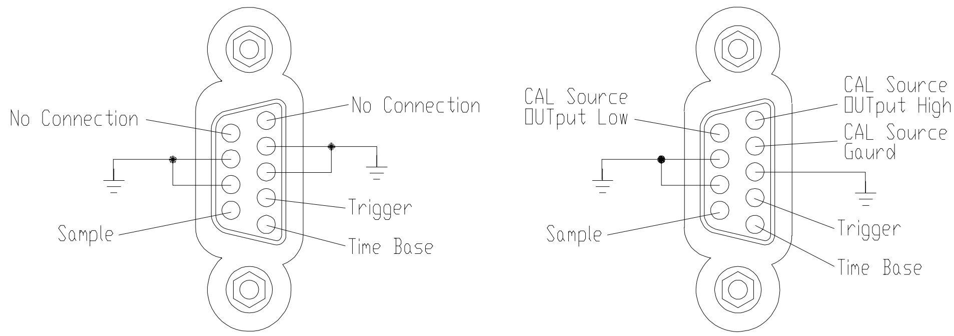
- This command is effective only when the sample source is EXTernal. The slope is set but will be ignored if the sample source is a source other than EXTernal.
- Executable when initiated: No
- Coupled command: No
- Reset (*RST) condition: POSitive (1)

:SLOPe?

SAMPle:SLOPe? command queries the present setting of the slope of the sample signal. The sample slope is effective only when the sample source is EXTernal.

SAMPle:SOURce HOLD | TIMer | TTLT0-7 | EXT command sets the source of the sample signal which causes a measurement to be made. The sample source is common to all channels. TIMer uses the internal time base. The EXTernal input is the TTL "Sample" input pin on the front panel D-subminiature connector (left pin column, bottom pin).

HP E1563A **HP E1564A**
("Sample" input - bottom left pin)



Parameters

Parameter Name	Parameter Type	Point of Source	Default Units
HOLD	discrete	SAMPle[:IMMediate] command	none
TIMer	discrete	uses specified SAMPle:TIMer *<interval>* as sample rate	none
TTLT0-7	discrete	VXIbus TTL trigger lines	none
EXTernal	discrete	"Sample" pin on front panel D-sub connector	none

Comments

- A rising or falling edge for the sample slope can be specified if the source is set to the EXTernal source, (see SAMPle:SLOPe command).
- A sampling period can be specified if the sample source is set to TIMer (see SAMPle:TIMer command).
- TRIG:MODE SLAVe<n> forces the sample source to be the appropriate TTL trigger line. Attempts to change the sample source while TRIG:MODE is SLAVe<n> will result in a settings conflict error message.
- Executable when initiated: No
- Coupled command:Yes; TRIG:MODE SLAVe<n> forces a certain TTL trigger line to the sample source. A settings conflict occurs if you attempt to change thid dedicated line with the

SAMPle;SOURce command. TTL sources may conflict with the output subsystem. Specifying a TTL source will force the output to be disabled. See OUTPut command.

- Reset (*RST) condition: TIMer source with 0.0000013 second sampling interval per reading.

:SOURce?

SAMPle:SOURce? command queries the present source setting for the sample signal. The returned string is either HOLD, TIMer, TTLT0-7 or EXT.

:TIMer

SAMPle:TIMer <*interval*> | MIN | MAX command sets the time interval for each sample event when the sample source is TIMer. Measurements are made on the input signal at this rate. This interval is common to all channels for sample source TIMer.

Parameter

Parameter Name	Parameter Type	Range of Values	Default Units
interval	numeric	1.25E-6 to 0.8 (in multiples of the reference oscillator period**; default TIMer period is 1.3E-7 seconds)	seconds

** See SENSe:ROSC:EXT:FREQ <freq> command

Comments

- The sample interval specified by the *period* parameter must be a multiple of the reference oscillator period. The specified time, if not a correct multiple of the reference oscillator period, will be rounded to the nearest value that can be attained. For SAMPle:SOURce INTernal, if not a correct multiple of 1E-7, it will be rounded to the nearest value that can be attained by the internal clock.

- Executable when initiated: No

- Coupled command: Yes, the value is changed to the nearest possible value if an external reference is specified.

- Reset (*RST) condition: 0.0000013 (1.3 µS)

:TIMer?

SAMPle:TIMer? [MIN | MAX] command queries the sample interval when the sample source is TIMer.

The STATus subsystem reports the bit values of the Operation Data/Signal Register and Questionable Data/Signal Register. It also allows you to unmask the bits you want reported from the Standard Event Register and to read the summary bits from the Status Byte Register.

The Operation Data/Signal Register and Questionable Data/Signal Register groups consist of a condition register, an event register and an enable register. The STATus :OPERation and :QUEStionable commands control and query these registers.

Subsystem Syntax

```
STATus
    :OPERation:CONDiton?
    :OPERation[:EVENt]?
    :OPERation:ENABle   <unmask>
    :OPERation:ENABle?
    :PRESet
    :QUEStionable:CONDition?
    :QUEStionable[:EVENt]?
    :QUEStionable:ENABle   <unmask>
    :QUEStionable:ENABle?
```

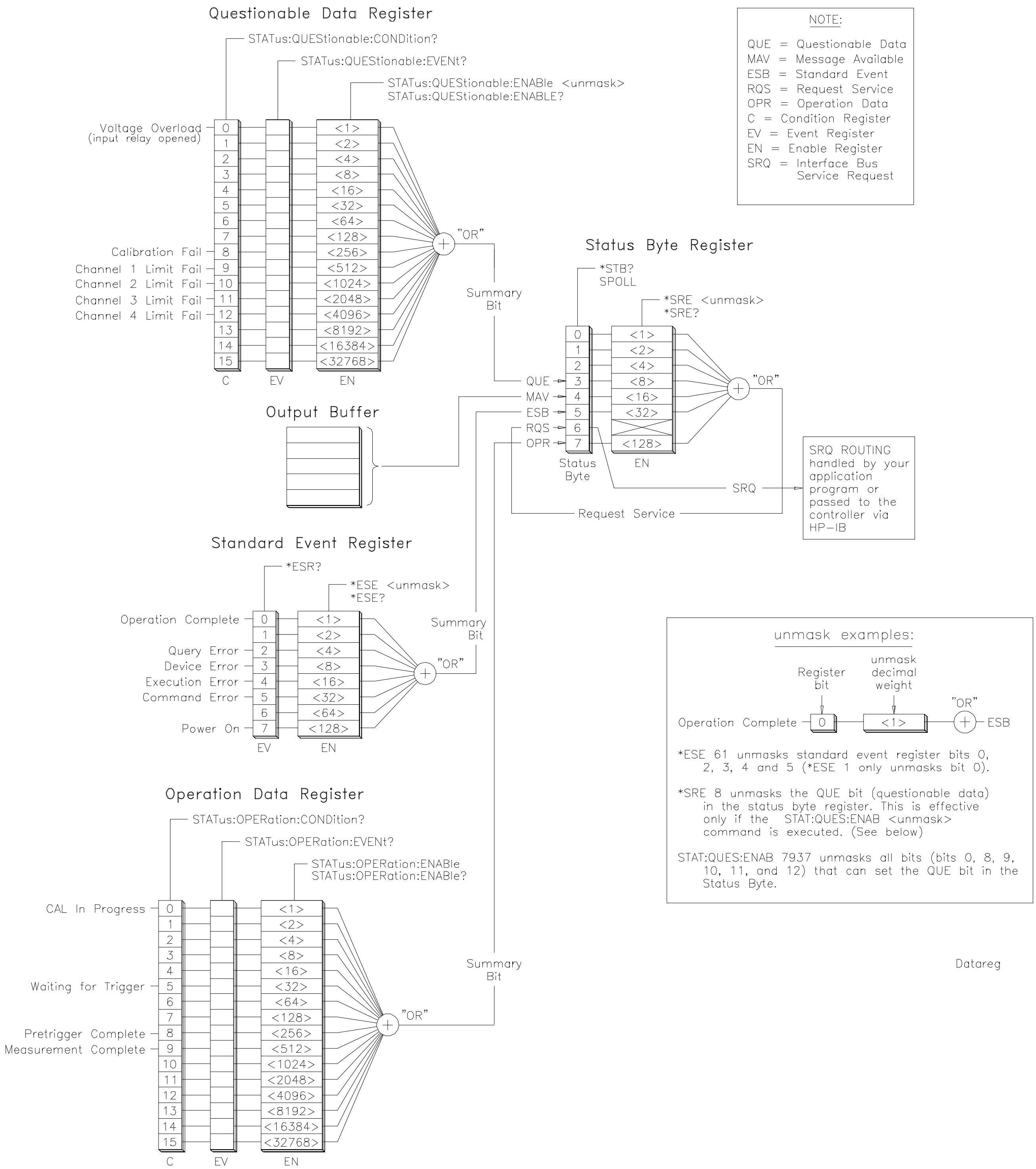
Status System Registers

The STATus system contains seven registers, four of which are under IEEE 488.2 control: the Standard Event Status Register (*ESR?), the Standard Event Enable Register (*ESE and *ESE?), the Status Byte Register (*STB?) and the Status Byte Enable Register (*SRE and *SRE?).

Questionable Data Register

The QUEStionable data register indicates failures as described in the following table. Limit failures occur at the sample rate so the condition register bits change rapidly and cannot be read until the measurement completes. You should read the EVENt register which latches the CONDition register once a measurement cycle to see if a limit failure occurred. You will then have to determine which reading failed by printing the reading number and the measurement value.

Bit #	Description
0	voltage overload
8	calibration failure
9	Chan 1 limit failure
10	Chan 2 limit failure
11	Chan 3 limit failure
12	Chan 4 limit failure


Questionable Data Register
STATus:QUEStionable:CONDition?
STATus:QUEStionable:EVENt?
STATus:QUEStionable:ENABle <unmask>
STATus:QUEStionable:ENABLE?
Voltage Overload (input relay opened)
Calibration Fail
Channel 1 Limit Fail
Channel 2 Limit Fail
Channel 3 Limit Fail
Channel 4 Limit Fail
"OR"
Summary Bit
C
EV
EN
NOTE:
QUE = Questionable Data
MAV = Message Available
ESB = Standard Event
RQS = Request Service
OPR = Operation Data
C = Condition Register
EV = Event Register
EN = Enable Register
SRQ = Interface Bus Service Request
Status Byte Register
*STB?
SPOLL
*SRE <unmask>
*SRE?
QUE
MAV
ESB
RQS
OPR
Status Byte
Output Buffer
SRQ
SRQ ROUTING handled by your application program or passed to the controller via HP-IB
Request Service
Standard Event Register
*ESR?
*ESE <unmask>
*ESE?
Operation Complete
Query Error
Device Error
Execution Error
Command Error
Power On
unmask examples:
Register bit
unmask decimal weight
Operation Complete
ESB
*ESE 61 unmasks standard event register bits 0, 2, 3, 4 and 5 (*ESE 1 only unmasks bit 0).
*SRE 8 unmasks the QUE bit (questionable data) in the status byte register. This is effective only if the STAT:QUES:ENAB <unmask> command is executed. (See below)
STAT:QUES:ENAB 7937 unmasks all bits (bits 0, 8, 9, 10, 11, and 12) that can set the QUE bit in the Status Byte.
Operation Data Register
STATus:OPERation:CONDition?
STATus:OPERation:EVENt?
STATus:OPERation:ENABle
STATus:OPERation:ENABle?
CAL In Progress
Waiting for Trigger
Pretrigger Complete
Measurement Complete
Datareg

Operation Data Register

The OPERation data register indicates operational status as follows:

Bit #	Description
0	CAL:STATe ON (calibration in progress)
5	waiting for trigger
8	pretrigger count is met
9	measurement complete

Status Byte Register

The OPR Operational Status bit, RQS Request Service bit, ESB Standard Event summary bit, MAV Message Available bit and QUE Questionable Data bit in the Status Byte Register (bits 7, 6, 5, 4 and 3 respectively) can be queried with the *STB? command but will be executed when previous commands are finished. Using the VISA I/O library, you can query the value of the status byte without going through the digitizer’s command parser by using the viReadSTB function call. The OPR bit is the summary bit for the Operation Data Register. The QUE bit is the summary bit for the Questionable Data Register.

Standard Event Register

Use the *ESE? command to query the "unmask" value for the Standard Event Status Register (bits you want logically OR'd into the summary bit). Query using decimal weighted bit values.

:OPERation:CONDition?

STATus:OPERation:CONDition? returns a decimal-weighted number representing the bits set in the Operation Data condition register.

:OPERation[:EVENt]?

STATus:OPERation[:EVENt]? returns a decimal-weighted number representing the bits set in the Operation Data/Signal Register's event register. This command clears all bits in the event register when executed.

:OPERation:ENABle

STATus:OPERation:ENABle <*unmask*> enables (unmasks) bits in the Operation Data/Signal Register's enable register to be reported to the summary bit (setting Status Byte Register bit 3 true). The event register bits are not reported in the Status Bytes Register unless specifically enabled.

:OPERation:ENABle?

STATus:OPERation:ENABle? returns a decimal-weighted number representing the bits enabled in the Operation Data/Signal Register's enable register signifying which bits will set OPR (bit 7) in the Status Byte.

:PRESet

STATus:PRESet command affects only the enable register by setting all enable register bits to 0. It does not affect either the "status byte" or the "standard event status". PRESet does not clear any of the event registers.

:QUEStionable :CONDition?

STATus:QUEStionable:CONDition? returns a decimal-weighted number representing the bits set in the Questionable Data condition register.

:QUEStionable [:EVENt]?

STATus:QUEStionable[:EVENt]? returns a decimal-weighted number representing the bits set in the Questionable Data/Signal Register's event register. This command clears all bits in the event register when executed.

:QUEStionable :ENABle

STATus:QUEStionable:ENABle <*unmask*> enables (unmasks) bits in the Questionable Data/Signal Register's enable register to be reported to the summary bit (setting Status Byte Register bit 3 true). The event register bits are not reported in the Status Bytes Register unless specifically enabled.

:QUEStionable :ENABle?

STATus:QUEStionable:ENABle? returns a decimal-weighted number representing the bits enabled in the Questionable Data/Signal Register's enable register signifying which bits will set QUE (bit 3) in the Status Byte.

The SYSTem command subsystem returns error numbers and their associated messages from the error queue. You can also query the SCPI version to which this instrument complies.

Subsystem Syntax

```
SYSTem
    :ERRor?
    :VERSion?
```

:ERRor?

SYSTem:ERRor? returns the error numbers and corresponding error messages in the error queue. See Appendix B in this manual for a listing of the error numbers, messages and descriptions.

Comments

- When an error is generated by the digitizer, it stores an error number and corresponding message in the error queue.
- One error is removed from the error queue each time the SYSTem:ERRor? command is executed. The errors are cleared in a first-in, first-out order. This means that if several errors are waiting in the queue, each SYSTem:ERRor? query returns the oldest (not the most recent) error. That error is then removed from the queue.
- When the error queue is empty, subsequent SYSTem:ERRor? queries return `+0,"No error"`. To clear all errors from the queue, execute the *CLS command.
- The error queue has a maximum capacity of 20 errors. If the queue overflows, the last error is replaced with `-350,"Too many errors"`. No additional errors are accepted by the queue until space becomes available.

Example Reading the Error Queue

SYST:ERR?	*Query the error queue*
enter statement	*Enter readings into computer.*

:VERSion?

SYSTem:VERSion? returns the SCPI version number this instrument complies.

Comments The information returned is in the format "YYYY.R" where "YYYY" is the year and "R" is the revision number within that year.

The TEST command subsystem allows you to run a particular self-test and returns information about self-test errors and results from the *TST? command.

Subsystem Syntax

```
TEST
    :ERRor? <test_number>
    :NUMBer? <test_number>,<cycles>
    :TST[:RESults]?
```

:ERRor?

TEST:ERRor? <*test_number*> returns a binary coded decimal (BCD) number and a string giving details about the error associated with the test number returned by the *TST? command or the array of errors returned by the TEST:TST[:RESults]? command. The string returns parameters of the test such as span, min, max and standard deviation. See the “Self-Test Error Definitions” section for details.

Parameter

Parameter Name	Parameter Type	Range of Values	Default Units
test_number	numeric	1 through 94	None

Comments

- The *TST? command returns only the first test that failed. Use the TEST:TST[:RESults]? command to obtain the complete list of all failures resulting from a *TST? command.
- The response may indicate, in detail, what caused the self-test error.
- See Appendix C, Error Messages, for retrieving information on self-test errors.

:NUMBer?

TEST:NUMBer? <*test_number*>,<*cycles*> allows you to cycle through a particular self-test a specified number of times instead of running the entire suite of self-tests as is performed with the *TST? command. This command is a query and returns the number of times the specified test failed out of the specified number of times the test was cycled. For example, send the command TEST:NUMB? 2,5 to cycle through test number “2” five times. A “5” is returned if all five test cycles fail.

