tek.com/keithley

Model 2182A Nanovoltmeter

User's Manual

2182A-900-01 Rev. C July 2022

2182A-900-01C

KEITHLEY
A Tektronix Company

Model 2182A

Nanovoltmeter

User's Manual

These are the original instructions in English.

Document number: 2182A-900-01 Rev. C July 2022

Safety precautions

The following safety precautions should be observed before using this product and any associated instrumentation. Although some instruments and accessories would normally be used with nonhazardous voltages, there are situations where hazardous conditions may be present.

This product is intended for use by personnel who recognize shock hazards and are familiar with the safety precautions required to avoid possible injury. Read and follow all installation, operation, and maintenance information carefully before using the product. Refer to the user documentation for complete product specifications.

If the product is used in a manner not specified, the protection provided by the product warranty may be impaired.

The types of product users are:

Responsible body is the individual or group responsible for the use and maintenance of equipment, for ensuring that the equipment is operated within its specifications and operating limits, and for ensuring that operators are adequately trained.

Operators use the product for its intended function. They must be trained in electrical safety procedures and proper use of the instrument. They must be protected from electric shock and contact with hazardous live circuits.

Maintenance personnel perform routine procedures on the product to keep it operating properly, for example, setting the line voltage or replacing consumable materials. Maintenance procedures are described in the user documentation. The procedures explicitly state if the operator may perform them. Otherwise, they should be performed only by service personnel.

Service personnel are trained to work on live circuits, perform safe installations, and repair products. Only properly trained service personnel may perform installation and service procedures.

Keithley products are designed for use with electrical signals that are measurement, control, and data I/O connections, with low transient overvoltages, and must not be directly connected to mains voltage or to voltage sources with high transient overvoltages. Measurement Category II (as referenced in IEC 60664) connections require protection for high transient overvoltages often associated with local AC mains connections. Certain Keithley measuring instruments may be connected to mains. These instruments will be marked as category II or higher.

Unless explicitly allowed in the specifications, operating manual, and instrument labels, do not connect any instrument to mains.

Exercise extreme caution when a shock hazard is present. Lethal voltage may be present on cable connector jacks or test fixtures. The American National Standards Institute (ANSI) states that a shock hazard exists when voltage levels greater than 30 V RMS, 42.4 V peak, or 60 VDC are present. A good safety practice is to expect that hazardous voltage is present in any unknown circuit before measuring.

Operators of this product must be protected from electric shock at all times. The responsible body must ensure that operators are prevented access and/or insulated from every connection point. In some cases, connections must be exposed to potential human contact. Product operators in these circumstances must be trained to protect themselves from the risk of electric shock. If the circuit is capable of operating at or above 1000 V, no conductive part of the circuit may be exposed.

Do not connect switching cards directly to unlimited power circuits. They are intended to be used with impedance-limited sources. NEVER connect switching cards directly to AC mains. When connecting sources to switching cards, install protective devices to limit fault current and voltage to the card.

Before operating an instrument, ensure that the line cord is connected to a properly-grounded power receptacle. Inspect the connecting cables, test leads, and jumpers for possible wear, cracks, or breaks before each use.

When installing equipment where access to the main power cord is restricted, such as rack mounting, a separate main input power disconnect device must be provided in close proximity to the equipment and within easy reach of the operator.

For maximum safety, do not touch the product, test cables, or any other instruments while power is applied to the circuit under test. ALWAYS remove power from the entire test system and discharge any capacitors before connecting or disconnecting cables or jumpers, installing or removing switching cards, or making internal changes, such as installing or removing jumpers.

Do not touch any object that could provide a current path to the common side of the circuit under test or power line (earth) ground. Always make measurements with dry hands while standing on a dry, insulated surface capable of withstanding the voltage being measured.

For safety, instruments and accessories must be used in accordance with the operating instructions. If the instruments or accessories are used in a manner not specified in the operating instructions, the protection provided by the equipment may be impaired.