Parameter

Parameter Name	Parameter Type	Range of Values	Default Units
test_number	numeric	1 through 94	None
cycles	numeric	1 through 32767	None

Comments

test_number	**Description**
1	General register read/write test
2	Cal constant/flash ROM read test
3	Channel 1: 62 mV range filter OFF, offset noise test
4	Channel 2: 62 mV range filter OFF, offset noise test
5*	Channel 3: 62 mV range filter OFF, offset noise test
6*	Channel 4: 62 mV range filter OFF, offset noise test
7	Channel 1: 62 mV range filter ON, offset noise test
8	Channel 2: 62 mV range filter ON, offset noise test
9*	Channel 3: 62 mV range filter ON, offset noise test
10*	Channel 4: 62 mV range filter ON, offset noise test
11	Channel 1: 0.25V range filter OFF, offset noise test
12	Channel 2: 0.25V range filter OFF, offset noise test
13*	Channel 3: 0.25V range filter OFF, offset noise test
14*	Channel 4: 0.25V range filter OFF, offset noise test
15	Channel 1: 0.25V range filter ON, offset noise test
16	Channel 2: 0.25V range filter ON, offset noise test
17*	Channel 3: 0.25V range filter ON, offset noise test
18*	Channel 4: 0.25V range filter ON, offset noise test
19	Channel 1: 1V range filter OFF, offset noise test
20	Channel 2: 1V range filter OFF, offset noise test
21*	Channel 3: 1V range filter OFF, offset noise test
22*	Channel 4: 1V range filter OFF, offset noise test
23	Channel 1: 1V range filter ON, offset noise test
24	Channel 2: 1V range filter ON, offset noise test
25*	Channel 3: 1V range filter ON, offset noise test

test_number	Description
26*	Channel 4: 1V range filter ON, offset noise test
27	Channel 1: 4V range filter OFF, offset noise test
28	Channel 2: 4V range filter OFF, offset noise test
29*	Channel 3: 4V range filter OFF, offset noise test
30*	Channel 4: 4V range filter OFF, offset noise test
31	Channel 1: 4V range filter ON, offset noise test
32	Channel 2: 4V range filter ON, offset noise test
33*	Channel 3: 4V range filter ON, offset noise test
34*	Channel 4: 4V range filter ON, offset noise test
35	Channel 1: 16V range filter OFF, offset noise test
36	Channel 2: 16V range filter OFF, offset noise test
37*	Channel 3: 16V range filter OFF, offset noise test
38*	Channel 4: 16V range filter OFF, offset noise test
39	Channel 1: 16V range filter ON, offset noise test
40	Channel 2: 16V range filter ON, offset noise test
41*	Channel 3: 16V range filter ON, offset noise test
42*	Channel 4: 16V range filter ON, offset noise test
43	Channel 1: 64V range filter OFF, offset noise test
44	Channel 2: 64V range filter OFF, offset noise test
45*	Channel 3: 64V range filter OFF, offset noise test
46*	Channel 4: 64V range filter OFF, offset noise test
47	Channel 1: 64V range filter ON, offset noise test
48	Channel 2: 64V range filter ON, offset noise test
49*	Channel 3: 64V range filter ON, offset noise test
50*	Channel 4: 64V range filter ON, offset noise test
51	Channel 1: 256V range filter OFF, offset noise test
52	Channel 2: 256V range filter OFF, offset noise test
53*	Channel 3: 256V range filter OFF, offset noise test
54*	Channel 4: 256V range filter OFF, offset noise test
55	Channel 1: 256V range filter ON, offset noise test
56	Channel 2: 256V range filter ON, offset noise test
57*	Channel 3: 256V range filter ON, offset noise test

test_number	Description
58*	Channel 4: 256V range filter ON, offset noise test
59*	Channel 1: Offset DAC test
60*	Channel 2: Offset DAC test
61*	Channel 3: Offset DAC test
62*	Channel 4: Offset DAC test
63*	Channel 1: Gain DAC test
64*	Channel 2: Gain DAC test
65*	Channel 3: Gain DAC test
66*	Channel 4: Gain DAC test
67*	Channel 1: 62 mV uncalibrated gain
68*	Channel 2: 62 mV uncalibrated gain
69*	Channel 3: 62 mV uncalibrated gain
70*	Channel 4: 62 mV uncalibrated gain
71*	Channel 1: 0.25V uncalibrated gain
72*	Channel 2: 0.25V uncalibrated gain
73*	Channel 3: 0.25V uncalibrated gain
74*	Channel 4: 0.25V uncalibrated gain
75*	Channel 1: 1V uncalibrated gain
76*	Channel 2: 1V uncalibrated gain
77*	Channel 3: 1V uncalibrated gain
78*	Channel 4: 1V uncalibrated gain
79*	Channel 1: 4V uncalibrated gain
80*	Channel 2: 4V uncalibrated gain
81*	Channel 3: 4V uncalibrated gain
82*	Channel 4: 4V uncalibrated gain
83*	Channel 1: 16V uncalibrated gain
84*	Channel 2: 16V uncalibrated gain
85*	Channel 3: 16V uncalibrated gain
86*	Channel 4: 16V uncalibrated gain
87*	Channel 1: 64V uncalibrated gain
88*	Channel 2: 64V uncalibrated gain
89*	Channel 3: 64V uncalibrated gain

test_number	Description
90*	Channel 4: 64V uncalibrated gain
91*	Channel 1: 256V uncalibrated gain
92*	Channel 2: 256V uncalibrated gain
93*	Channel 3: 256V uncalibrated gain
94*	Channel 4: 256V uncalibrated gain

* These tests require the HP E1564A 4-Channel Digitizer.

Self-Test Error Definitions

A failed self-test will return a number other than zero. The binary value of that number defines the failure mode. More than one failure mode may result from one self-test. The failure modes are defined in the following sections for each type of self-test. The bits and their weighting are shown below.

bit #	7	6	5	4	3	2	1	0
weight	128	64	32	16	8	4	2	1

Offset Noise Test (self-test numbers 3 - 58)

BCD weight	Failure mode
1	Span is zero
2	Span is too large
4	Mean is too low
8	Mean is too high
16	Standard deviation is too large

Offset DAC Test (self-test numbers 59-62)
(HP E1564A 4-Channel Digitizer)

BCD weight	Failure mode
1	DAC measurement is noisy
2	Measured data span is too small
4	Lower end point to upper end point span is too small
8	Lower end point to upper end point span is too large
16	Offset DAC span does not include 0
32	Bit weight is out of limits; the offending bit is in B15-B8.

Gain DAC Test (self-test numbers 63-66)
(HP E1564A 4-Channel Digitizer)

BCD weight	Failure mode
1	DAC measurement is noisy
2	Measured data span is too small
4	Lower end point to upper end point span is too small
8	Lower end point to upper end point span is too large
16	Gain DAC span does not include 0
32	Bit weight is out of limits; the offending bit is in B15-B8.
64	Gain DAC nominal setting is out of limits.

Uncalibrated Gain Test (self-test numbers 67-94)
(HP E1564A 4-Channel Digitizer)

BCD weight	Failure mode
1	The max-to-min span is 0.0.
2	Gain span is too large.
4	Gain mean is too low.
8	Gain mean is too high.
16	Gain standard deviation is too large.
32	Gain is out of limits.

:TST[:RESults]?

TEST:TST[:RESults]? returns an array of integers that result from the self-test command *TST?.

Comments

- A response of "0" indicates there is no error.
- Use the TEST:ERR? <*test_number*> command to retrieve details about the failed test number(s) returned by the TEST:TST:RESults? command.

The TRIGger command subsystem controls the behavior of the trigger system.

Subsystem Syntax

```
TRIGger
        [:IMMediate]
        :LEVel<channel>   <level> | MIN | MAX
        :LEVel<channel>?
        :MODE   NORMal | MASTer0,2,4,6 | SLAVe0,2,4,6
        :MODE?
        :SLOPe[<n>]   POS | 1 | NEG | 0
        :SLOPe[<n>]?
        :SOURce[<n>]   OFF | BUS | EXT | HOLD | IMMediate |
                       INTernal1-4 | TTLT0-7
        :SOURce[<n>]?
```

[:IMMediate]

TRIGger[:IMMediate] causes the instrument to transition to the wait-for-sample state immediately, regardless of the trigger source selected. The instrument must be initiated (INITiate command) and be in the wait-for-trigger state when TRIG:IMM is executed. A “Trigger ignored” error will be generated if the instrument has not been initiated prior to this command or if it is not in the wait-for-trigger state.

Comments

- Executable when initiated: Yes
- Coupled command: No
- Reset (*RST) condition: none

:LEVel

TRIGger:LEVel<channel>   <*voltage*> | MIN | MAX sets the level on the specified channel that can be used for *internally* triggering the instrument.

Note This command is valid only for TRIGger:SOURce INTernal1-4.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
level	numeric	see Comments	volts

Comments

- The present range setting will determine the maximum and minimum values that can be entered without error.

- Changing range will keep the level at the same percentage of the new range. For example, if level is set to 2.0 on the 4V range, the level is set to 8.0 if you change to the 16V range (50% of full range).

- Changing range will change an existing level to the same percent of full scale on the new range (e.g., an 8.0 level on the 16V range and then change range to 4V range changes the level to 2.0V; still 50%) and an error message to be generated. In this situation, the level is set to the maximum or minimum the new range will support.

- The TRIG:SLOPe command specifies which direction of signal movement through the level will trigger the digitizer. TRIG:SLOPe is POSitive causes a trigger when the signal passes through the level and rises above the specified level. A trigger occurs when TRIG:SLOPe is NEGative and the trigger signal passes through the level and falls below the specified level.

- Executable when initiated: No

- Coupled command: Yes, range setting

- Reset (*RST) condition: 0.00 on all channels

:LEVel?

TRIGger:LEVel<channel>? queries the value of the trigger level set on the specified channel.

:MODE

TRIGger:MODE NORMal | MASTer<n> | SLAVe<n> sets the trigger mode. Master and slave parameters set the modules for use in connecting more than one module together for simultaneous measurements from the same trigger and sample.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
n	numeric	0, 2, 4, 6	None

- NORMal sets standard trigger operation and the specified trigger and sample sources are used.

- MASTer<n> and SLAVe<n> pairs a sample line and a trigger line which are then used for multiple unit synchronization. See the section titled “Master-Slave Operation” in Chapter 2 for more information including diagrams.

		TTLT pairs to SLAVe modules	
MASTer MODE	**SLAVe MODE**	**Sample line**	**Trigger line**
MASTer0	SLAVe0	TTLT0	TTLT1
MASTer2	SLAVe2	TTLT2	TTLT3
MASTer4	SLAVe4	TTLT4	TTLT5
MASTer6	SLAVe6	TTLT6	TTLT7

Comments

- Executable when initiated: Yes
- Coupled command: No
- Reset (*RST) condition: NORMal mode

:MODE?

TRIGger:MODE? queries the trigger mode setting. Returns NORMal, MASTer or SLAVe.

:SLOPe[<n>]

TRIGger:SLOPe[<n>] POS | 1 | NEG | 0 sets the active edge of the trigger signal which causes a measurement to be made. There are two trigger sources and you must designate which source you are setting the slope. n = 1 for the slope of trigger source number 1. n = 2 for the slope of trigger source number 2. Trigger slope defaults to n=1 if “n” is not designated.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
n	numeric	1 or 2	None

Note Trigger slope is active only when the trigger source is one of the four INTernal levels (TRIG:SOURce INT1-4) or when the EXTernal trigger source is specified (TRIG:SOURce EXTernal).

Comments

- Executable when initiated: Yes
- Coupled command: TRIG:SOURce INT1-4 and TRIG:SOURce EXTernal

- Reset (*RST) condition: SLOPe1 = POSitive; SLOPe2 = POSitive

:SLOPe[<n>]?

TRIGger:SLOPe[<n>]? queries the present setting for the slope of the trigger signal for the trigger source (1 or 2) specified. Trigger slope for source number 1 is returned if "n" is not designated.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
n	numeric	1 or 2	None

Note Trigger slope applies only for TRIG:LEVel when the trigger source is INTernal or when EXTernal trigger sources are specified.

- Returns either "POS" or "NEG".

:SOURce[<n>]

TRIGger:SOURce[<n>] BUS | EXTernal | HOLD | IMMediate | INTernal1-4 | TTLT0-7 sets the source of the trigger for all channels or can disable the trigger source. The command defaults to trigger source number 1 if "n" is not designated.

Note Two trigger sources are allowed, TRIG:SOUR1 and TRIG:SOUR2, **which are common to <u>ALL</u> channels** on the E1563A and E1564A. SOUR1 <u>is not</u> associated only with channel 1 and SOUR2 <u>is not</u> associated only with channel 2.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
n	numeric	1 or 2	None

Comments

- The TRIGger:SOURce command only selects the trigger source. You must use the INITiate command to place the digitizer in the wait-for-trigger state.

- TRIGger:SOURce EXT uses the digitizer's front panel "Trig" pin in the D sub-miniature connector as the trigger source. The digitizer triggers on the falling (negative-going) edge of a ±5V TTL input signal (maximum input is +5V peak to the front panel D sub-miniature connector "Trig" pin).

- TRIGger:IMMediate causes a trigger to occur immediately provided the digitizer is placed in the wait-for-trigger state using the INITiate command.

- When a Group Execute Trigger (GET) bus command or *TRG common command is executed and the digitizer is not in the wait-for-trigger state, the "Trigger ignored" error is generated.
- TRIGger:SOURce INTernal1-2 (E1563A) or TRIGger:SOURce:INTernal1-4 (E1564A) triggers a reading when the level specified by the TRIG:LEVel<channel> command is met. The TRIG:SLOPe setting determines whether the trigger occurs when the signal rises above (POSitive) or falls below (NEGative) the specified level on that channel.

Note If TRIGger:SOURce INT<n> is set, CALCulate<n>:LIMit:LOWer[:STATe] or CALCulate<n>:LIMit:UPPer[:STATe] are disabled if they were enabled. <n> represents the channel number used for the internal trigger source and the channel used for testing a limit. Refer to the first block diagram in chapter 2 for information about how the internal trigger source is driven by the level signal.

- TRIG:SOURce1 is set to the appropriate TTLT<n> line by the TRIG:MODE MASTer | SLAVe command. TRIG:SOURce1 can not be changed unless the trigger mode is NORMal. Attempting to change TRIG:SOURce1 when mode is MASTer or SLAVe will cause a "settings conflict" error.
- TRIG:SOURce2 is not affected by TRIG:MODE MASTer | SLAVe operation.
- Executable when initiated: No
- Coupled command: Yes; TRIGger:LEVel, TRIGer:MODE, OUTPut:TTLT<n>:SOURce TRIG and CALC:LIMit:LOWer[:STATe] and CALC:LIMit:UPPer[:STATe]. Changes to TRIG:SOURce1 will cause a "settings conflict" error if TRIG:MODE is set to MASTer or SLAVe.
- Reset (*RST) condition: TRIGger:SOURce1 IMMediate
TRIGger:SOURce2 HOLD

TRIGger:SOURce[<n>]?

TRIGger:SOURce[<n>]? queries present setting for the specified trigger source (1 or 2). The command defaults to trigger source number 1 if "n" is not designated.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
n	numeric	1 or 2	None

Comments

- This command returns one of the following responses indicating the trigger source setting: BUS, EXT, HOLD, IMM, INT, INT2, INT3, INT4, TTLTn (where n = 0 to 7).

Note Internal level trigger on channel 1 is returned as INT versus INT1; the “1” is implied. The internal level trigger for channels 2, 3 and 4 return INT2, INT3 and INT4.

IEEE 488.2 Common Command Quick Reference

The table below lists, by functional group, the IEEE 488.2 Common (*) Commands that can be executed by the HP E1563A and HP E1564A Digitizers. However, commands are listed alphabetically in the following reference. Examples are shown in the reference when the command has parameters or returns a non-trivial response; otherwise, the command string is as shown in the table. For additional information, refer to IEEE Standard 488.2-1987.

Category	Command	Title	Description
System Data	*IDN	Identification	Returns the identification string of the Digitizer which includes the latest firmware version.
Internal Operations	*RST	Reset	Resets the Digitizer to: range: 256V input state: ON input filter: OFF TTLT states: OFF data format: ASCii See Table 2-1 in Chapter 2 for the module's complete reset state.
Internal Operations	*TST	Self-Test	Returns "0" if self-test passes. Returns a non-zero value if self-test fails. Use SYST:ERR? to retrieve the error from the Digitizer. See "Self-Test Errors" in Appendix B for a complete list of error numbers and their description. Return the digitizer to Hewlett-Packard for repair if repair is required.
Synchronization	*OPC *OPC? *WAI	Operation Complete Operation Complete Query Wait to Complete	Operation Complete Command Operation Complete Query Wait-to-Continue Command
Status & Event	*CLS *ESE <*unmask*> *ESE? *ESR? *SRE <*unmask*> *SRE? *STB?	Clear Status Event Status Enable Event Status Enable Query Event Status Register Query Service Request Enable Service Request Query Read Status Byte Query	Clear Status Command Standard Event Status Enable Cmd Standard Event Status Enable Query Standard Event Status Register Query Service Request Enable Command Service Request Enable Query Read Status Byte Query
Bus Operation	*TRG	Bus Trigger	When the digitizer is in the wait-for-trigger state and the trigger source is TRIGger:SOURce BUS, use *TRG to trigger the digitizer.

*CLS

***CLS** clears the Standard Event Status Register, the Operation Status Register, the Questionable Signal Register, and the error queue. This clears the corresponding summary bits (3, 5 and 7) in the Status Byte Register. *CLS does not affect the enable unmasks of any of the Status Registers.

Comments

- Executable when initiated: No
- Coupled command: No
- Related Commands: STATus:PRESet
- Reset (*RST) condition: None

*ESE and *ESE?

***ESE <*unmask*>** enables (unmasks) one or more event bits of the Standard Event Status Register to be reported in bit 5 (the Standard Event Status Summary Bit) of the Status Byte Register. *unmask* is the sum of the decimal weights of the bits to be enabled allowing these bits to pass through to the summary bit ESB (bit 5 in the status byte).

The query form returns the current enable unmask value.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
unmask	numeric	0 through 255	None

A 1 in a bit position enables the corresponding event; a 0 disables it.

Comments

- Executable when initiated: Yes
- Coupled command: No
- Related Commands: *ESR?, *SRE, *STB?
- Reset (*RST) condition: unaffected
- Power-On condition: no events are enabled

Example Enable all error events

*ESE 60	*Enable error events*

*ESR?

***ESR?** returns the value of the Standard Event Status Register. The register is then cleared (all bits 0).

Comments

- Executable when initiated: Yes
- Coupled command: No
- Reset (*RST) condition: none
- Power-On condition: register is cleared

*IDN?

***IDN?** returns identification information for the HP E1563A and E1564A Digitizers. The response consists of four fields:

```
HEWLETT-PACKARD, E1563A, 0, A.01.00
HEWLETT-PACKARD, E1564A, 0, A.01.00
```

Note The first two fields identify this instrument as model number HP E1563A (or HP E1564A) manufactured by Hewlett-Packard. The third field is 0 since the serial number of the digitizer is unknown to the firmware. The last field indicates the revision level of the firmware. The revision level shown above is an example and the actual response you receive will likely be different than the example.

Comments

- Executable when initiated: Yes
- Coupled command: No
- Reset (*RST) condition: none
- Power-On condition: register is cleared

*OPC

***OPC** causes the HP E1563A and E1564A Digitizers to wait for all pending operations to complete after which the Operation Complete bit (bit 0) in the Standard Event Status Register is set. The *OPC suspends any other activity on the bus until the digitizer completes all commands sent to it prior to the *OPC command.

Comments

- The INIT command is considered complete when the measurement is started. *OPC will not suspend activity once INIT is processed and measurements start but the instrument may not be finished taking all readings initiated.
- Executable when initiated: Yes

- Coupled command: No
- Related commands: *OPC?, *WAI
- Reset (*RST) condition: none

*OPC?

***OPC?** causes the HP E1563A and E1564A Digitizers to wait for all pending operations to complete. A single ASCII “1” is then placed in the output queue.

Comments

- The INIT command is considered complete when the measurement is started. *OPC? will return “1” once INIT is processed and measurements start but the instrument may not be finished taking all readings initiated.
- Executable when initiated: Yes
- Coupled command: No
- Related commands: *OPC?, *WAI
- Reset (*RST) condition: none

*RST

***RST** resets the HP E1563A and E1564A Digitizers as follows:

- Sets all commands to their *RST state.
- Aborts a calibration (CAL:STATe ON), resetting the CAL:STATe to OFF.
- Aborts all pending operations.

***RST does not affect:**

- The output queue
- The Service Request and Standard Event Status Enable Registers
- The enable unmasks for the Questionable Signal Registers
- Calibration data

Comments

- Executable when initiated: Yes
- Coupled command: No
- Reset (*RST) condition: none

*SRE and *SRE?

***SRE <*unmask*>** specifies which bits of the Status Byte Register are enabled (unmasked) to generate a IEEE-488.1 service request. Event and summary bits are always set and cleared in the Status Byte Register regardless of the unmask value. *unmask* is the sum of the decimal weights of the bits to be enabled allowing these bits to pass through to the summary bit RQS (bit 6 in the status byte).

The query form returns the current enable unmask value.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
unmask	numeric	0 through 255	None

A 1 in a bit position enables service request generation when the corresponding Status Byte Register bit is set; a “0” disables it.

Comments

- Executable when initiated: Yes
- Coupled command: No
- Reset (*RST) condition: unaffected
- Power-On condition: no bits are enabled

Example

Enable service request on Message Available bit

```
*SRE 16                         Enable request on MAV
```

*STB?