Do not exceed the maximum signal levels of the instruments and accessories. Maximum signal levels are defined in the specifications and operating information and shown on the instrument panels, test fixture panels, and switching cards.

When fuses are used in a product, replace with the same type and rating for continued protection against fire hazard.

Chassis connections must only be used as shield connections for measuring circuits, NOT as protective earth (safety ground) connections.

If you are using a test fixture, keep the lid closed while power is applied to the device under test. Safe operation requires the use of a lid interlock.

If a screw is present, connect it to protective earth (safety ground) using the wire recommended in the user documentation.

The symbol on an instrument means caution, risk of hazard. The user must refer to the operating instructions located in the user documentation in all cases where the symbol is marked on the instrument.

The symbol on an instrument means warning, risk of electric shock. Use standard safety precautions to avoid personal contact with these voltages.

The symbol on an instrument shows that the surface may be hot. Avoid personal contact to prevent burns.

The symbol indicates a connection terminal to the equipment frame.

If this symbol is on a product, it indicates that mercury is present in the display lamp. Please note that the lamp must be properly disposed of according to federal, state, and local laws.

The **WARNING** heading in the user documentation explains hazards that might result in personal injury or death. Always read the associated information very carefully before performing the indicated procedure.

The **CAUTION** heading in the user documentation explains hazards that could damage the instrument. Such damage may invalidate the warranty.

The **CAUTION** heading with the symbol in the user documentation explains hazards that could result in moderate or minor injury or damage the instrument. Always read the associated information very carefully before performing the indicated procedure. Damage to the instrument may invalidate the warranty.

Instrumentation and accessories shall not be connected to humans.

Before performing any maintenance, disconnect the line cord and all test cables.

To maintain protection from electric shock and fire, replacement components in mains circuits — including the power transformer, test leads, and input jacks — must be purchased from Keithley. Standard fuses with applicable national safety approvals may be used if the rating and type are the same. The detachable mains power cord provided with the instrument may only be replaced with a similarly rated power cord. Other components that are not safety-related may be purchased from other suppliers as long as they are equivalent to the original component (note that selected parts should be purchased only through Keithley to maintain accuracy and functionality of the product). If you are unsure about the applicability of a replacement component, call a Keithley office for information.

Unless otherwise noted in product-specific literature, Keithley instruments are designed to operate indoors only, in the following environment: Altitude at or below 2,000 m (6,562 ft); temperature 0 °C to 50 °C (32 °F to 122 °F); and pollution degree 1 or 2.

To clean an instrument, use a cloth dampened with deionized water or mild, water-based cleaner. Clean the exterior of the instrument only. Do not apply cleaner directly to the instrument or allow liquids to enter or spill on the instrument. Products that consist of a circuit board with no case or chassis (e.g., a data acquisition board for installation into a computer) should never require cleaning if handled according to instructions. If the board becomes contaminated and operation is affected, the board should be returned to the factory for proper cleaning/servicing.

Safety precaution revision as of June 2018.

Table of contents

Section 1

Introduction

In this section:

Welcome

The two-channel Model 2182A Nanovoltmeter is optimized for making stable, low noise voltage measurements and for characterizing low resistance materials and devices reliably and repeatably.

Features of the 2182A Nanovoltmeter include:

- 1 nV sensitivity to provide accurate ultra-low voltage measurements.
- Typically just 15 nV peak-to-peak noise at 1 s response time, with 40 nV to 50 nV peak-to-peak noise at 60 ms, which ensures low noise levels over a wide range of useful response times.
- Dual channels support measuring voltage, temperature, or the ratio of an unknown resistance to a reference resistor.
- Measurement cycle is synchronized to the power line ac cycle to minimize variations due to readings that begin at different phases of the line cycle. The result is exceptionally high immunity to line interference with little or no shielding and filtering required.
- Built-in thermocouple linearization and cold junction compensation to simplify making accurate temperature measurements.
- Internal polarity reversal measurement technique to eliminate thermal error sources.
- Optimized for use with the Keithley Model 6220 and 6221 current sources, which allows both instruments to be operated like a single instrument when making differential conductance, pulsed, and resistance measurements.
- Delta mode current reversal measurement technique when combined with the 6220 or 6221 to enable resistance measurements down to 10 nΩ.