***STB?** returns the value of the Status Byte Register. The RQS bit (bit 6 in the status byte having decimal weight 64) is set if a service request is pending.

Comments

- Executable when initiated: Yes
- Coupled command: No
- Related commands: *SRE
- Reset (*RST) condition: none

*TST?

***TST?** causes the HP E1563A and E1564A Digitizers to execute its internal self-test and returns the number of the first failed test.

A zero response indicates that the self-test passed. Any non-zero response indicates that the test failed. Input the failed test number into the TEST:ERR? <*number*> command. The returned values from this will be the result code and a string. See Appendix B, Error Messages, for information on interpreting the result code and string.

Comments

- Executable when initiated: No
- Coupled command: No
- Reset (*RST) condition: none

*WAI

***WAI** causes the HP E1563A and E1564A Digitizers to wait for all pending operations to complete before executing any further commands.

Comments

- The *WAI will not wait for all measurements to complete when an INIT command is executed to start measurements. *WAI considers INIT finished once it is processed although the instrument may still be taking measurements. In this case, the instrument will move on to the next command following *WAI while measurements are being taken.
- Executable when initiated: Yes
- Coupled command: No
- Related commands: *OPC, *OPC?
- Reset (*RST) condition: none

SCPI Command Quick Reference

The following tables summarize SCPI commands for the HP E1563A and HP E1564A C-size Digitizers.

<table>
<tr><th>Command</th><th>Description</th></tr>
<tr><td>ABORt</td><td>Stops any measurement in progress and puts instrument in the idle state.</td></tr>
<tr><td>CALCulate[<channel>]
:LIMit:FAIL?

:LIMit:LOWer[:STATe] ON | 1 | OFF | 0
:LIMit:LOWer[:STATe]?
:LIMit:LOWer:DATA <*value*> | MIN | MAX
:LIMit:LOWer:DATA? [MIN | MAX]

:LIMit:UPPer[:STATe] ON | 1 | OFF | 0
:LIMit:UPPer[:STATe]?
:LIMit:UPPer:DATA <*value*> | MIN | MAX
:LIMit:UPPer:DATA? [MIN | MAX]</td><td>Defaults to channel 1 if none specified.
Checks for a limit failure.

Enable lower limit checking.

Set lower limit value.

Enable upper limit checking.

Set upper limit value.</td></tr>
<tr><td>CALibrate
:DAC:VOLTage <voltage> | MIN | MAX
:DAC:VOLTage? MIN | MAX

:DATA?

:GAIN[<channel>] [<readings>][,<rate>][,ON | 1 | OFF | 0]

:SOURce INTernal | EXTernal
:SOURce?

:STATe ON | 1 | OFF | 0
:STATe?

:STORe

:VALue <voltage>
:VALue?

:ZERO[<channel>] [<readings>][,<rate>]
:ZERO[<channel>]:ALL? [<readings>][,<rate>]</td><td>Calibration commands.
E1564A: sets internal cal source

Returns calibration constants.

Perform gain cal using :VAL <voltage>.

Set calibration source (INTernal is available only on E1564A).

Enable/disable ability to calibrate.

Store cal constants in NV memory.

Tell digitizer what cal value is input.

Perform zero cal on current range.
Perform zero cal on all ranges and return zero cal status response.</td></tr>
</table>

Command	Description
DIAGnostic	Troubleshooting commands.
:DAC:OFFSet[<channel>] <voltage>	Set offset voltage for the DAC.
:DAC:OFFSet[<channel>]:RAMP <count>	Output offset ramp from the DAC.
:DAC:GAIN[<channel>] <value>	Set DAC gain as specified.
:DAC:SOURce <voltage>	Output specified DAC voltage.
:DAC:SOURce:RAMP <count>	Output ramp from the DAC.
:INTerrupt:LINE 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7	Sets the interrupt line used (“0” =none).
:INTerrupt:LINE?	
:MEMory:SIZE <size>	Sets new value when you upsize RAM.
:MEMory:SIZE?	
:PEEK? <reg_num>	Query contents of a register.
:POKE <reg_num>,<data>	Write data to a register.
:SHORt[<channel>] ON | 1 | OFF | 0	Connect internal short to the channel.
:SHORt[<channel>]?	
:STATus?	Query interrupt sources register status.
FORMat	
[:DATA] ASCii | PACKed | REAL	Set data format.
[:DATA]?	
INITiate	
[:IMMediate]	Initiate a measurement now.
:CONTinuous ON | 1 | OFF | 0	Initiate measurements continuously.
:CONTinuous?	
INPut[<channel>]	Set the input filter and enable/disable.
:FILTer[:LPASs]:FREQ 1.5K | 6K | 25K | 100K (4-chan)	E1564A only; E1563A has fixed 25K.
:FILTer[:LPASs]:FREQ?	
:FILTer[:LPASs][:STATe] ON | 1 | OFF | 0	Enable/disable channel’s filter.
:FILTer[:LPASs][:STATe]?	
[:STATe] ON | 1 | OFF | 0	Enable/disable channel’s input.
[:STATe]?	
OUTPut	
:TTLT<n>:SOURce TRIGger | SAMPle | BOTH	Define TTLTrigger lines 0-7 to output trigger and/or sample signal.
:TTLT<n>:SOURce?	
:TTLT<n>[:STATe] ON | 1 | OFF | 0	Enable/disable the specified output.
:TTLT<n>[:STATe]?	

Command	Description
[SENSe:]	
DATA? <Rdgs_per_channel>[,<channel_list>]	Read data from list of channels.
DATA:ALL? <Rdgs_per_channel>	Read data from all channels.
DATA:COUNt? [MIN | MAX]	Query available readings per channel.
DATA:CVTable? <channel_list>	Query last reading taken from channel.
ROSCillator:EXTernal:FREQuency <*freq*> ROSCillator:EXTernal:FREQuency?	Declare external source's frequency.
ROSCillator:SOURce INTernal | EXTernal ROSCillator:SOURce?	Set reference oscillator source.
SWEep:POINts <neg_value> | MIN | MAX SWEep:POINts? [MIN | MAX]	Set number of sweep points.
SWEep:OFFSet:POINts <neg_value> | MIN | MAX SWEep:OFFSet:POINts? [MIN | MAX]	Set number of sweep offset points.
VOLTage[<channel>][:DC]:RANGe <range> | MIN | MAX VOLTage[<channel>][:DC]:RANGe? [MIN | MAX]	Set channel's voltage range.
VOLTage[<channel>][:DC]:RESolution? [MIN | MAX]	Query channel's resolution.
SAMPle [:STARt | :SEQuence[1]]	
:COUNt <count> | MIN | MAX :COUNt? [MIN | MAX]	Set the number of samples to take.
[:IMMediate]	Take a sample now.
:PRETrigger:COUNt <count> | MIN | MAX :PRETrigger:COUNt? [MIN | MAX]	Set the number of pretrigger samples.
:SLOPe POS | 1 | NEG | 0 :SLOPe?	Set the sample signal slope.
:SOURce HOLD | TIMer | TTLT0-7 | EXT :SOURce?	Set the sample source.
:TIMer <interval> | MIN | MAX :TIMer? [MIN | MAX]	Set sampling interval for source TIMer.

Command	Description
STATus	
:OPERation:CONDition?	Read OPER:CONDition register.
:OPERation[:EVENt]?	Read OPER:EVENt register.
:OPERation:ENABle <unmask>	Unmask operation register bits.
:OPERation:ENABle?	Read OPER:ENABle register.
:PRESet	PRESet the status registers.
:QUEStionable:CONDition?	Read OPER:CONDition register.
:QUEStionable[:EVENt]?	Read OPER:EVENt register.
:QUEStionableENABle <unmask>	Unmask questionable register bits.
:QUEStionable:ENABle?	Read OPER:EVENt register.
SYSTem	
:ERRor?	Read system errors from error queue.
:VERSion?	Query system version.
TEST	
:ERRor?	Return details about self-test errors.
:NUMBer? <*test_number*>	Run a specific self-test.
:TST[:RESults]?	Return results of the *TST command.
TRIGger	
[:STARt | :SEQuence[1]]	
[:IMMediate]	Trigger now.
:LEVel<channel> <voltage> | MIN | MAX	Set trigger level for internal trigger.
:LEVel<channel>? [MIN | MAX]	
:MODE NORMal | MASTer | SLAVe	Set trigger mode.
:MODE?	
:SLOPe<n> POS | 1 | NEG | 0	Set slope of trigger signal.
:SLOPe<n>?	
:SOURce<n> OFF | BUS | EXT | HOLD | IMMediate | INTernal1-4 | TTLT0-7	Set source of trigger signal.
:SOURce<n>?	

Appendix A
HP E1563A and E1564A Digitizer Specifications

Number of channels

E1563A:	2 channels
E1564A:	4 channels

Bandwidth:	>400 kHz for all ranges
Resolution:	14 bits (including sign)
Sample rates:	1 Sa/s to 800 kSa/s
Built-in DSP:	No
Alias protection:	Oversample
Time base resolution:	0.1 us
Low-frequency CMRR:	113 dB

Selectable input filters:

E1563A (per channel):	25 kHz
E1564A (per channel):	1.5kHz, 6 kHz, 25 kHz, 100 kHz

Trigger:	Time and Event
Pre-trigger capture:	Yes
Memory:	4 Mbyte to 128 Mbyte PC SIMM
FIFO memory	

Cooling/Slot:

Watts/slot

E1563A:	20.6W
E1564A:	37.4W
ΔP mm H2O:	0.18
Air flow liter/s:	2.8

HP E1563A/E1564A Accuracy Specifications

Range	Zero Offset[1] (with filter OFF)		Zero Offset[1] (with filter ON)		Gain (% of reading)		Noise (3 sigma)
	Specification[2]	Temperature Coefficient[3]	Specification[2]	Temperature Coefficient[3]	Specification[2]	Temperature Coefficient[3]	Specification
0.0625V	20 µV	1.9 µV/°C	28 µV	4.3 µV/°C	0.034%	0.0061%/°C	57 µV
0.25V	78 µV	6 µV/°C	110 µV	16 µV/°C	0.034%	0.0061%/°C	180 µV
1V	300 µV	15 µV/°C	430 µV	63 µV/°C	0.034%	0.0061%/°C	720 µV
4V	1.2 mV	60 µV/°C	1.7 mV	251 µV/°C	0.034%	0.0061%/°C	2.88 mV
16V	21 mV	1.3 mV/°C	21 mV	1.63 mV/°C	0.034%	0.0061%/°C	14.7 mV
64V	28 mV	1.65 mV/°C	34 mV	4.24 mV/°C	0.034%	0.0061%/°C	48 mV
256V	79 mV	4.28 mV/°C	110 mV	16.2 mV/°C	0.034%	0.0061%/°C	189 mV

[1] Valid within the range of 0°C to 55°C. A zero offset calibration for all channels must be performed if the instrument experiences a temperature <0°C or >55°C for these specifications to remain valid.
[2] Specification is valid when tested at a temperature within ±5°C of the calibration temperature.
[3] Amount of error that must be added for each °C outside of ±5°C of the calibration temperature.

Integral Non-linearity Specification

All ranges: 2.5 LSB

General Specifications

For indoor use, pollution degree 2

Operating altitude: 3000 meters or mainframe altitude specification, whichever is lower

Operating temperature: 0°C to 55°C

Relative humidity: up to 80% at 31°C, decreasing to 50% at 40°C

Appendix B
HP E1563A and E1564A Register-Based Programming

About This Appendix

The HP E1563A 2-Channel and HP E1564A 4-Channel Digitizers are register-based modules which do not support the VXIbus word serial protocol. When a SCPI command is sent to the digitizer, the instrument driver resident in the HP E1406A Command Module parses the command and programs the digitizer at the register level.

Register-based programming is a series of reads and writes directly to the digitizer registers. This increases throughput speed since it eliminates command parsing and allows the use of an embedded controller. Also, register programming provides an avenue for users to control a VXI module with an alternate VXI controller device and eliminate the need for using an HP E1405/E1406 Command Module.

This appendix contains the information you need for register-based programming. The contents include:

- Register Addressing 113
- Register Descriptions 116
- Program Timing and Execution 130
- Programming Example 130

Register Addressing

Register addresses for register-based devices are located in the upper 25% of VXI A16 address space. Every VXI device (up to 256 devices) is allocated a 64 byte block of addresses. Figure B-1 shows the register address location within A16 as it might be mapped by an embedded controller. Figure B-1 shows the location of A16 address space in the HP E1405A/B and E1406A Command Modules.

The Base Address

When you are reading or writing to a digitizer register, a hexadecimal or decimal register address is specified. This address consists of a base address plus a register offset.

The base address used in register-based programming depends on whether the A16 address space is outside or inside the HP E1406A Command Module.

A16 Address Space Outside the Command Module

When the HP E1406A Command Module is not part of your VXIbus system (Figure B-2), the digitizer's base address is computed as:[1]

$$C000_{16} + (LADDR * 64)_{16}$$

or

$$49{,}152 + (LADDR * 64)$$

where $C000_{16}$ (49,152) is the starting location of the register addresses, LADDR is the digitizer's logical address, and 64_{10} is the number of address bytes per VXI device. For example, the digitizer's factory set logical address is 40 (28_{16}). If this address is not changed, the digitizer will have a base address of:

$$C000_{16} + (40 * 64)_{16} = C000_{16} + A00_{16} = CA00_{16}$$

or (decimal)

$$49{,}152 + (40 * 64) = 49{,}152 + 2560 = 51{,}712$$

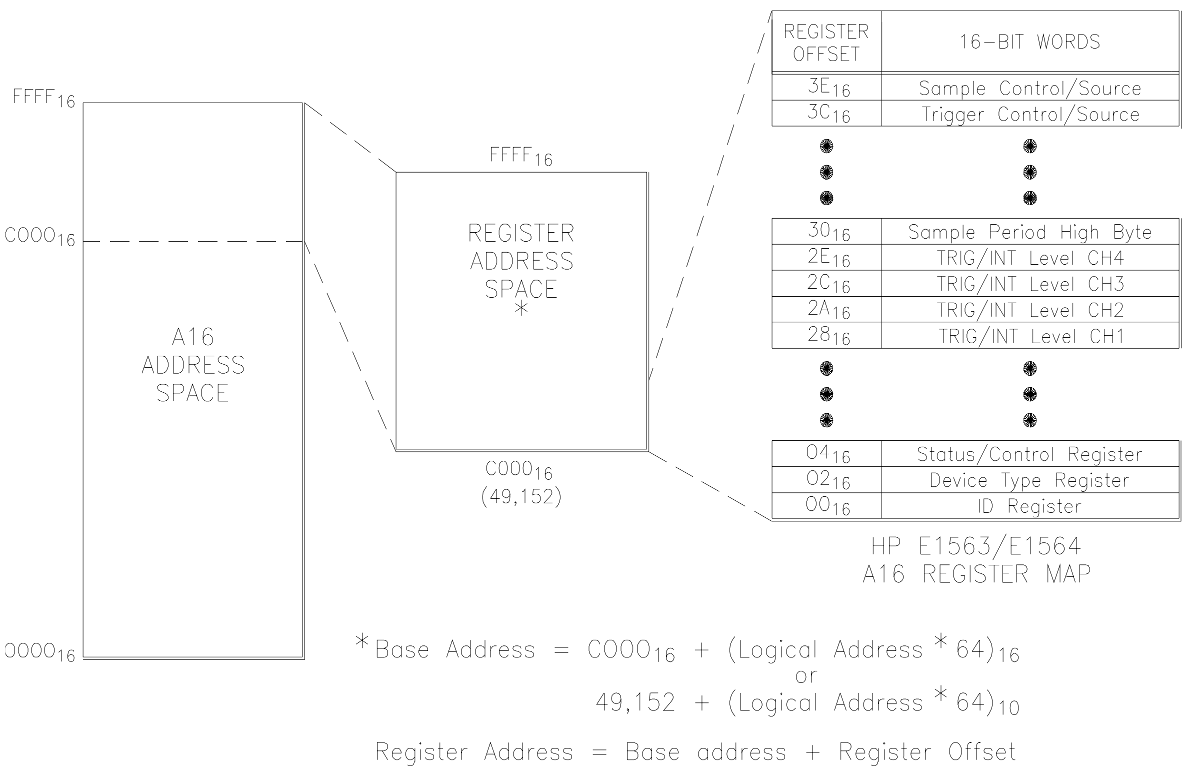


Figure B-1. Registers within A16 Address Space

1. The "16" at the end of the address indicates a hexadecimal number.

A16 Address Space Inside the Command Module or Mainframe

When the A16 address space is inside the HP E1406A Command Module (Figure B-2), the digitizer's base address is computed as:

$$1FC000_{16} + (LADDR * 64)_{16}$$

or

$$2,080,768 + (LADDR * 64)$$

where $1FC000_{16}$ (2,080,768) is the starting location of the VXI A16 addresses, LADDR is the digitizer's logical address, and 64 is the number of address bytes per register-based device. Again, the digitizer's factory set logical address is 40. If this address is not changed, the digitizer will have a base address of:

$$1FC000_{16} + (40 * 64)_{16} = 1FC000_{16} + A00_{16} = 1FCA00_{16}$$

or

$$2,080,768 + (40 * 64) = 2,080,768 + 2560 = 2,083,328$$

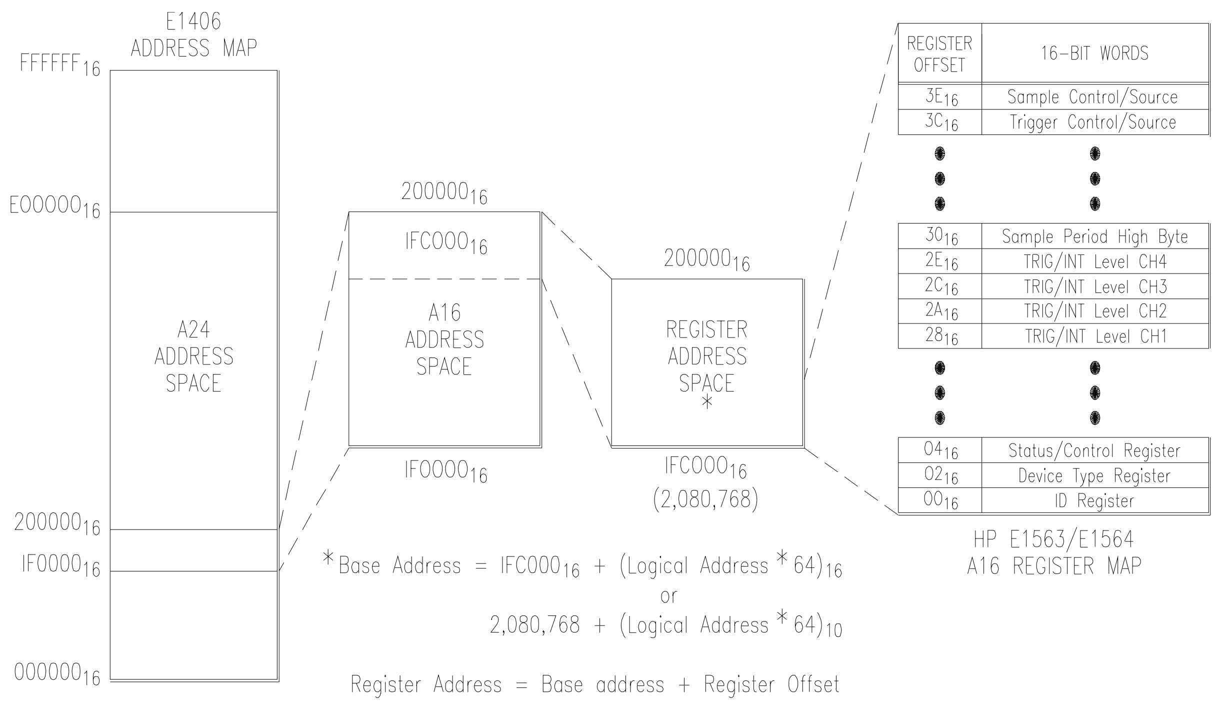


Figure B-2. Registers within Command Module's A16 Address Space

Register Offset

The register offset is the register's location in the block of 64 address bytes. For example, the multiplexer's Status/Control Register has an offset of 04_{16}. When you write a command to this register, the offset is added to the base address to form the register address:

$$CA00_{16} + 04_{16} = CA04_{16}$$

$$1FCA00_{16} + 04_{16} = 1FCA04_{16}$$

or

51,712 + 4 = 51,716

2,083,328 + 4 = 2,083,332

Register Descriptions

There are twenty WRITE and thirty-one READ registers on the digitizer. This section contains a description of the registers followed by a bit map of the registers in sequential address order. Undefined register bits appear as "0" when the register is read, and have no effect when written to.

WRITE Registers

You can write to the following digitizer registers:

-- Status/Control Register (base + 04 $_{16}$)
-- Offset Register (base + 06 $_{16}$)
-- Interrupt Control Register (base + 0C $_{16}$)
-- Calibration Flash ROM Address Register (base + 1C $_{16}$)
-- Calibration Flash ROM Data Register (base + 1E $_{16}$)
-- Calibration Source Register (base + 20 $_{16}$)
-- Range, Filter, Connect Channels 1 and 2 Register (base + 24 $_{16}$)
-- Range, Filter, Connect Channels 3 and 4 Register (base + 26 $_{16}$)
-- Trigger/Interrupt Level Channel 1 Register (base + 28 $_{16}$)
-- Trigger/Interrupt Level Channel 2 Register (base + 2A $_{16}$)
-- Trigger/Interrupt Level Channel 3 Register (base + 2C $_{16}$)
-- Trigger/Interrupt Level Channel 4 Register (base + 2E $_{16}$)
-- Sample Period High Word Register (base + 30 $_{16}$)
-- Sample Period Low Word Register (base + 32 $_{16}$)
-- Pre-Trigger Count High Register (base + 34 $_{16}$)
-- Pre-Trigger Count Low Register (base + 36 $_{16}$)
-- Post-Trigger Count High Register (base + 38 $_{16}$)
-- Post-Trigger Count Low Register (base + 3A $_{16}$)
-- Trigger Control/Source Register (base + 3C $_{16}$)
-- Sample Control/Source Register (base + 3E $_{16}$)

READ Registers

You can read the following digitizer registers:

-- Manufacturer ID Register (base + 00 $_{16}$)
-- Device Type Register (base + 02 $_{16}$)
-- Status/Control Register (base + 04 $_{16}$)
-- Offset Register (base + 06 $_{16}$)
-- FIFO High Word Register (base + 08 $_{16}$)
-- FIFO Low Word Register (base + 0A $_{16}$)
-- Interrupt Control Register (base + 0C $_{16}$)
-- Interrupt Sources Register (base + 0E $_{16}$)
-- CVTable Channel 1 Register (base + 10 $_{16}$)
-- CVTable Channel 2 Register (base + 12 $_{16}$)
-- CVTable Channel 3 Register (base + 14 $_{16}$)

-- CVTable Channel 4 Register (base + 16_{16})
-- Samples Taken High Word Register (base + 18_{16})
-- Samples Taken Low Word Register (base + $1A_{16}$)
-- Calibration Flash ROM Address Register (base + $1C_{16}$)
-- Calibration Flash ROM Data Register (base + $1E_{16}$)
-- Calibration Source Register (base + 20_{16})
-- Range, Filter, Connect Channels 1 and 2 Register (base + 24_{16})
-- Range, Filter, Connect Channels 3 and 4 Register (base + 26_{16})
-- Trigger/Interrupt Level Channel 1 Register (base + 28_{16})
-- Trigger/Interrupt Level Channel 2 Register (base + $2A_{16}$)
-- Trigger/Interrupt Level Channel 3 Register (base + $2C_{16}$)
-- Trigger/Interrupt Level Channel 4 Register (base + $2E_{16}$)
-- Sample Period High Word Register (base + 30_{16})
-- Sample Period Low Word Register (base + 32_{16})
-- Pre-Trigger Count High Register (base + 34_{16})
-- Pre-Trigger Count Low Register (base + 36_{16})
-- Post-Trigger Count High Register (base + 38_{16})
-- Post-Trigger Count Low Register (base + $3A_{16}$)
-- Trigger Control/Source Register (base + $3C_{16}$)
-- Sample Control/Source Register (base + $3E_{16}$)

ID Register

Reading the ID register returns FFF_{16} in the least significant bits to indicate the manufacturer is Hewlett-Packard and the module is an A16 register-based device.

base + 00_{16}	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	Device Class 1	1	Addr Space 0	0	Manufacturer ID - returns FFF_{16} (-12289_{10})in Hewlett-Packard A16 only register-based											

Reading the Register

- Via Command Module PEEK command: DIAG:PEEK? 2083328,16
  (2083328 = base with logical address 40 + 0 offset; see Figure B-2)

- Via Digitizer Module PEEK command: DIAG:PEEK? 0
  (0 signifies the first word, 16 bits, zero-base numbering system)

The “Programming Examples” on page 130 shows how to read the ID Register.

Device Type Register

Reading the Device Type Register returns 266_{16} in the least significant bits to identify the device as the HP E1563A 2-Channel Digitizer or 267_{16} in the least significant bits to identify the device as the HP E1564A 4-Channel Digitizer.

base + 02_{16}	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	0	1	1	1	HP E1563A (2-Channel Digitizer) = 266_{16}, 614_{10}; HP E1564A (4-Channel Digitizer) = 267_{16}, 615_{10}											

Reading the Register

- Via Command Module PEEK command: DIAG:PEEK? 2083330,16 (2083328 = base with logical address 40 + 02 offset; see Figure B-2)

- Via Digitizer Module PEEK command: DIAG:PEEK? 1 (1 signifies the second word, 16 bits, zero-base numbering system)

The “Programming Examples” on page 130 shows how to read the Device Type Register.

Status/Control Register

Writes to the Status/Control Register (base + 04_{16}) which enables you to reset the module and set either A24 or A32 decoding. You can also read the MODID bit.

base + 04_{16}	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Write*	A	Unde-fined	MOT-INTEL	A24	undefined				F	E	Undefined				S	R
Read**	A	M	MOT-INTEL	A24	Unde-fined	Memory Size			F	E	Arm Delay		RDY	P	S	R

Status/Control register bits defined:

*WRITE BITS (Status/Control Register)		
bit 0	R	Writing a "1" to this bit resets the digitizer to the power-on state. You must set bit 0 back to a logical "0" before resuming normal operations of the module.
bit 1	S	“1” inhibits sysfail, “0” does not inhibit sysfail.
bit 6	E	“1” disables error reporting LED, “0” enables error reporting LED (front panel).
bit 7	F	“1” disables Flash ROM “write”, “0” enables Flash ROM “write”.
bit 12	A24	“1” sets A24 space as all FIFO, “0” sets A24 space as broken up.
bit 13	MOT-INTEL	“1” sets Motorola format for reading ordering, “0” sets Intel format for reading ordering.
bit 15	A	“1” enables A32 decoding, “0” enables A24 decoding.

**READ BITS (Status/Control Register)		
bit 0	R	Reset Status; "1" = module reset, "0" = normal operation.
bit 1	S	SYSFAIL inhibit; “1” = inhibited, “0” = not inhibited.
bit 2	P	Passed; “1” = passed, “0” = failed.
bit 3	RDY	Ready; “1” = A32 decoding enabled, “0” = A24 decoding enabled.
bits 4 & 5	Arm Delay	Bit 4 is “1” for 1 mS after a range/filter change then returns to “0”, bit 5 is “1” for 30 mS after range/filter change then returns to “0”.
bit 6	E	Error; “1” disables front panel error LED, “0” enables front panel error LED.
bit 7	F	Flash ROM; “1” disables Flash ROM “write”, “0” enables Flash ROM “write”.
bits 8, 9, and 10	Memory Size	Memory Size; “000” = 4 MBytes, “001” = 8 MBytes, “010” = 16 MBytes, “011” = 32 MBytes, “100” = 64 MBytes, “101” = 128 MBytes.

**READ BITS (Status/Control Register)		
bit 12	A24	“1” sets A24 space as all FIFO, “0” sets A24 space as broken up.
bit 13	MOT-INTEL	“1” = Motorola big endian byte swapping, “0” = Intel little endian byte swapping.
bit 14	M	MODID bit; if the bit is "0", module has been selected.
bit 15	A	A24/A32 enable; “1” = A32 decoding enabled, “0” = A24 decoding enabled.

Reading the Register

- Via Command Module PEEK command: DIAG:PEEK? 2083332,16 (2083328 = base with logical address 40 + 04 offset; see Figure B-2)
- Via Digitizer Module PEEK command: DIAG:PEEK? 2 (2 signifies the third word, 16 bits, zero-base numbering system)

A24 Offset Register

The offset of the module in A24 space is set by the upper eight bits (15-8) of this register. The lower eight bits (7-0) of this register are zero.

base + 06_{16}	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Write*	A23	A22	A21	A20	A19	A18	A17	A16	undefined							
Read**	A24 Offset								0	0	0	0	0	0	0	0

A24 Offset register bits defined:

*WRITE BITS (A24 Offset Register)		
bits 8-15	A16-A23	These bits set the offset of the module in A24 space.

**READ BITS (A24 Offset Register)		
bits 8-15	A24 Offset	The module’s offset in A24 space.

Cache FIFO High Word Register

Data is stored on the module in large, slow dynamic RAM and fast, small backplane cache. Each of these data stores is a FIFO. The dynamic RAM FIFO receives the data from the ADC. As soon as the pre-trigger data has been identified, data is moved from the dynamic RAM FIFO to the backplane cache FIFO. Data is removed from the module using the cache FIFO. Data is 16-bit 2’s compliment and is packed into the FIFO registers. Always read register 08h before 0Ah if using D16. The FIFO is incremented after reading register 0Eh. If D32 is used, reading 08h will increment the FIFO correctly. The data is interwoven from all channels.

The following is the ordering of the data when D16 is used to remove the data on a 4-channel module.

- Read 08h channel 1 data (bit 15 is MSB of chan1, bit 0 is LSB of chan 1)
- Read 0Ah channel 2 data
- FIFO is automatically incremented to bring in the next data

- Read 08h channel 3 data
- Read 0Ah channel 4 data
- FIFO is automatically incremented to bring in the next data

The following is the ordering of the data when D32 is used to remove the data on a 4-channel module.

- Read 08h channel 1 data, channel 2 data (bit 31 is MSB of chan 1, bit 16 is LSB of chan 1, bit 15 is MSB of chan 2, bit 0 is LSB of chan 2)
- FIFO is automatically incremented to bring in the next data
- Read 0Ah channel 3 data, channel 4 data (bit 31 is MSB of chan 3, bit 16 is LSB of chan 3, bit 15 is MSB of chan 4, bit 0 is LSB of chan 4)
- FIFO is automatically incremented to bring in the next data

base + 08_{16}	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read																LSB

Cache FIFO Low Word Register

base + $0A_{16}$	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read																LSB

Interrupt Control Register

The interrupt level and the interrupt source are controlled by the interrupt control register. There are several sources of interrupt. A logical OR is performed on the enabled sources to determine if an IRQ should be pulled. This allows a user to set an interrupt if any channel exceeds a predetermined level or if data is available. Bits 0, 1 and 2 control the interrupt level (1 - 7). Level 0 (000) is not a valid setting. The enable bit (bit 3) allows an IRQ to occur when it is set high. All interrupt sources are edge sensitive. If a masked latched interrupt source is high during the interrupt acknowledge (iack) cycle, the latch of the source is cleared and will not be set until another edge from the source occurs.

base + $0C_{16}$	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Write*													Enable	L2	L1	L0
Read**	TRIG	DONE	PRE	OVER	CH4	CH3	CH2	CH1	undefined				Enable	Interrupt Level		

Interrupt Control register bits defined:

*WRITE BITS (Interrupt Control Register)		
bits 0-2	L0-2	Specifies the interrupt level (1 - 7); “001” = 1, “111” = 7
bit 3	Enable	Enable the interrupt; “1” = interrupt enabled, “0” = interrupt disabled.

**READ BITS (Interrupt Control Register)		
bit 15	TRIG	A trigger has been received after pre-trigger acquisition is done.
bit 14	DONE	Memory is full or post trigger acquisition is done.
bit 13	PRE	Pre-trigger data has been acquired.
bit 12	OVER	A dangerous OVERvoltage caused the channel input relay to open.
bit 11	CH4	Channel 4 exceeded the set limit.
bit 10	CH3	Channel 3 exceeded the set limit.
bit 9	CH2	Channel 2 exceeded the set limit.
bit 8	CH1	Channel 1 exceeded the set limit.

Interrupt Source Register

Eight events can be enabled to interrupt the digitizer. These events are listed in the above Interrupt Control Register definition for bits 8 through 15. The Interrupt Source Register contains the latched version (bits 8-15) and the unlatched version (bits 0-7) of these sources. The value of a source is latched high when the source has a low-to-high transition. The latched bits are cleared if they are masked as an interrupt source or by reading the register and writing back the contents. Writing a "1" to the bit clears the latch. The non-latched state of the interrupts is available all the time. The bit ordering of the latched bits and the unlatched bits is the same as the mask.

base + $0E_{16}$	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	TRIG	DONE	PRE	OVER	CH4	CH3	CH2	CH1	TRIG	DONE	PRE	OVER	CH4	CH3	CH2	CH1

Interrupt Source register bits defined:

READ BITS (Interrupt Source Register)		
bit 15, 7	TRIG	A trigger has been received after pre-trigger acquisition is complete and measurement count is not complete.
bit 14, 6	DONE	Memory is full or post trigger acquisition is complete.
bit 13, 5	PRE	Pre-trigger data has been acquired and waiting for trigger.
bit 12, 4	OVER	A dangerous OVERvoltage caused the channel input relay to open.
bit 11, 3	CH4	Channel 4 exceeded the set limit during the last sample taken.
bit 10, 2	CH3	Channel 3 exceeded the set limit during the last sample taken.
bit 9, 1	CH2	Channel 2 exceeded the set limit during the last sample taken.
bit 8, 0	CH1	Channel 1 exceeded the set limit during the last sample taken.

CVTable Channel 1 Register

This register holds the last value of the 2's compliment data stored in FIFO for channel 1. Data is 14 bits with the LSB at bit 2.

base + 10_{16}	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	MSB													LSB	0	0

CVTable Channel 2 Register

This register holds the last value of the 2's compliment data stored in FIFO for channel 2. Data is 14 bits with the LSB at bit 2.

base + 12_{16}	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	MSB													LSB	0	0

CVTable Channel 3 Register

This register holds the last value of the 2's compliment data stored in FIFO for channel 3. Data is 14 bits with the LSB at bit 2.

base + 14_{16}	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	MSB													LSB	0	0

CVTable Channel 4 Register

This register holds the last value of the 2's compliment data stored in FIFO for channel 4. Data is 14 bits with the LSB at bit 2.

base + 16_{16}	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	MSB													LSB	0	0

Samples Taken High Byte Register

This register holds the upper 16 bits of the number of samples taken (number of readings). The value in this register will continuously change as readings are taken.

base + 18_{16}	15 (31)	14 (30)	13 (29)	12 (28)	11 (27)	10 (26)	9 (25)	8 (24)	7 (23)	6 (22)	5 (21)	4 (20)	3 (19)	2 (18)	1 (17)	0 (16)
Read	0	0	0	0	0	0	0	0	0	0	x	x	x	x	x	x

Samples Taken Low Word Register

This register holds the lower 16 bits of the number of samples taken (number of readings). The value in this register will continuously change as readings are taken.

base + $1A_{16}$	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	LSB

Calibration Flash ROM Address Register

This register holds the address of the calibration flash ROM that is used for storing the calibration constants. Note the bit pattern 01010 for bits 15-11 in the upper byte. A write to Flash ROM is aborted if this pattern is not present.

base + $1C_{16}$	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Write*	0	1	0	1	0	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
Read**	0	0	0	0	0	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0

Calibration Flash ROM Data Register

This register holds the data of the calibration flash ROM that is used for the calibration constants. The upper eight bits return “0” when this register is read. Note the bit pattern 01010 for bits 15-11 in the upper byte. A write to Flash ROM is aborted if this pattern is not present.

base + $1E_{16}$	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Write*	0	1	0	1	0	1	0	1	D7	D6	D5	D4	D3	D2	D1	D0
Read**	0	0	0	0	0	0	0	0	D7	D6	D5	D4	D3	D2	D1	D0

Calibration Source Register

The E1564A 4-Channel Digitizer has an on-board calibration source. The source is a 12-bit DAC with a gain switch. Bit 15 is the gain switch and bits 11 through 0 are the calibration value.

base + 20_{16}	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Write	RANGE		MUX1	MUX0	DAC Data											

Calibration Source register bits defined:

*WRITE BITS (Calibration Source Register)		
bits 0-11	DAC	DAC data
bit 12, 13	MUX0, 1	connects choices to the output; 00 = CAL source, 01 = Raw DAC output, 10 = Internal +5V reference, 11 = Input short
bit 15	RANGE	DAC output ranges: 0 = ±15V DAC output, 1 = ±0.5V DAC output

Cache Count Register

The total number of samples taken by the digitizer is the ((cache count x 2) divided by the number of channels) + the sample count (registers at offset 18_{16} and $1A_{16}$).

base + 22_{16}	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Write	0	0	0	0	0	0	0	0	Cache count							

Range, Filter, and Channel 1, 2 Connect Register

Each channel has an 8-bit byte which controls the input signal range, filter cut off and the relay that connects the channel to the front panel connector. The fastest way to change range, filter or the connect relay is to write a 32-bit word to the register. After every write to this register the bus is held off 10 µs until the range, filter and relay information is sent to the isolated channel. The settling time for the relays, filters and the gain amplifier is about 20 ms. This register controls channels 1 and 2.

base + 24_{16}	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Write	Conn1	CH 1 Filter Code			short1	CH 1 Gain Code			Conn2	CH 2 Filter Code			short2	CH 2 Gain Code		
Read	Conn1	CH 1 Filter Code			short1	CH 1 Gain Code			Conn2	CH 2 Filter Code			short2	CH 2 Gain Code		

Range, Filter and Channel Connect register bits defined:

WRITE/READ BITS (Range, Filter and Channel 1/2 Connect Register)		
bits 0-2 and 8-10	Gain Code	These bits set the gain of the input channel by the codes shown below: 000 = 62.5 mV range 001 = 0.25V range 010 = 1.0V range 011 = 4.0V range 100 = 16V range 101 = 64V range 110 = 256V range (also 111 = 256V range)
bits 3 and 11	short1, short2	These bits connect an internal short to the channel inputs when the bit is “1”. When it is “0”, bits 7 & 15 connect the channel to the input or the calibration bus.
bits 4-6 and 12-14	Filter Code	These bits set the input channel filter cut-off frequency by the codes shown below: 000 = 1.5 kHz 001 = 6 kHz 010 = 25 kHz 011 = 100 kHz 111 = NO filter
bits 7 and 15	Connect Code	This bit connects the input channel to the front panel connector (Connect Code = 0) or to the calibration bus (Connect Code = 1).