Extended warranty

Additional years of warranty coverage are available on many products. These valuable contracts protect you from unbudgeted service expenses and provide additional years of protection at a fraction of the price of a repair. Extended warranties are available on new and existing products. Contact your local Keithley Instruments office, sales partner, or distributor for details.

Contact information

If you have any questions after you review the information in this documentation, please contact your local Keithley Instruments office, sales partner, or distributor. You can also call the Tektronix corporate headquarters (toll-free inside the U.S. and Canada only) at 1-800-833-9200. For worldwide contact numbers, visit tek.com/en/contact-tek.

Organization of manual sections

The information in this manual is organized into the following major categories:

- **Installation:** Describes how to turn on the instrument and set up remote communications interfaces.
- **Instrument description:** Describes the front-panel and rear-panel components, error and status messages, and reset.
- **Making measurements:** Describes how to make test connections and how to make measurements.
- **Measure considerations:** Describes how to minimize noise and other unwanted signals that can adversely affect low-level measurements.
- **Delta, pulse delta, and differential conductance:** Describes how to use the delta, pulse delta, and differential conductance features.
- **Stepping and scanning:** Describes how to use the 2182A for stepping and scanning.
- **Reading buffers:** Describes how to use the reading buffers.
- **Triggering:** Describes the front panel and remote triggering options.
- **Limits:** Describes how to use limits and provides a resistor-sorting application example.
- **Maintenance:** Contains information about instrument maintenance, including line fuse replacement and firmware upgrades.
- **Introduction to SCPI commands:** Describes how to control the instrument using SCPI commands.
- **SCPI command reference:** Contains programming notes and an alphabetical listing of all SCPI commands available for the 2182A.

- **Common commands:** Contains descriptions of IEEE Std 488.2 common commands.
- **Program examples:** Provides code samples.
- **Status model:** Describes the 2182A status model.
- **Model 182 emulation commands:** Provides a list of commands for 182 emulation.
- **IEEE-488 bus overview:** Describes the IEEE-488 bus.

General ratings

The general ratings of the 2182A are listed in the following table.

Category	Specification
Supply voltage range	100 V setting: 90 V to 110 V 120 V setting: 108 V to 132 V 220 V setting: 198 V to 242 V 240 V setting: 216 V to 264 V
Supply voltage frequency	50 Hz, 60 Hz, or 400 Hz (automatically sensed at power on)
Input and output connections	See Rear-panel overview (on page 3-5).
Environmental conditions	For indoor use only **Altitude:** Maximum 2000 meters (6562 feet) above sea level **Operating:** 0 °C to 50 °C, ≤80 percent relative humidity at 35 °C **Storage:** −40 °C to 70 °C **Pollution degree:** 2

Section 2

Installation

In this section:

Instrument power

The 2182A operates from a line voltage of 100 V ac, 120 V ac, 220 V ac, or 240 V ac at line frequencies of 45 Hz to 66 Hz or 360 Hz to 440 Hz.

On power-up, the 2182A detects the line-power frequency and automatically selects the correct frequency setting. You can check the line-power frequency setting by sending the line frequency query:

```
:SYSTem:LFRequency?
```

The response message is `50` or `60`. The `50` indicates that the line frequency is set for 50 Hz or 400 Hz. `60` indicates that it is set for 60 Hz.

CAUTION

Operating the instrument on an incorrect line voltage may cause damage to the instrument, possibly voiding the warranty.

NOTE

You must turn on the 2182A and allow it to warm up for at least 2½ hours to achieve rated accuracies.

WARNING

The power cord supplied with the 2182A contains a separate protective earth (safety ground) wire for use with grounded outlets. When proper connections are made, the instrument chassis is connected to power-line ground through the ground wire in the power cord. In the event of a failure, not using a properly grounded protective earth and grounded outlet may result in personal injury or death due to electric shock.