Range, Filter, and Channel 3, 4 Connect Register

Each channel has an 8-bit byte which controls the input signal range, filter cut off and the relay that connects the channel to the front panel connector. The fastest way to change range, filter or the connect relay is to write a 32-bit word to the register. After every write to this register the bus is held off 10 µs until the range, filter and relay information is sent to the isolated channel. The settling time for the relays, filters and the gain amplifier is about 10 ms. This register controls channels 3 and 4.

base + 26_{16}	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Write	Conn3	CH 3 Filter Code			short3	CH 3 Gain Code			Conn4	CH 4 Filter Code			short4	CH 4 Gain Code		
Read	Conn3	CH 3 Filter Code			short3	CH 3 Gain Code			Conn4	CH 4 Filter Code			short4	CH 4 Gain Code		

Range, Filter and Channel Connect register bits defined:

WRITE/READ BITS (Range, Filter and Channel 3/4 Connect Register)		
bits 0-2 and 8-10	Gain Code	These bits set the gain of the input channel by the codes shown below: 000 = 62.5 mV range 001 = 0.25V range 010 = 1.0V range 011 = 4.0V range 100 = 16V range 101 = 64V range 110 = 256V range (also 111 = 256V range)
bits 3 and 11	short3, short4	These bits connect an internal short to the channel inputs when the bit is “1”. When it is “0”, bits 7 & 15 connect the channel to the input or the calibration bus.
bits 4-6 and 12-14	Filter Code	These bits set the input channel filter cut-off frequency by the codes shown below: 000 = 1.5 kHz 001 = 6 kHz 010 = 25 kHz 011 = 100 kHz 111 = NO filter
bits 7 and 15	Connect Code	This bit connects the input channel to the front panel connector (Connect Code = 0) or to the calibration bus (Connect Code = 1).

Trigger/Interrupt Level Channel 1 Register

This register provides 8-bit data corrected for offset and gain in 2’s compliment format.

base + 28_{16}	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Write*	MSB-D7	D6	D5	D4	D3	D2	D1	D0	0	0	0	0	0	0	0	GL
Read**	MSB-D7	D6	D5	D4	D3	D2	D1	D0	0	0	0	0	0	0	0	GL

Trigger/Interrupt Level Channel 1 register bits defined:

*WRITE/**READ BITS (Trigger/Interrupt Level Channel 1 Register)		
bit 0	GL	Greater than or Less than; “0” = >, “1” = <.
bits 15-8	D7-D0	data bits.

Trigger/Interrupt Level Channel 2 Register

This register provides 8-bit data corrected for offset and gain in 2’s compliment format.

Trigger/Interrupt Level Channel 2 register and bit definition:

base + $2A_{16}$	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Write*	MSB-D7	D6	D5	D4	D3	D2	D1	D0	0	0	0	0	0	0	0	GL
Read**	MSB-D7	D6	D5	D4	D3	D2	D1	D0	0	0	0	0	0	0	0	GL

*WRITE/**READ BITS (Trigger/Interrupt Level Channel 2 Register)		
bit 0	GL	Greater than or Less than; “0” = >, “1” = <.
bits 15-8	D7-D0	data bits.

Trigger/Interrupt Level Channel 3 Register

This register provides 8-bit data corrected for offset and gain in 2’s compliment format.

base + $2C_{16}$	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Write*	MSB-D7	D6	D5	D4	D3	D2	D1	D0	0	0	0	0	0	0	0	GL
Read**	MSB-D7	D6	D5	D4	D3	D2	D1	D0	0	0	0	0	0	0	0	GL

Trigger/Interrupt Level Channel 3 register bits defined:

*WRITE/**READ BITS (Trigger/Interrupt Level Channel 3 Register)		
bit 0	GL	Greater than or Less than; “0” = >, “1” = <.
bits 15-8	D7-D0	data bits.

Trigger/Interrupt Level Channel 4 Register

This register provides 8-bit data corrected for offset and gain in 2’s compliment format.

base + $2E_{16}$	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Write*	MSB-D7	D6	D5	D4	D3	D2	D1	D0	0	0	0	0	0	0	0	GL
Read**	MSB-D7	D6	D5	D4	D3	D2	D1	D0	0	0	0	0	0	0	0	GL

Trigger/Interrupt Level Channel 4 register bits defined:

*WRITE/**READ BITS (Trigger/Interrupt Level Channel 4 Register)		
bit 0	GL	Greater than or Less than; “0” = >, “1” = <.
bits 15-8	D7-D0	data bits.

Sample Period High Byte Register

This register provides the high byte of the sample period.

base + 30_{16}	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Write*									x	x	x	x	x	x	x	x
Read**	0	0	0	0	0	0	0	0	x	x	x	x	x	x	x	x

Sample Period Low Word Register

This register provides the low word (2 bytes) of the sample period.

base + 32_{16}	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Write*	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	LSB
Read**	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	LSB

Pre-Trigger Count High Byte Register

Pre-trigger count is the number of readings stored before the trigger is received. The minimum value is 0. The maximum number of readings is the size of memory in bytes divided by 8 for the E1563 and divided by 4 for the E1564.

base + 34_{16}	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Write	undefined										C5	C4	C3	C2	C1	C0
Read	0	0	0	0	0	0	0	0	0	0	C5	C4	C3	C2	C1	C0

Pre-Trigger Count Low Word Register

This register holds the low word (2 bytes) for the pre-trigger count.

base + 36_{16}	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Write	C15	C14	C13	C12	C11	C10	C9	C8	C7	C6	C5	C4	C3	C2	C1	C0
Read	C15	C14	C13	C12	C11	C10	C9	C8	C7	C6	C5	C4	C3	C2	C1	C0

Sample Count High Byte Register

Sample count is the total number of readings to be taken including the pre-trigger readings. The minimum value is 1. Zero (0) causes continuous readings and will not stop the acquisition until all of memory is full. The module will not stop acquiring data if the host can remove readings fast enough. The maximum number of readings is the size of memory in bytes divided by 8 for the E1563 and divided by 4 for the E1564.

base + 38_{16}	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Write	undefined										C5	C4	C3	C2	C1	C0
Read	0	0	0	0	0	0	0	0	0	0	C5	C4	C3	C2	C1	C0

Sample Count Low Word Register

Register containing the low word (2 bytes) for the sample count. This register and the high byte in register 38_{16} hold a value that can be set by the SAMPle:COUNt command in Chapter 3. Chapter 2, Figure 2-3 shows the relationship of the sample count and the pre-trigger count.

base + $3A_{16}$	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Write	C15	C14	C13	C12	C11	C10	C9	C8	C7	C6	C5	C4	C3	C2	C1	C0
Read	C15	C14	C13	C12	C11	C10	C9	C8	C7	C6	C5	C4	C3	C2	C1	C0

Trigger Source/ Control Register

This register provides the bits that control the trigger system.

base + $3C_{16}$	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Write*	CMP4	CMP3	CMP2	CMP1	SLAVING PAIR		EX_ TRIG	POS_ NEG	SOFT TRIG	MAS- TER	SLAVE	EN_ TTL	IN/ OUT	TTL_3	TTL_1	TTL_0
Read**	CMP4	CMP3	CMP2	CMP1	SLAVING PAIR		EX_ TRIG	POS_ NEG	SOFT TRIG	MAS- TER	SLAVE	EN_ TTL	IN/ OUT	TTL_3	TTL_1	TTL_0

Trigger Source/Control register bits defined:

***WRITE BITS (Trigger Source Register) and **READ BITS (Trigger Control Register)**		
bits 0-2	TTL_n	000 = TTLT0, 001 = TTLT1, 010 = TTLT2, ... , 011 = TTLT6, 111 = TTLT7.
bit 3	IN/OUT	TTLTn line is: 0 = IN, 1 = OUT.
bit 4	EN_TTL	0 = disable TTLTn, 1 = enable TTLTn
bit 5	SLAVE	0 = not a slave module, 1 = slave module.
bit 6	MASTER	0 = not a master module, 1 = master module.
bit 7	SOFT TRIG	software trigger: 0 = IMMediate disabled, 1 = IMMediate enabled.
bit 8	POS_NEG	trigger slope: 0 = NEG, 1 = POS.
bit 9	EX_TRIG	0 = EXTernal trigger disabled, 1 = EXTernal trigger enabled and must be input on the “Trig” pin on the front panel D-subminiature connector.
bits 10-11	SLAVING_PAIR	00 = MASTer0/SLAVe0; 01 = MASTer2/SLAVe2; 10 = MASTer4/SLAVe4; 11 = MASTer6/SLAVe6
bits 12-15	CMP1-4	0 = INTn disabled, 1 = INTn enabled; Example: a “1” in CMP2 means the level set in the Trigger/Interrupt Level Channel 2 Register will be used as the INTernal trigger source.

NOTE See the section titled “Master-Slave Operation” at the end of this appendix for more information on register programming the digitizer in a master-slave configuration. This uses bits 5, 6, 10 and 11 of the above register.

Sample Source/ Control Register

This register provides the bits that control the sample system.

base + $3E_{16}$	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Write*	SW ARM IMM		SW ARM 30 mS delay	ABORT			EX_ Sample	POS_ NEG	SOFT SAMPLE	EXT TIME-BASE	INT CLOCK	EN_TTL	IN/OUT	TTL_3	TTL_1	TTL_0
Read**	SW ARM IMM		SW ARM 30 mS delay	ABORT			EX_ Sample	POS_ NEG	SOFT SAMPLE	EXT TIME-BASE	INT CLOCK	EN_TTL	IN/OUT	TTL_3	TTL_1	TTL_0

Sample Source/Control register bits defined:

*WRITE BITS (Sample Source Register) and **READ BITS (Sample Control Register)		
bits 0-2	TTL_n	000 = TTLT0, 001 = TTLT1, 010 = TTLT2, ... , 011 = TTLT6, 111 = TTLT7.
bit 3	IN/OUT	TTLTn line is: 0 = IN, 1 = OUT.
bit 4	EN_TTL	1 = enable TTLTn, 0 = disable TTLTn
bit 5	INT Clock	0 = disable sampling from internal clock source, 1 = sample from the internal clock source.
bit 6	EXT Timebase	0 = timebase is internal 10 MHz clock, 1 = timebase is external clock source you must input on the “Time Base” pin on the front panel D-subminiature connector.
bit 7	Soft Sample	software sample: 0 = IMMediate disabled, 1 = IMMediate enabled.
bit 8	POS_ NEG	External sample slope: 0 = NEG, 1 = POS.
bit 9	EX_ Sample	1 = EXTernal sample is an external source you must input on the “Sample” pin on the front panel D-subminiature connector. 0 = EXTernal sample disabled.
bit 12	ABORT	1 = aborts measurement and flushes all reading data in all memory. The bit is set to “0” when the digitizer is initiated.
bit 13*	INIT with 30 mS delay	This bit will initiate measurements after a 30 mS delay when it is set to “1”. It is set to “0” when pre-trigger readings are complete.
bit 15*	INIT IMM	This bit will initiate measurements immediately when it is set to “1”. It is set to “0” when pre-trigger readings are complete.

If bit 12 and either bit 13 or 15 is set during the same write, an ABORT followed by an INIT is executed.

If bit 12 is “0”, either bit 13 or 15 is set and the previous measurement completed, an ABORT followed by an INIT is executed.

If bits 12, 13 and 15 are all “0”, no action is initiated.

Program Timing and Execution

This section contains generalized flowcharts and comments for performing these and other procedures. The flowcharts identify the registers used and the status bits monitored to ensure execution of the program.

System Configuration

The following C language example programs were developed on an HP RADI-EPC7 VXI embedded computer using Microsoft® Visual C++™ programming language and using HP VISA I/O calls. You can also use an HP Vectra (IBM PC compatible) computer connected via HP-IB to the HP E1406A slot 0 Command Module. The command module simply provides direct access to the VXI backplane.

If you use the HP E1406A with SCPI commands, you would use the HPE1563A/E1564A SCPI driver which you installed in the HP E1406A firmware and register programming would not be necessary. Chapter 3 describes the SCPI commands for the digitizer's driver.

Programming Example

The example program in this section demonstrates how to direct register program the digitizer. The following example program contains segments that:

1. Read the ID and Device Type Registers.
2. Read the Status Register.
3. Make digitizer measurements.
4. Retrieve the last readings from each channel's CVT register.
5. Retrieve all the readings from the two cache registers.
6. Reset the module.

Beginning of Program

```
/* This program resets the E1563A/E1564A, reads the ID Register, the Device */
/* Type Register, the Status Register, makes measurements and retrieves data*/
/* Programmed with MS Visual C++ version 2.0 using HP VISA I/O calls. */

#include <visa.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <time.h>

/* function prototypes */
void err_handler();
void wait (int wait_seconds);
void reset(ViSession vi, ViStatus x);
```

Program Main

```
void main(void)
{
   unsigned short id_reg, dt_reg;/* ID and Device Type */
   unsigned short stat_reg; /* Status Register */
   unsigned short cvt_reg, cache_reg; /* last value and cache registers */
   double last_reading, reading; /* decimal values of readings */
   int i;

   /* create and open a device session */
   ViStatus err;
   ViSession defaultRM, digitizer;
   ViOpenDefaultRM(&defaultRM);
   /* HP-IB interface address is 9 */
   /* digitizer logical address switch = 40 (factory setting) */
   ViOpen(defaultRM,"GPIB-VXI0::9::40",VI_NULL,VI_NULL,&digitizer);

   /* reset the E1563A/E1564A */
   reset(digitizer, err);
```

Read ID and Device Type Registers

```
/********* read the digitizer's ID and Device Type registers *********/

    err=ViIn16(digitizer,VI_A16_SPACE,0x00,&id_reg);/* read reg 00 */
        if (err<VI_SUCCESS) err_handler(digitizer,err);
    err=ViIn16(digitizer,VI_A16_SPACE,0x02,&dt_reg);/* read reg 02 */
        if (err<VI_SUCCESS) err_handler(digitizer,err);

   printf("ID register = 0x%4X\n",id_reg)
   printf("Device Type register = 0x%4X\n",dt_reg);
```

Read Status Register

```
/**************** read the digitizer's status register **************/

    err=ViIn16(digitizer,VI_A16_SPACE,0x04,&stat_reg);/* read status reg */
        if (err<VI_SUCCESS) err_handler(digitizer,err);
    printf("Status register = 0x%4X\n", stat_reg);
```

Make some measurements and retrieve readings

```
/******************** make measurements *******************/
/* set channel 1 and 2 to 4V range */
    err=ViOut16(digitizer,VI_A16_SPACE,0x24,0x7373);/* 0x7373 sets 4V range */
        if (err<VI_SUCCESS) err_handler(digitizer,err);
/* set channel 3 and 4 to 4V range */
    err=ViOut16(digitizer,VI_A16_SPACE,0x26,0x7373);/* 0x7373 sets 4V range */
        if (err<VI_SUCCESS) err_handler(digitizer,err);

/* set pre-trigger count of 4 */
    err=ViOut16(digitizer,VI_A16_SPACE,0x34,0x0);     /* high word = 0 */
        if (err<VI_SUCCESS) err_handler(digitizer,err);
    err=ViOut16(digitizer,VI_A16_SPACE,0x36,0x4);     /* low word = 4 */
        if (err<VI_SUCCESS) err_handler(digitizer,err);

/* set sample count of 7 */
    err=ViOut16(digitizer,VI_A16_SPACE,0x38,0x0);     /* high word = 0 */
        if (err<VI_SUCCESS) err_handler(digitizer,err);
    err=ViOut16(digitizer,VI_A16_SPACE,0x3A,0x7);     /* low word = 7 */
        if (err<VI_SUCCESS) err_handler(digitizer,err);

/* set trigger source */
    err=ViOut16(digitizer,VI_A16_SPACE,0x3C,0x180);   /* set bits 7 and 8 */
        if (err<VI_SUCCESS) err_handler(digitizer,err);

/* initiate a reading with a 30 mS delay */
    err=ViOut16(digitizer,VI_A16_SPACE,0x3E,0x21A0);  /* set bits 5,7,8 & 13 */
        if (err<VI_SUCCESS) err_handler(digitizer,err);


/******************** retrieve readings *******************/
/* read the CVT registers */
    err=ViIn16(digitizer,VI_A16_SPACE,0x10,&cvt_reg);
        if (err<VI_SUCCESS) err_handler(digitizer,err);
    printf(“channel 1 = 0x%4X\n”, cvt_reg);
    last_reading = (double)cvt_reg*4/32768;
    printf(“channel 1 = %lf Volts\n”, last_reading);

    err=ViIn16(digitizer,VI_A16_SPACE,0x12,&cvt_reg);
        if (err<VI_SUCCESS) err_handler(digitizer,err);
    printf(“channel 2 = 0x%4X\n”, cvt_reg);
    last_reading = (double)cvt_reg*4/32768;
    printf(“channel 2 = %lf Volts\n”, last_reading);

/* E1564A only for channels 3 and 4 -------------- */
    err=ViIn16(digitizer,VI_A16_SPACE,0x14,&cvt_reg);
        if (err<VI_SUCCESS) err_handler(digitizer,err);
    printf(“channel 3 = 0x%4X\n”, cvt_reg);
    last_reading = (double)cvt_reg*4/32768;
    printf(“channel 3 = %lf Volts\n”, last_reading);

    err=ViIn16(digitizer,VI_A16_SPACE,0x16,&cvt_reg);
        if (err<VI_SUCCESS) err_handler(digitizer,err);
    printf(“channel 4 = 0x%4X\n”, cvt_reg);
    last_reading = (double)cvt_reg*4/32768;
    printf(“channel 4 = %lf Volts\n”, last_reading);
```

```
/* read all 7 readings from all channels */
/* comment the channel 3/4 lines out if running the 2-channel E1563A */

for (i=0; i<7; ++i)
{
    err=ViIn16(digitizer,VI_A16_SPACE,0x08,&cache_reg);
        if (err<VI_SUCCESS) err_handler(digitizer,err);
    reading = (double)cache_reg*4/32768;
    printf("channel 1 = %lf Volts\n", reading);

    err=ViIn16(digitizer,VI_A16_SPACE,0x0A,&cache_reg);
        if (err<VI_SUCCESS) err_handler(digitizer,err);
    reading = (double)cache_reg*4/32768;
    printf("channel 2 = %lf Volts\n", reading);

/* E1564A only for channels 3 and 4 -- comment out for E1563A */
    err=ViIn16(digitizer,VI_A16_SPACE,0x08,&cache_reg);
        if (err<VI_SUCCESS) err_handler(digitizer,err);
    reading = (double)cache_reg*4/32768;
    printf("channel 3 = %lf Volts\n", reading);

    err=ViIn16(digitizer,VI_A16_SPACE,0x0A,&cache_reg);
        if (err<VI_SUCCESS) err_handler(digitizer,err);
    reading = (double)cache_reg*4/32768;
    printf("channel 4 = %lf Volts\n", reading);
}  /* end of if statement */

    /* reset the digitizer */
   reset(digitizer,err);
   printf("\nHP E1563A/E1564A is reset");

/***** Close session *****/
   ViClose(digitizer);
   ViClose(defaultRM);
}
```

Reset Function

```
/*************************************************************/
void reset(ViSession digitizer, ViStatus err)
   /* reset the digitizer (write a 1 to status bit 0) delay 1 second for reset */
   /* then set bit back to 0 to allow module to operate */
{
    /* write a "1" to the reset bit then set the bit back to "0" */
    err=ViOut16(digitizer,VI_A16_SPACE,0x04,1);     /* set reset bit to "1" */
        if (err<VI_SUCCESS) err_handler(digitizer,err);
    wait(1);
    err=ViOut16(digitizer,VI_A16_SPACE,0x04,0);     /* set reset bit to "0" */
        if (err<VI_SUCCESS) err_handler(digitizer,err);

   return;
}
```

Wait Function

```
/**************************************************************/
void wait(int wait_seconds)  /* wait for specified period in seconds */
{
time_t current_time;
time_t entry_time;
fflush(stdout);
if(-1==time(&entry_time))
   {  printf("Call failed, exiting...\n");
      exit(1);
   }
do
   { if(-1==time(&current_time))
      {  printf("Call failed, exiting...\n");
         exit(1);
      }
   }
while((current_time-entry_time)<((time_t)wait_seconds));
fflush(stdout);
}   /* end of wait function */
```

Program Output

Printout from example program:

```
ID register = 0xCFFF
Device Type register = 0x7267
Status register = 0x40CE

last readings printout

all readings from all channels printout

HP E1563A/E1564A is reset
```

Master-Slave Operation

The HP E1563A and HP E1564A Digitizers can be configured in a master-slave configuration. This configuration allows a master module and one or more slave modules to have their measurements synchronized. Synchronization occurs by having all channels trigger off of the same trigger event as well as all channels sampling from one sample signal.

- The sample synchronization signal is always generated by the master.
- The TTL trigger event can be generated by either the master module or any of the slave modules. This allows a slave module (as well as the master module) to use one of the four internal trigger sources or their external trigger source to trigger a measurement.

Both the trigger signal and the sample signal are put on the VXI backplane TTL trigger (TTLT) lines where the master module and all slave modules receive the signals simultaneously. TTL trigger lines are used in pairs between the master and slave(s) where one TTL trigger line carries the sample signal and the other carries the trigger signal. The next section describes how these TTL trigger lines are paired.

Trigger Mode

Bits 5 and 6 of the Trigger Source/Control register are used to configure Digitizers for master-slave operation. The module functions as a normal module when both bits are "0".

NORMal Mode

The default setting for trigger mode is normal (bits 5 and 6 of the Trigger Source/Control register are both "0") which configures the module as an individual instrument.

MASTer Mode

Bit 6 of the Trigger Source/Control register is used to configure a module as a master. The eight TTL trigger lines (TTLT0-TTLT7) on the VXI backplane allow four different pairings as shown in Table 2-1 (MASTer0-SLAVe0, MASTer2-SLAVe2, MASTer4-SLAVe4 and MASTer6-SLAVe6). You must select an unused set of TTL trigger lines for the master-slave coupling when determining which master mode to set. Do not use a TTLT line already used as a sample or trigger source.

SLAVe Mode

Bit 5 of the Trigger Source/Control register is used to configure a module as a slave to a MASTer0 module. MASTer0 and SLAVe0 modules share TTL trigger lines TTLT0 and TTLT1. TTLT0 carries the sample signal and TTLT1 carries the trigger signal. The following table shows all pairs of TTL trigger lines for each master-slave mode.

Table 3-2. Master-Slave Modes.

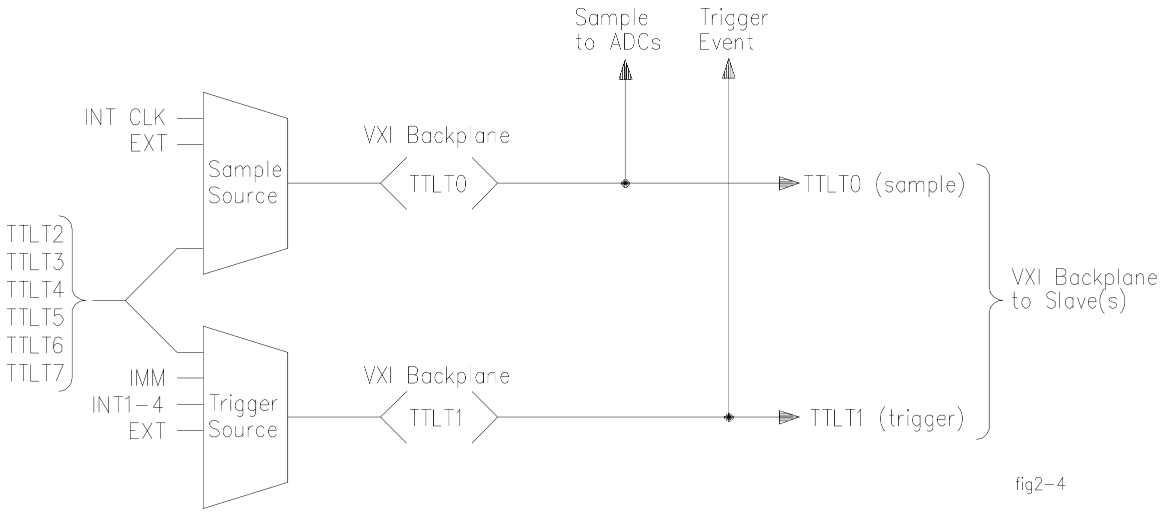
MASTer MODE	SLAVe MODE	Sample line	Trigger line
MASTer0	SLAVe0	TTLT0	TTLT1
MASTer2	SLAVe2	TTLT2	TTLT3
MASTer4	SLAVe4	TTLT4	TTLT5
MASTer6	SLAVe6	TTLT6	TTLT7

Master-Slave Diagrams

Figure B-1 illustrates a module configured as a master module.
Figure B-2 illustrates a module configured as a slave module.

Figure B-1. Master Module Configuration.

TRIG:MODE MASTer0 pairs TTLT0 (sample) with TTLT1 (trigger) The MASTer0 module will function with all SLAVe0 modules.



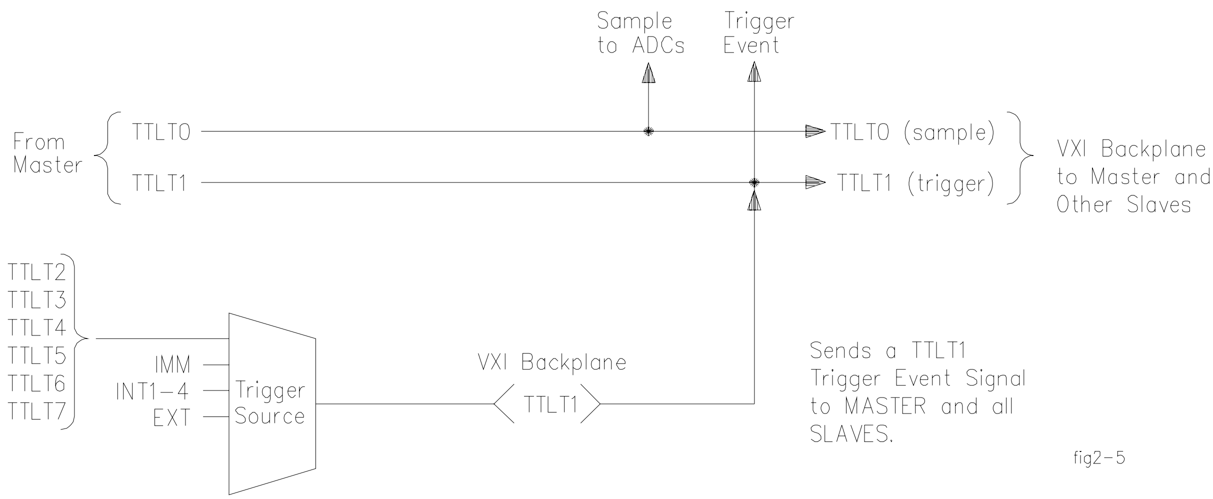
1) The trigger source from the slave can be set with the Trigger Source/Control Register bits 0,1,2,3,4,7,8,9,12,13,14, and 15. .

MODE	Sample signal	Trigger signal
SLAVe0	TTLT0	TTLT1
SLAVe2	TTLT2	TTLT3
SLAVe4	TTLT4	TTLT5
SLAVe6	TTLT6	TTLT7

2) SLAVe0 sets the TTLT0 line as if the sample source is TTLT0 and sets the TTLT1 line as if the trigger source is TTLT1. These lines are simply dedicated for synchronization between the modules in the master-slave mode. You should not use these lines for any other purpose.

Figure B-2. Slave Module Configuration.

TRIG:MODE   SLAVe0 pairs TTLT0 (sample) with TTLT1 (trigger) A SLAVe0 module will function with other SLAVe0 modules and the MASTer0 module.



1) The trigger source from the slave can be set with the Trigger Source/Control Register bits 0,1,2,3,4,7,8,9,12,13,14, and 15. .

MODE	Sample signal	Trigger signal
SLAVe0	TTLT0	TTLT1
SLAVe2	TTLT2	TTLT3
SLAVe4	TTLT4	TTLT5
SLAVe6	TTLT6	TTLT7

2) SLAVe0 sets the TTLT0 line as if the sample source is TTLT0 and sets the TTLT1 line as if the trigger source is TTLT1. These lines are simply dedicated for synchronization between the modules in the master-slave mode. You should not use these lines for any other purpose.

Appendix C
HP E1563A and E1564A Digitizer Error Messages

The following sections describe the types of errors the HP E1563A and HP E1564A report; Execution Errors, Self-Test Errors and Calibration Errors. The error code is given (e.g., -101) followed by the associated error message and a description of what the error message means.

Execution Errors

-101 **Invalid character**
An invalid character was found in the command string. You may have inserted a character such as #, $, or % in the command header or within a parameter. Example: `INP:FILT:FREQ#6E3`

-102 **Syntax error**
Invalid syntax was found in the command string. You may have inserted a blank space before or after a colon in the command header, or before a comma. Example: `SAMP:COUN    ,1`

-103 **Invalid separator**
An invalid separator was found in the command string. You may have used a comma instead of a colon, semicolon, or blank space, you may have used a comma where none was required – or you may have used a blank space instead of a comma. Example: `TRIG:LEV,1` or `DATA? 400  1`

-104 **Data type error**
The wrong parameter type was found in the command string. You may have specified a number where a string was expected, or vice versa.
Example: `SAMP:COUN '150' or SAMP:COUN A`

-105 **GET not allowed**
A Group Execute Trigger (GET) is not allowed within a command string.

-108 **Parameter not allowed**
More parameters were received than expected for the command. You may have entered an extra parameter, or you added a parameter to a command that does not accept a parameter. Example: `SYST:ERR? 10`

-109 **Missing parameter**
Fewer parameters were received than expected for the command. You omitted one or more parameters that are required for this command.

Example: `SAMP:COUN`

-112 Program mnemonic too long

A command header was received which contained more than the maximum 12 characters allowed. Example: `SAMPLE:PRETRIGGER:COUNT 10`

-113 Undefined header

A command was received that is not valid for this digitizer. You may have misspelled the command or it may not be a valid command. If you are using the short form of the command, remember that it may contain up to four letters. Example: `TRIGG:LEV 1.2`

-121 Invalid character in number

An invalid character was found in the number specified for a parameter value. Example: `STAT:QUES:ENAB #B01010102`

-123 Numeric overflow

A numeric parameter was found whose exponent was larger than 32,000. Example: `SAMP:COUN 1E34000`

-124 Too many digits

A numeric parameter was found whose mantissa contained more than 255 digits, excluding leading zeros.

-128 Numeric data not allowed

A numeric parameter was found but a character string was expected. Check the list of parameters to verify you have used a correct parameter type. Example: `TRIG:SOUR 2 EXT (should be TRIG:SOUR2 EXT)`

-138 Suffix not allowed

A suffix was received following a numeric parameter which does not accept a suffix. Example: `SAMP:COUN 1 SEC` (SEC is not a valid suffix).

-148 Character data not allowed

A character string was received but a numeric parameter was expected. Check the list of parameters to verify that you have used a valid parameter type. Example: `CAL:VAL XYZ`

-158 String data not allowed

A character string was received but is not allowed for the command. Check the list of parameters to verify that you have used a valid parameter type. Example: `CALC:LIM:LOW:STAT 'ON'`

-160 to -168 Block data errors

The digitizer does accept block data.

-170 to -178 **Expression errors**
The digitizer does not accept mathematical expressions.

-211 **Trigger ignored**
A Group Execute Trigger (GET) or `*TRG` was received but the trigger was ignored. Make sure the digitizer is in the “wait-for-trigger” state before issuing a trigger, and make sure the correct trigger source is selected.

-213 **Init ignored**
An `INITiate` command was received but could not be executed because a measurement was already in progress. Send a device clear to halt a measurement in progress and place the digitizer in the “idle” state.

-214 **Sample Trigger deadlock**
You .....

-215 **Arm Trigger deadlock**
You .....

-221 **Settings conflict**
You tried to set a pretrigger count that exceeds the sample count -1. Or you enabled one of the internal triggers as the source for a particular channel such as channel 2 (`TRIG:SOUR INT2`) and then tried to enable one of the limit checking features on channel 2 (`CALC2:LIM:UPP:STAT ON`).

-222 **Data out of range**
A numeric parameter value is outside the valid range for the command. Example: digitizer is on the 1V range and you send `TRIG:LEV -3`

-224 **Illegal parameter value**
A discrete parameter was received which was not a valid choice for the command. You may have used an invalid parameter choice.
Examples: `CAL:SOUR TTLT2` (TTLT2 is not a valid choice) or `SAMP:COUN ON` (ON is not a valid choice).

-230 **Data corrupt or stale**

-231 **Data questionable**

-240 **Hardware error**

-241 **Hardware missing**

-300 **Device-specific error**

-311 **Memory error**

-312 **PUD memory lost**

-313 **Calibration memory lost**

-330 **Self-test failed**

The digitizer's complete self-test failed from the remote interface (`*TST?` command). In addition to this error, more specific self-test errors are also reported. *See also "Self-Test Errors," following this section.*

-350 **Too many errors**

The error queue is full because more than 20 errors have occurred. No additional errors are stored until you remove errors from the queue. The error queue is cleared when power has been off, or after a `*CLS` (clear status) command has been executed.

-410 **Query INTERRUPTED**

A command was received which sends data to the output buffer, but the output buffer contained data from a previous command (the previous data is not overwritten). The output buffer is cleared when power has been off, or after a `*RST` (reset) command has been executed.

-420 **Query UNTERMINATED**

The digitizer was addressed to talk (i.e., to send data over the interface) but a command has not been received which sends data to the output buffer. For example, you may have executed a `SAMPle:COUNt <count>` command (which does not generate data) and then attempted an `ENTER` statement to read data from the remote interface.

-430 **Query DEADLOCKED**

A command was received which generates too much data to fit in the output buffer and the input buffer is also full. Command execution continues but all data is lost.

-440 **Query UNTERMINATED after indefinite response**

The `*IDN?` command must be the last query command within a command string. Example: `*IDN?;:SYST:VERS?`

300 **Not yet implemented**

1000 **Illegal when initiated**

1001 **Illegal while calibrating**

1002 **Trigger ignored**

1003 **Sample Trigger ignored**

1004 **Insufficient data for query**

1005 **Invalid channel number**

1006 **Invalid channel range**

1007 **Error in CAL**

1008 **Data fetch timed out waiting for trigger**

1009 **Error reading data, viMoveIn16 failed**

1010 **Self test failed**

1011 **Visa error**

Self-Test Errors

The self-test command (*TST?) will return a non-zero number if self-test fails. Self-test error descriptions are retrieved using the TEST:ERRor? <*test_number*> command. Use the number returned by self-test as the <*test_number*> to obtain the description of the failure.

Calibration Errors

Zero Calibration

The CAL:ZERO[<channel>]:ALL? command returns a non-zero number if zero calibration fails. For example, a return value of 0x0021 (binary value 100001) indicates that the 62 mV range and the 64V range failed. A “1” in the range position indicates a failure (range = 256, 64, 16, 4, 1, 0.25, 0.062). The error string returned by the SYST:ERR? command will contain information about the failure on the highest range (for 0x0021, binary value 100001, information is returned on the 64V range).

Zero Non-converging error.

A non-converging error usually indicates some internal problem with the instrument. It is recommended you run the self-test (*TST command) to identify any instrument problems.

Gain Calibration

Calibration value (CAL:VALue <*voltage*>) not within 85% to 98% of full scale.

You have entered a voltage with the CAL:VALue command that is not between 85% and 98% of the full scale range. For example, a calibration value between 0.85 and 0.98 is required on the 1V range

Gain Non-converging error.

A non-converging error usually indicates some internal problem with the instrument. It is recommended you run the self-test (*TST command) to identify any instrument problems.

If you use an external calibration source, you may have set the correct CAL:VALue but did not connect the calibration source to the digitizer’s input for the channel you are calibrating. The calibration source may still be connected to the last channel calibrated.

Appendix D

HP E1563A and E1564A Verification Tests

Introduction

This chapter provides information on performance verification of the HP E1563A 2-Channel and HP E1564A 4-Channel Digitizers. You can perform performance verification tests at two different levels depending on need:

- Functional Verification Test - A series of internal verification tests (self-tests) that give a high confidence that the digitizer is operational. The self-tests take less than 20 seconds to complete.
- Performance Verification Test - A complete set of tests that are recommended as an acceptance test when the instrument is first received or after performing calibration of the digitizer.

WARNING **Do not perform any of the following verification tests unless you are a qualified, service-trained technician and have read the WARNINGS and CAUTIONS in Chapter 1 (and the Warnings and Safety information preceding chapter 1 on page 4).**

Recommended Test Equipment

Test equipment recommended for the performance verification and calibration procedures are listed in Table D-1. Use a source with accuracy requirements indicated in the table for any substitute calibration standard.

Table D-1. Recommended Test Equipment.

Application	Recommended	Accuracy Requirements
Gain Calibration and Verification	Fluke 5700A	<1/5 digitizer spec ±1ppm linearity

Special care must be taken to ensure that the calibration standards and test procedures used do not introduce additional errors. Ideally, the standards used to test and calibrate the digitizer should be an order of magnitude more accurate than each digitizer range full scale error specification.

Test Conditions

All test procedures should comply with the following test conditions:

- Ambient temperature of the test area is between 18°C and 28°C and stable to within ±1°C.