Do not replace detachable mains supply cords with inadequately rated cords. Failure to use properly rated cords may result in personal injury or death due to electric shock.

Changing the line voltage

In most cases, the line voltage is set for the area to which the instrument is shipped and does not need to be changed.

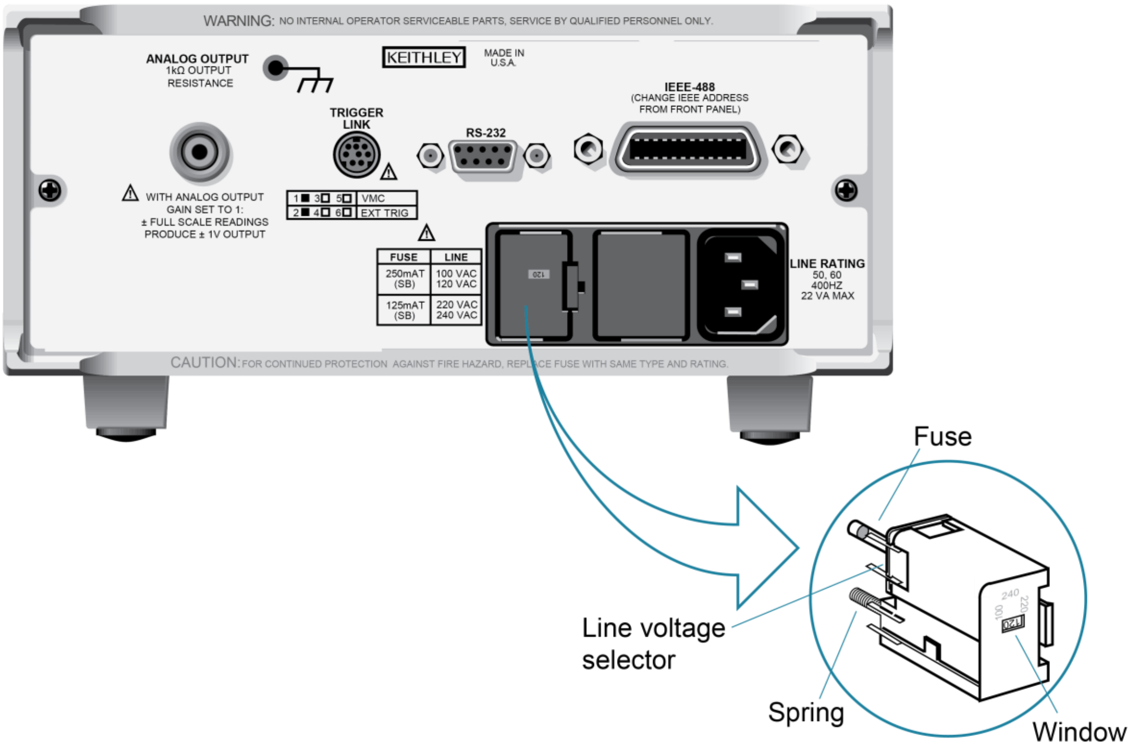
To verify the line voltage setting, look at the window of the fuse holder assembly, as shown in the following figure.

If you need to change the line voltage, use the following instructions.

To change the line voltage:

1. Power off the instrument.
2. Remove all test leads connected to the instrument.
3. Remove the line cord.
4. Place the tip of a flat-blade screwdriver into the power module by the fuse holder assembly.
5. Gently push in and move to the left. Release pressure on the assembly. Its internal spring pushes it out of the power module, as shown in the following figure.

Figure 1: Fuse and line voltage selector



Fuse holder assembly

6. Remove the line voltage selector from the assembly and rotate it to the proper position. When the selector is installed into the fuse holder assembly, the correct line voltage appears in the window.
7. Install the fuse holder assembly into the power module by pushing it in until it locks in place.