- Ambient relative humidity of the test area is <80%.
- Must have a one hour warm-up with all input signals removed before verification or adjustment.
- Use only copper connections to minimize thermal offset voltages.
- Use shielded twisted Teflon® insulated cable or other high impedance, low dielectric absorption cable for all measurements to reduce high resistance errors.
- Keep cables as short as possible. Long test leads can act as an antenna causing the pick-up of ac signals and contributing to measurement error.
- Allow 5 minutes after handling input connections for thermal offset voltage settling.

Verification Tests

Recording Your Test Results

Make copies of the Performance Test Record at the end of this chapter for use in performance verifying each channel (use one test record copy per channel). The test record provides space to enter the results of each Performance Verification test and to compare the results with the upper and lower limits for the test.

The value in the "Measurement Uncertainty" column of the test record is derived from the specifications of the source used for the test, and represents the expected accuracy of the source. The value in the "Test Accuracy Ratio (TAR)" column of the test record is the ratio of digitizer accuracy to measurement uncertainty.

Performance Verification Test Programs

Performance Verification Test programs are provided so you can performance verify your digitizer. These programs were developed on a 486 IBM compatible computer running Windows 3.1® with an HP 82341 HP-IB interface and HP SICL/ Windows 3.1® and Windows NT® for HP-IB software.

C Programs

All projects written in C programming language require the following settings, files or paths to work properly:

Project Type: QuickWin application (.EXE)

Project Files:
1. <source code file name>.C (which includes the VISA.h header file)
2. One of the following files from the HP I/O Libraries for Instrument Control:
   [drive:]\VXIPNP\WIN\LIB\MSC\VISA.LIB (Microsoft® compiler)
   [drive:]\VXIPNP\WIN\LIB\BC\VISA.LIB (Borland® compiler)

Memory Model: Options | Project | Compiler | Memory Model ⇒ large

Directory Paths: Options | Directories
   Include File Paths: [drive:]\VXIPNP\WIN\INCLUDE
   Library File Paths: [drive:]\VXIPNP\WIN\LIB\MSC (Microsoft®)
   [drive:]\VXIPNP\WIN\LIB\BC (Borland®)

Printing Results Manually you can use the Performance Test Record as described on the previous page in the section "Recording Your Test Results".

Test Frequency See Table D-1, *Recommended Test Equipment*, for test equipment requirements. You should complete the Performance Verification tests at one year intervals. For heavy use or severe operating environments, perform the tests more often.

Functional Test

The procedure in this section is used to quickly verify that the digitizer is functioning. This test should be performed anytime the user wants to verify that the digitizer is connected properly and is responding to basic commands.

Functional Test Procedure

This test verifies that the digitizer is communicating with the command module, external controller, and/or external terminal by accepting the *TST? common command and performing a digitizer self-test. You have a high confidence (90%) that the digitizer is operational if self-test passes.

1. Verify that the digitizer and command module or system resource manager (e.g., embedded controller) are properly installed in the mainframe.

2. Remove any input connections to the digitizer input terminals. Errors may be induced by ac signals present on the digitizer's input terminals during a self-test.

3. Execute the digitizer self-test using the *TST? command.

4. A "0" returned means self-test passed with no failures. Any other value returned is a self-test error code and means a failure was detected. See the TEST command in Chapter 3 for obtaining information about self-test failures. See Appendix C and the TEST command in Chapter 3 for self-test error codes.

Note If an incorrect address is used, the digitizer will not respond. Verify proper address selection before troubleshooting.

Example: Self-Test This BASIC example performs a digitizer self-test. Any number other than 0 returned indicates a test failure. See Appendix C and the TEST command in Chapter 3 for self-test error codes.

```
10 OUTPUT 70905;"*TST?"        !Send the self-test command
20 ENTER 70905;A               !Read the test result
30 PRINT A                     !Display the result
40 END
```

Performance Verification

The procedures in this section are used to test the electrical performance of the digitizer using the specifications in Appendix A of the *HP E1563A and E1564A User's Manual* as the performance standards.

The Performance Verification Tests are recommended as acceptance tests when the instrument is first received. The performance verification tests should be repeated at each calibration interval following acceptance. If the HP E1563A or HP E1564A digitizer fails performance verification, adjustment or repair is needed.

Performance Verification Programs

Performance verification program source code is provided written in ANSI C. They are located on the driver CD and are titled E1563VER.C and E1564VER.C.

Zero Offset Verification

This procedure is used to check the zero offset performance of the HP E1563A or HP E1564A Digitizer. The digitizer's internal short is applied to the H (HI) and L (LO) input terminals of the channel being tested using the DIAG:SHORt <*channel*> command.

1. Check the "Test Conditions" section at the beginning of this chapter.

2. Execute the command DIAG:SHOR1 ON to enable the internal short across the H and L terminals of channel 1.

3. Select each range in the order shown in Table D-2. Compare the measurement results to the appropriate test limits shown in the table.

Table D-2. Zero Offset Verification Test Points.

INPUT	HP E1563A / HP E1564A Range	Error from nominal
internal H-L short DIAG:SHORt command	62 mV	± 15 µV
	0.25 V	± 61 µV
	1 V	± 244 µV
	4 V	± 976 µV
	16 V	± 3.9 mV
	64 V	± 15.6 mV
	256 V	± 62.5 mV

4. Repeat steps 2 and 3 for channel 2 on the E1563A 2-Channel Digitizer and channels 2 through 4 on the E1564A 4-Channel Digitizer changing the channel number in the DIAG:SHORt<*channel*> ON command.

Noise Verification

This procedure is used to check the noise performance of the HP E1563A or HP E1564A Digitizer. The digitizer's internal short is applied to the H (HI) and L (LO) input terminals of the channel being tested using the DIAG:SHORt <*channel*> ON command.

1. Check the "Test Conditions" section at the beginning of this chapter.
2. Execute the command DIAG:SHOR1 ON to enable the internal short across the H and L terminals of channel 1.
3. Set a sample interval of 25 µS by executing SAMP:TIM 25e-6.
4. Select the first range (62 mV) shown in Table D-3.

Table D-3. Noise Verification Test Points.

INPUT	HP E1563A / HP E1564A Range	Error from zero
internal H-L short DIAG:SHORt command	62 mV	23 µV max
	0.25 V	92 µV max
	1 V	366 µV max
	4 V	1.5 mV
	16 V	5.9 mV
	64 V	23.4 mV
	256 V	93.8 mV

5. Make 100 readings, sum them, divide by 100 and obtain the mean reading.
6. Calculate the standard deviation using the following formula (this is the rms noise value). "reading_n" represents the 100 readings where n = 1 to 100.

$$\sigma = \sqrt{\frac{\Sigma(reading_n)^2 - 100(mean)^2}{99}}$$

7. Record the rms noise value on the Performance Test Record and compare the result to the appropriate test limit shown in the test record or the above table.
8. Repeat steps 4, 5 and 6 for each range listed in table D-3.
9. Repeat steps 3 to 7 for channel 2 on the E1563A 2-Channel Digitizer and channels 2 through 4 on the E1564A 4-Channel Digitizer changing the channel number in the DIAG:SHORt<*channel*> ON command and executing the command prior to performing the steps.

Gain Verification

The gain verification tests check the positive and negative full scale gain on each range for each channel. An external DCV source provides the input and the digitizer's "L" terminal is connected to the "G" terminal connecting LO to GUARD. The input voltage is slightly less than full scale to avoid overloading the range.

1. Set up the digitizer as follows:
   Reset the digitizer *RST
   Set channel 1 to the 62 mV range; VOLT1:RANG 62E-3

2. Set the DC Standard output to 55 mV.

3. Perform the measurement using the INITiate command. Retrieve the reading using the DATA? 1,(@1) command.

4. Verify the result is within specified limits and record the result.

5. Change ranges using VOLT<*channel*>:RANG <range> and make a measurement for each DCV input and range shown in Table D-4 verifying the result is within specified limits. Record the result.

6. Repeat step 5 for channel 2 on the E1563A 2-Channel Digitizer and channels 2 through 4 on the E1564A 4-Channel Digitizer.

Table D-4 Gain Verification Test Points.

INPUT	HP E1563A / HP E1564A Range	Error from nominal
+55 mV	62 mV	± 30.25 µV
-55 mV	62 mV	± 30.25 µV
+0.24V	0.25V	± 110 µV
-0.24V	0.25V	± 110 µV
+0.95V	1 V	± 522.5 µV
-0.95V	1 V	± 522.5 µV
+3.8V	4 V	± 2.1 mV
-3.8V	4 V	± 2.1 mV
+15V	16 V	± 8.25 mV
-15V	16 V	± 8.25 mV
+60V	64 V	± 33 mV
-60V	64 V	± 33 mV
+100V	256 V	± 55 mV
-100V	256 V	± 55 mV

Filter Bandwidth

This test checks the filter input bandwidth for the 25 kHz filter on the HP E1563A or each of the four filters (1.5 kHz, 6 kHz, 25 kHz and 100 kHz) on the HP E1564A. The test This test uses an external source connected to the HI and LO Input terminals and has the "L" terminal connected to the "G" terminal. The digitizer is set to the 1V range for all tests.

1. Set up the digitizer as follows:
   Reset the digitizer *RST
   Set all channel's to 1V range; VOLT1:RANG 1; VOLT2:RANG 1; etc.
   Set input filter frequency to 25 kHz;INPut1:FILTer:FREQ 25e3
   INPut2:FILTer:FREQ 25e3
   Enable the input filter; INPut1:FILTer:STATe ON
   INPut2:FILTer:STATe ON

2. Set the AC Standard output to 0.95V @ 25 kHz and connect it to the digitizer's channel 1.

3. Perform the filter bandwidth measurement using the INITiate command. Retrieve the reading using the DATA? 1,(@1) command.

4. Record the result on the Performance Test Record and verify the result is within specified limits.

5. Move the AC Standard output to the channel 2 input. Perform the filter bandwidth measurement using the INITiate command. Retrieve the reading using the DATA? 1,(@2) command.

6. Verify the result is within specified limits and record the result.

7. Repeat steps 1 through 6 for channels 3 and 4 on the E1564A 4-Channel Digitizer using:
   VOLT3:RANG 1, INPut3:FILTer:FREQ 25e3,
   INPut3:FILTer:STATe ON, VOLT4:RANG 1,
   INPut4:FILTer:FREQ 25e3, INPut4:FILTer:STATe ON.

8. HP E1564A 4-Channel Digitizer: Test the remaining three filters present on the HP E1564A 4-Channel Digitizer. Repeat steps 2 through 6 for the remaining three input frequencies listed at the bottom of Table D-5 for channels 3 and 4.

   This requires you change input filters before you begin testing by executing the INPut<*channel*>:FILTer:FREQ <*filter_frequency*> command.

   Additionally, step 3 requires the DATA? 1,(@3) command and step 5 requires the DATA? 1,(@4) command.

Table D-5. Filter Bandwidth Verification Test Points.

INPUT	INPUT FREQ	E1563A RANGE	Error from input value
1 V	25 kHz	1V	-3 dB ±2 dB
INPUT	INPUT FREQ	E1564A RANGE	Error from input value
1 V	25 kHz	1V	-3 dB ±2 dB
1 V	1.5 kHz	1V	-3 dB ±2 dB
1 V	6 kHz	1V	-3 dB ±2 dB
1 V	100 kHz	1V	-3 dB ±2 dB

Performance Test Record

Note The Performance Test Record for the HP E1563A or HP E1564A digitizer, is a form you can copy and use to record performance test results for the digitizer. This form shows the digitizer accuracy limits, the measurement uncertainty from the source and the test accuracy ratio (TAR).

Note The accuracy, measurement uncertainty and TAR values shown the Performance Test Record are valid ONLY for the specific test conditions, test equipment and assumptions described. If you use different test equipment or change the test conditions, you will need to compute the specific values for your test setup.

Digitizer Accuracy

Accuracy is defined for gain measurements using the 1-year specifications in Appendix A. The "High Limit" and "Low Limit" columns represent the digitizer accuracy for the specified test conditions.

Measurement Uncertainty

Measurement uncertainties listed in the Performance Test Record are calculated assuming a Fluke 5700A for all measurements. The uncertainties describe error you can expect from the source. These uncertainties are calculated from the 90-day accuracy specifications for the Fluke 5700A.

Test Accuracy Ratio (TAR)

The "Test Accuracy Ratio (TAR)" = (high limit - expected measurement) divided by measurement uncertainty. "N/A" means measurement uncertainty and TAR do not apply to the measurement. A small TAR indicates the uncertainty of the source signal starts to approach the digitizer's specification limit.

HP E1563A / HP E1564A Digitizer Performance Test Record

Date: ____________________________

Test Facility: ____

Name _____________________________________ Report No._________________________________

Address____________________________________ Date ____________________________________

City/State__________________________________ Customer_________________________________

Phone______________________________________ Tested By_________________________________

Model________________________________ Ambient temperature __________________________°C

Serial No.____________________________ Relative humidity ____________________________%

Firmware Rev.________________________ Line frequency_________________________Hz (nominal)

Special Notes:

Test Equipment Used	Model No.	Trace No.	Cal Due Date

❒HP E1563A 2-Channel Digitizer **❒HP E1564A 4-Channel Digitizer**

PERFORMANCE TEST RECORD **Date____________________**

CHANNEL: ❒1 ❒2 ❒3 ❒4

Test Input	Digitizer Range	Low Limit	Measured Reading	High Limit	Meas Uncert	Test Accuracy Ratio
Zero Offset Test						
0	62 mV	-.000020		.000020	N/A	N/A
0	250 mV	-.000078		.000078	N/A	N/A
0	1V	-.000300		.000300	N/A	N/A
0	4V	-.001200		.001200	N/A	N/A
0	16V	-.021000		.021000	N/A	N/A
0	64V	-.028000		.028000	N/A	N/A
0	256V	-.079000		.079000	N/A	N/A
Noise Test						
0	62 mV	0		57 µV max	N/A	N/A
0	250 mV	0		180 µV max	N/A	N/A
0	1V	0		720 µV max	N/A	N/A
0	4V	0		2.8 mV max	N/A	N/A
0	16V	0		14.7 mV	N/A	N/A
0	64V	0		48 mV	N/A	N/A
0	256V	0		189 mV	N/A	N/A
Gain Test						
55 mV	62 mV	.05498100		.05501900	.0000011	>10:1
-55 mV	62 mV	-.05501900		-.05498100	.0000011	>10:1
200 mV	250 mV	.19993200		.20006800	.0000022	>10:1
-200 mV	250 mV	-.20006800		-.19993200	.0000022	>10:1
0.95V	1V	.94967700		.95032300	.0000069	>10:1
-0.95V	1V	-.95032300		-.94967700	.0000069	>10:1
3.8V	4V	3.7987080		3.8012920	.000023	>10:1
-3.8V	4V	-3.8012920		-3.7987080	.000023	>10:1
15V	16V	14.994900		15.005100	.000083	>10:1
-15V	16V	-15.005100		-14.994900	.000083	>10:1

Test Input	Digitizer Range	Low Limit	Measured Reading	High Limit	Meas Uncert	Test Accuracy Ratio
Gain Test (continued)						
60V	64V	59.9796		60.0204	.00046	>10:1
-60V	64V	-60.0204		-59.9796	.00046	>10:1
100V	256V	99.9660		100.0340	.0007	>10:1
-100V	256V	-100.0340		-99.9660	.0007	>10:1
E1563A 25 kHz Filter Bandwidth Test						
1V @ 1 MHz	1V no filter	-5 dB		-1 dB	N/A	N/A
1V @ 25 kHz	1V 25 kHz filter	-5 dB		-1 dB	N/A	N/A
E1564A Filter Bandwidth Test (4 filters)						
1V @ 1 MHz	1V no filter	-5 dB		-1 dB	N/A	N/A
1V @ 1.5 kHz	1V 1.5 kHz filter	-5 dB		-1 dB	N/A	N/A
1V @ 6 kHz	1V 6 kHz filter	-5 dB		-1 dB	N/A	N/A
1V @ 25 kHz	1V 25 kHz filter	-5 dB		-1 dB	N/A	N/A
1V @ 100 kHz	1V 100 kHz filter	-5 dB		-1 dB	N/A	N/A

Appendix E
HP E1563A and E1564A Adjustments

Introduction

This chapter contains procedures for adjusting the calibration constants in the HP E1563A and HP E1564A digitizer. See the section titled "Calibration Interval" for recommendations on time intervals.

Write Enable CAL CONSTANTS

You must set the module's "FLASH" and "CALIBRATION CONSTANTS" switches to the "Write Enable" position before you perform any adjustment. This allows modified calibration constants to be stored in memory when you execute the CAL:STORe command.

Closed-Cover Electronic Calibration

The HP E1563A and HP E1564A Digitizers feature closed-cover electronic calibration. ***There are no internal mechanical adjustments.*** You input the CAL:VALue <voltage> command and the digitizer will measure the applied voltage when performing a zero or a range gain calibration, then calculate correction factors based on this known input reference value. You store the new correction factors in non-volatile memory using the CAL:STORe command. (Non-volatile memory does not change when power is turned off or after a remote interface reset.)

WARNING **Do not perform any of the following adjustments unless you are a qualified, service-trained technician, and have read the WARNINGS and CAUTIONS in Chapter 1 (and the Warnings and safety information preceding chapter 1 on page 4). Adjustment procedures should be performed in the order shown in this manual.**

Hewlett-Packard Calibration Services

Contact your local HP Service Center for a low cost recalibration when your digitizer is due for calibration. Both the HP E1563A or HP E1564A digitizer is supported on automated calibration systems which allow us to provide this service at competitive prices.

Calibration Interval

The HP E1563A and HP E1564A digitizer should be calibrated on a regular interval determined by the measurement accuracy requirements of your application. A 90 day interval is recommended for the most demanding applications, while a 1 year or 2 year interval may be adequate for less demanding applications. HP does not recommend extending calibration intervals beyond 2 years in any application.

Whatever calibration interval you select, HP recommends that complete re-adjustment should always be performed at the calibration interval. This will increase your confidence that the HP E1563A or HP E1564A will remain in specification for the next calibration interval. This criteria for readjustment provides the best measure of the digitizer's long-term stability. Performance data measured this way can easily be used to extend future calibration intervals.

Aborting an Adjustment in Progress

Sometimes it becomes necessary to abort an adjustment once the procedure has been initiated. Issuing a remote interface device clear command will abort the adjustment in progress.

Caution **Never turn off mainframe power while the digitizer is making an adjustment. If power is removed during a zero adjustment, ALL calibration memory may be lost. If power is removed during any gain adjustment, calibration memory for the function being adjusted may be lost.**

Adjustment Procedures

Adjustment Conditions

See Table D-1, *Recommended Test Equipment*, for test equipment requirements. For optimum performance, all adjustment procedures should comply with following test conditions:

- Ambient temperature of the test area is between 18°C and 28°C and stable to within ±1°C.
- Ambient relative humidity of the test area is <80%.
- Must have a one hour warm-up with all input signals removed.
- Shielded twisted Teflon® insulated cable or other high impedance, low dielectric absorption cable is recommended for all measurements.
- Keep cables as short as possible. Long test leads can act as an antenna causing pick-up of ac signals and contributing to measurement errors.

General Procedure

Follow each adjustment by a performance verification test for added confidence. We recommend the following general procedure.

- Perform the Zero Adjustment Procedure.
- Perform the Gain Adjustment Procedure(s).
- Perform the Performance Verification Tests.

Caution **ORDER OF ADJUSTMENTS REQUIREMENT**
Range adjustments MUST be performed in the order given in the adjustment table. An accurate range adjustment requires the range adjustments prior to the one in progress be within specification.

Caution **ZERO ADJUSTMENT REQUIREMENT**
The zero adjustment must be a recent adjustment prior to performing the gain adjustments. It is recommended you perform the zero adjustment one time just before performing the gain adjustments.

Adjustment Programs

The driver CD received with this module contains calibration and performance verification program source code written in ANSI C. Calibration programs are E1563CAL.C and E1564CAL.C. Performance verification programs are E1563VER.C and E1564VER.C.

Zero Adjustment

This procedure sets the zero calibration constants for each digitizer range. The digitizer calculates a new offset correction constant for the current range when the CALibration:ZERO[<*channel*>] command is executed. The digitizer calculates a new set of offset correction constants for all ranges of a channel when the CALibration:ZERO[<*channel*>]:ALL? command is executed. The digitizer will sequence through all ranges automatically and calculate new zero offset calibration constants automatically.

1. Reset the Digitizer by executing a *RST command.
2. Switch the internal short across each channel’s input by executing the DIAG:SHORt<*channel*> command for all channels e.g., DIAG:SHOR1; DIAG:SHOR2; etc.
3. Send the CAL:VAL 0 input CALibration value command.
4. Perform the adjustment by sending the CAL:ZERO<*channel*>:ALL? command once for each channel and reading the calibration success result (a non-zero response indicates a calibration error occurred).

Caution **Do not remove power from the mainframe during the digitizer’s Zero Adjustment. You may lose ALL calibration memory if power is removed while the digitizer is adjusting.**

Note The zero adjustment procedure takes about 20 seconds per channel to calculate new zero offset cal constants for all ranges of the channel.

E1563A Gain Adjustment

The zero adjustment procedure MUST have been recently performed prior to beginning any gain adjustment procedure. Zero adjustment should be performed one time followed by the other gain adjustments.

1. Reset the HP E1563A Digitizer by executing a *RST command.

2. Set the DC Standard output to 55 mV for the first gain adjustment.

3. Connect the DC Standard output across the HP E1563A "H" and "L" input terminals of channel 1.

4. Prepare the HP E1563A for calibration:
   Set the channel's range: VOLT<*channel*>:RANG <*range*>
   Set calibration source to external: CAL:SOUR EXT
   Send input value: CAL:VAL <*input voltage*>
   (see Table E-1, Gain Adjustment Range Input Voltages, for <*range*> and <*input voltage*> values)

5. Perform the adjustment by sending the CAL:GAIN<*channel*> command (adjusts each channel in about 5 seconds).

6. Send the SYST:ERR? command and read the result to verify the calibration command was successful. See Chapter 3, SCPI Command Reference, for details on reading system errors.

7. Repeat steps 3 through 6 for the ranges and inputs given in Table E-1.

8. Repeat steps 2 through 7 for channel 2.

Note Each range in the gain adjustment procedure for each channel takes less than 5 seconds to complete.

Table E-1. Gain Adjustment Range Input Voltages.

Channel Range	Input Voltage
62 mV	55 mV
0.25V	0.24V
1V	0.95V
4V	3.8V
16V	15V

** Valid calibration input values sent to the digitizer are 0.85 to 0.98 of Full Scale for the range being adjusted. The <*input voltage*> parameter of the CAL:VAL command must equal the actual input value. For example, if you input 0.9V to calibrate the 1V range instead of 0.95, you send CAL:VAL 0.9 to the digitizer prior to the CAL:GAIN<*channel*> command.

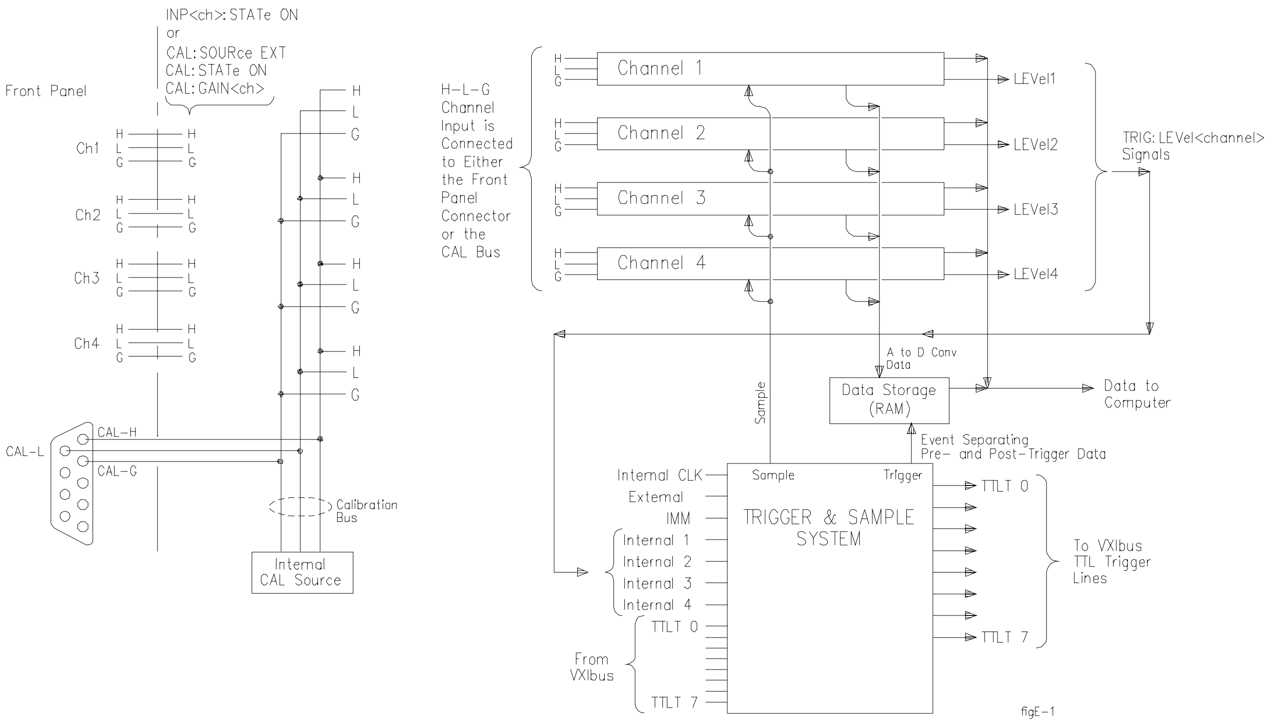
E1564A Gain Adjustment

The zero adjustment procedure MUST have been recently performed prior to beginning any gain adjustment procedure. Zero adjustment should be performed one time followed by the other gain adjustments.

The HP E1564A 4-Channel Digitizer has an internal DAC that outputs to a calibration bus on the front panel D-connector. This procedure uses the calibration bus and does not require an external DC Standard.

1. Reset the HP E1564A Digitizer by executing a *RST command.

2. Connect a voltmeter to the Calibration Bus Output on the front panel D-connector (see the following diagram). Set the voltmeter to the DCV function.



3. Prepare the HP E1564A for calibration:
   Set the channel's range: VOLT<*channel*>:RANG <*range*>
   Set the calibration source to internal: CAL:SOUR INT
   Set the CAL DAC output voltage: CAL:DAC:VOLT <*voltage*>
   (see the following Gain Adjustment Points table for <*range*> settings and CAL DAC <*voltage*> setting)

4. Note the voltmeter reading from the calibration bus output.

5. Send the value measured from the calibration bus output as the parameter for the calibration value: CAL:VAL <*voltage*>

6. Perform the adjustment by sending the CAL:GAIN<*channel*> command (adjusts each channel in about 5 seconds).

7. Send the SYST:ERR? command and read the result to verify the calibration command was successful. See Chapter 3, SCPI Command Reference, for details on reading system errors.

8. Repeat steps 3 through 7 for the ranges and inputs given in Table E-2.

9. Repeat steps 3 through 8 for channel's 2, 3 and 4.

Note Each range in the gain adjustment procedure for each channel takes less than 5 seconds to complete.

Table E-2. Gain Adjustment Range Input Voltages.

Channel Range	CAL DAC Voltage
62 mV	55 mV
0.25V	0.24V
1V	0.95V
4V	3.8V
16V	15V

** Valid calibration input values sent to the digitizer are 0.85 to 0.98 of Full Scale for the range being adjusted. The <*input voltage*> parameter of the CAL:VAL command must equal the actual input value. For example, if you input 0.9V to calibrate the 1V range instead of 0.95, you send CAL:VAL 0.9 to the digitizer prior to the CAL:GAIN<*channel*> command.

Store Calibration Constants

You must set the "FLASH" and "CALIBRATION CONSTANTS" switch to "write enable" before you can store new calibration constants. It is recommended you do this prior to starting the calibration procedures. Execute the CAL:STORe command to store the new calibration constants following the calibration procedures. Restore the switches to the "Read Only" position after you store the new calibration constants.

Index
HP E1563A and HP E1564A

A

A16 Address Space, 113
A16 Address Space Inside the Command Module, 115
A16 Address Space Outside the Command Module, 114
abbreviated commands, 40
ABORt subsystem, 42
  ABORt, 42
Address
  A16 address space, 113
  base address, 113
  logical, 114, 115
Addressing
  Register, 113
adjustment conditions, 158
adjustment, aborting, 158
adjustments, 157
  dc gain (E1563A), 160
  dc gain (E1564A), 161
  zero, 159
Application Examples, 35

B

Base Address, 113
boolean parameters, 41

C

C verification programs, 146
CAL constants, 157
CALCulate subsystem, 43
  CALCulate:LIMit:FAIL?, 43
  CALCulate:LIMit:LOWer:DATA, 44
  CALCulate:LIMit:LOWer:DATA?, 45
  CALCulate:LIMit:LOWer[:STATe], 43
  CALCulate:LIMit:LOWer[:STATe]?, 44
  CALCulate:LIMit:UPPer:DATA, 46
  CALCulate:LIMit:UPPer:DATA?, 46
  CALCulate:LIMit:UPPer[:STATe], 45
  CALCulate:LIMit:UPPer[:STATe]?, 45
calibration, 157
  interval, 157
calibration bus, 47, 50
calibration erros, 144
Calibration Flash ROM Address register, 123
Calibration Flash ROM Data register, 123
calibration services by Hewlett-Packard Co., 157
calibration source
  external, 50
  internal, 50
Calibration Source register, 123
CALibration subsystem, 47
  CALibration:DAC:VOLTage, 47
  CALibration:DATA?, 48
  CALibration:GAIN, 48
  CALibration:SOURce, 50
  CALibration:SOURce?, 50
  CALibration:STATe, 51
  CALibration:STATe?, 51
  CALibration:STORe, 51
  CALibration:VALue, 52
  CALibration:VALue?, 53
  CALibration:ZERO, 53
  CALibration:ZERO:ALL?, 54
calibration voltage, E1564A
  internal, 47
Cautions, 158, 159
clearing the multimeter
  error queue, 87
Command, 114, 115
Command Module
  A16 address space inside the, 115
  A16 address space outside the, 114
command reference
  ABORt subsystem, 42
  CALCulate subsystem, 43
  CALibration subsystem, 47
  DIAGnostic subsystem, 55
  FORMat subsystem, 63
  INITiate subsystem, 64
  INPut subsystem, 66
  OUTPut subsystem, 68
  SAMPle subsystem, 78
  SENSe subsystem, 70
  STATus subsystem, 83, 85, 86
  SYSTem subsystem, 87
  TEST subsystem, 88
  TRIGger subsystem, 94
command separator, 39
command types, 39

commands
abbreviated, 40
common command format, 39
implied, 40
linking, 41
parameter types, 41
SCPI command format, 39
separators, 39
upper case vs. lower case, 40
common (*) command
format, 39
CVTable Channel 1 register, 122
CVTable Channel 2 register, 122
CVTable Channel 3 register, 122
CVTable Channel 4 register, 122

D

Descriptions
register, 116
DIAGnostic subsystem, 55
DIAGnostic:DAC:GAIN, 56
DIAGnostic:DAC:OFFSet, 55
DIAGnostic:DAC:OFFSet:RAMP, 55
DIAGnostic:DAC:SOURce, 57
DIAGnostic:DAC:SOURce:RAMP, 57
DIAGnostic:INTerrupt:LINE, 58
DIAGnostic:MEMory:SIZE, 58
DIAGnostic:MEMory:SIZE?, 59
DIAGnostic:PEEK?, 59
DIAGnostic:POKE, 60
DIAGnostic:SHORt, 61
DIAGnostic:SHORt?, 62
DIAGnostic:STATus?, 62
discrete parameters, 41

E

error messages
execution errors, 139
self-test errors, 144
error queue
capacity, 87
clearing, 87
reading, 87
ERRor, see SYSTem subsystem, 87
Example programs
hardware used, 35
Execution
program, 130

F

format
common command, 39
SCPI command, 39
FORMat subsystem, 63
FORMat[:DATA], 63
FORMat[:DATA]?, 63
functional test procedure, 147

H

HP I/O Libraries
for Instrument Control, 146

I

implied commands, 40
INITiate subsystem, 64
INITiate:CONTinuous, 64
INITiate:CONTinuous?, 65
INITiate[:IMMediate], 64
INPut subsystem, 66
INPut:FILTer[:LPASs]:FREQ, 66
INPut:FILTer[:LPASs]:FREQ?, 66
INPut:FILTer[:LPASs][:STATe], 67
INPut:FILTer[:LPASs][:STATe]?, 67
INPut[:STATe], 67
INPut[:STATe]?, 67
internal calibration source, 50
internal calibration voltage, 47
internal trigger detection, 29
Interrupt Control Register, 120
Interrupt source register, 121

L

LADDR, 114
linking commands, 41
Logical Address
factory setting, 114, 115
register-based, 114
setting, 114, 115

M

MASTer mode, 30, 135
master-slave diagrams, 30, 135
master-slave operation, 30, 135

mode
MASTer, 30, 135
NORMal, 30, 135
SLAVe, 31, 135
Module
A16 address space inside the command, 115
A16 address space outside the command, 114

N

NORMal mode, 30, 135
numeric parameters, 41

O

Offset and Cache Count Register, 119
OPERation Data Register, 84
optional parameters, 41
OUTPut subsystem, 68
OUTPut:TTLT<n>:SOURce, 68
OUTPut:TTLT<n>:SOURce?, 68
OUTPut:TTLT<n>[:STATe], 69
OUTPut:TTLT<n>[:STATe]?, 69
overload condition, input, 34
overload reporting, 34

P

parameters
boolean, 41
discrete, 41
numeric, 41
optional, 41
types of, 41
Performance Test Record, 152, 153
performance verification
noise, 149
zero offset, 148
performance verification programs, 148
power-on state, 27
Pre-Trigger Count High Word register, 127
Pre-Trigger Count Low Word register, 127
printing test results, 147
Program execution, 130
Program Timing, 130
Program Timing and Execution, 130
Programming
Register-based, 113
programs, verification tests, 146

Q

QUEStionable Data Register, 83, 85
Questionable Data Register, 84
questionable data, status, 83

R

Range, Filter, Connect Channel 3, 4 Register, 124
Reading
registers, 113
reading the error queue, 87
Register
the device type, 117
the ID, 117
the status/control, 118
Register Addressing, 113
Register Descriptions, 116
Register-based Programming, 113
base address, 113
description, 113
Registers
base address, 113
reading registers, 113
the READ, 116
the WRITE, 116
writing to registers, 113
reset state, 27

S

safety symbols, 4
Sample Control register, 129
Sample Count High Byte register, 127
Sample Count Low Word register, 128
Sample Period High Byte register, 127
Sample Period Low Word register, 127
SAMPle subsystem, 78
SAMPle:COUNt, 78
SAMPle:COUNt?, 79
SAMPle:PRETrigger:COUNt, 79
SAMPle:PRETrigger:COUNt?, 80
SAMPle:SLOPe, 80
SAMPle:SLOPe?, 80
SAMPle:SOURce, 81
SAMPle:SOURce?, 82
SAMPle:TIMer, 82
SAMPle:TIMer?, 82
SAMPle[:IMMediate], 80
Samples Taken High Byte register, 122
Samples Taken Low Word register, 122

SCPI commands
abbreviated, 40
command format, 39
implied, 40
linking with common (*) commands, 41
parameters, 41
SCPI, programming examples
application SCPI command strings, 35
self-test errors, 144
SENSe subsystem, 70
SENSe:DATA:COUNt?, 73
SENSe:DATA:CVT?, 73
SENSe:DATA?, 70, 72
SENSe:ROSCillator:EXTernal:FREQuency, 74
SENSe:ROSCillator:EXTernal:FREQuency?, 74
SENSe:ROSCillator:SOURce, 75
SENSe:ROSCillator:SOURce?, 75
SENSe:SWEep:OFFSet:POINts, 76
SENSe:SWEep:OFFSet:POINts?, 76
SENSe:SWEep:POINts, 75
SENSe:SWEep:POINts?, 76
SENSe:VOLTage[:DC]:RANGe, 76
SENSe:VOLTage[:DC]:RANGe?, 77
SENSe:VOLTage[:DC]:RESolution, 77
SLAVe mode, 31, 135
Standard Commands for Programmable Instrument SCPI, 39
Standard Event Register, 84, 85
Status Byte Register, 84, 85
STATus subsystem, 83, 85, 86
STATus:OPERation:CONDition?, 85
STATus:OPERation:ENABle, 85
STATus:OPERation:ENABle?, 85
STATus:OPERation[:EVENt]?, 85
STATus:PRESet, 86
STATus:QUEStionable:CONDition?, 86
STATus:QUEStionable:ENABle, 86
STATus:QUEStionable:ENABle?, 86
STATus:QUEStionable[:EVENt]?, 86
symbols, safety, 4
SYSTem subsystem, 87
SYSTem:ERRor?, 87
SYSTem:VERSion?, 87

T

test conditions, 145
TEST subsystem, 88
TEST:ERRor?, 88
TEST:NUMBer?, 88
TEST:TST[:RESults], 93
The Device Type Register, 117
The ID Register, 117
The READ Registers, 116
The Status/Control Register, 118
The WRITE Registers, 116
Timing
program, 130
trigger, 29
external, 30
internal, 29
sources per channel, 29
two common sources, 29
Trigger Control register, 128
trigger mode
MASTer, 30, 135
NORMal, 30, 135
SLAVe, 31, 135
TRIGger subsystem, 94
TRIGger:LEVel, 94
TRIGger:LEVel?, 95
TRIGger:MODE, 95
TRIGger:MODE?, 96
TRIGger:SLOPe, 96
TRIGger:SLOPe?, 97
TRIGger:SOURce, 97
TRIGger:SOURce?, 98
TRIGger[:IMMediate], 94
Trigger/Interrupt Level Channel 1 register, 125
Trigger/Interrupt Level Channel 2 register, 125
Trigger/Interrupt Level Channel 3 register, 126
Trigger/Interrupt Level Channel 4 register, 126

V

verification tests, 145
functional, 147
performance, 148
programs, 146

W

Warnings and Cautions, 145, 157
Writing to Registers, 113