Line power connection

To connect the 2182A to line power and turn on the instrument:

1. Check the line voltage setting on the power module. Make sure it is correct for the operating voltage in your area. If not, refer to Changing the line voltage (on page 2-2).
2. Make sure the front-panel power switch is in the off (0) position.
3. Connect the supplied power cord to the ac receptacle on the rear panel.
4. Connect the other end of the power cord to a grounded ac outlet.
5. Turn on the instrument by pressing the front-panel power switch to the on (|) position.

Power-up sequence

On power-up, the 2182A performs self-tests on its EPROM and RAM, and momentarily lights all digit segments and annunciators. If a failure is detected, the instrument momentarily displays an error message and the ERR annunciator turns on. For a list of error messages, refer to Error messages (on page 11-4).

When the instrument passes the self-tests, the firmware revision levels are displayed. For example:

```
REV: A01 A02
```

Where `A01` is the main board ROM revision and `A02` is the display board ROM revision.

After the power-up sequence, the instrument begins to display readings.

Remote communications interfaces

The 2182A supports the following built-in remote interfaces:

- GPIB (IEEE-488) interface
- RS-232 interface

You can use only one interface at a time. At the factory, the GPIB bus is selected. You can select the interface from the front panel. The interface selection is stored in nonvolatile memory, so it does not change if power was off or after a remote interface reset. When you enable the GPIB interface, the RS-232 interface is disabled. Conversely, when you enable the RS-232 interface, the GPIB interface is disabled.

NOTE

When the interface is changed, all data in the buffer clears.

Programming languages

The 2182A supports the Standard Commands for Programmable Instrument (SCPI) programming language or the 182 programming language.

SCPI is fully supported by the GPIB and RS-232 interfaces. Always calibrate the 2182A using the SCPI language.

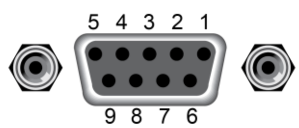
The 182 language is only available with GPIB communications. It implements most commands (DDCs) that are available in the Keithley Model 182 Sensitive Digital Voltmeter. The commands and the programming limitations are provided in Model 182 emulation commands (on page 17-1).

RS-232 connections

The RS-232 serial port is connected to the serial port of a computer using a straight-through RS-232 cable terminated with DB-9 connectors. The serial port uses the transmit (TXD), receive (RXD), and signal ground (GND) lines of the RS-232 standard. It does not use the hardware handshaking lines CTS and RTS.

If your computer uses a DB-25 connector for the RS-232 interface, you need a cable or adapter with a DB-25 connector on one end and a DB-9 connector on the other, wired straight through. Do not use a null modem cable. The following figure shows the pins and the following tables provide pinout identification for the 9-pin (DB-9) connector or 25-pin (DB-25) serial port connector on the computer.

Figure 2: RS-232 interface connector



Pinouts for the RS-232 connector	
Pin	**Description**
1	No connection
2	TXD, transmit data
3	RXD, receive data
4	No connection
5	GND, signal ground
6	No connection
7	RTS, ready to send
8	CTS, clear to send
9	No connection

Pinouts for the DB-25 computer serial port		
Signal	**DB-9 pin number**	**DB-25 pin number**
DCD, data carrier detect	1	8
RXD, receive data	2	3
TXD, transmit data	3	2
DTR, data terminal ready	4	20
GND, signal ground	5	7
DSR, data set ready	6	6
RTS, request to send	7	4
CTS, clear to send	8	5
RI, ring indicator	9	22

The following figure shows the rear-panel connections for the RS-232 interface.

Figure 3: RS-232 connections

RS-232 interface selection and configuration

The RS-232 interface is selected and configured from the RS-232 menu. From this menu, you can enable or disable the RS-232 interface, and check or change the following settings:

- Baud rate (192,000, 9600, 4800, 2400, 1200, 600, or 300)
- Flow control (none or Xon/Xoff)
- Terminator (CR, LF, CRLF, or LFCR)

The RS-232 interface transfers data using eight data bits, one stop bit, and no parity. Make sure the controller you connect to the 2182A also uses these settings.

Make sure that the terminal that you are connecting to the 2182A can support the baud rate you select. Both the 2182A and the other device must be configured for the same baud rate.

You can break data transmissions by sending a `^C` or `^X` character string to the controller. This clears any pending operation and discards any pending output.

NOTE

For the RS-232 interface, you can only use the SCPI language to program the instrument. The instrument automatically switches to the SCPI language when the RS-232 interface is enabled.

NOTE

To retain a present RS-232 setting, press ENTER with the setting displayed. You can exit from the menu structure at any time by pressing EXIT.

To select and configure the RS-232 interface:

1. Press **SHIFT** and then **RS232** to access the RS-232 menu. The present state (ON or OFF) of the RS-232 is displayed.
2. To enable the RS-232 interface, place the cursor on the on/off selection by pressing the ► key.
3. Press the ▲ or ▼ key to toggle the selection to **ON**.
4. Press **ENTER**. The present baud rate is displayed.
5. To change baud rate, place the cursor on the baud rate value.
6. Use the ▲ and ▼ keys to display the correct baud rate value.
7. Press **ENTER**. The present flow control setting is displayed.
8. To change flow control setting, place the cursor on the present flow control selection.
9. Press the ▲ or ▼ key to toggle the selection.
10. Press **ENTER**. The present terminator is displayed.
11. To change the terminator, place the cursor on the present terminator selection.
12. Press the ▲ or ▼ key to display the correct terminator. You can select:
    - **LF:** Line feed
    - **CR:** Carriage return
    - **LFCR:** Line feed, carriage return
    - **CRLF:** Carriage return, line feed
13. Press **ENTER**. The instrument returns to the normal display.

Flow control (signal handshaking)

Signal handshaking between the controller and the instrument allows the two devices to indicate being ready or not ready to receive data. The 2182A does not support hardware flow control.

Software flow control is in the form of `X_ON` and `X_OFF` characters and is enabled when `XonXoFF` is selected from the RS232 FLOW menu. When the input queue of the 2182A becomes more than 3/4 full, the instrument issues an `X_OFF` command. The control program responds to this and stops sending characters until the 2182A issues the `X_ON`, which it does when its input buffer has dropped below half-full.

The 2182A recognizes `X_ON` and `X_OFF` sent from the controller. An `X_OFF` causes the 2182A to stop outputting characters until it sees an `X_ON`. Incoming commands are processed after the `<CR>` character is received from the controller.

If `NONE` is the selected flow control, there is no signal handshaking between the controller and the 2182A. Data will be lost if it is transmitted before the receiving device is ready.

NOTE

For RS-232 operation, use *OPC or *OPC? with commands that respond slowly. Refer to *OPC (on page 14-7) for detail on the command and a list of the slowest responding commands.

GPIB setup

The following topics contain information about GPIB standards, bus connections, and primary address selection.

The 2182A GPIB interface is IEEE Std 488.1 compliant and supports IEEE Std 488.2 common commands and status model topology.

You can have up to 15 devices connected to a GPIB interface, including the controller. The maximum cable length is the lesser of either:

- The number of devices multiplied by 2 m (6.5 ft)
- 20 m (65.6 ft)

You may see erratic bus operation if you ignore these limits.

Install the GPIB driver software

Check the documentation for your GPIB controller for information about where to acquire drivers. Keithley Instruments also recommends that you check with the manufacturer of the GPIB controller for the latest version of drivers or software.

It is important that you install the drivers before you connect the hardware. This prevents associating the incorrect driver to the hardware.

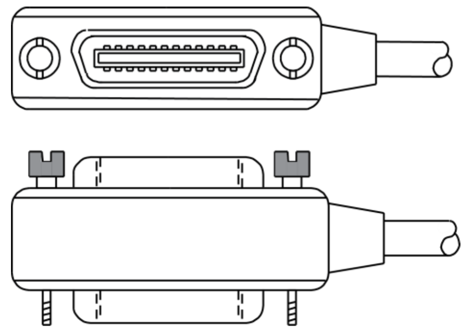
Install the GPIB cards in your computer

Refer to the documentation from the GPIB controller manufacturer for information about installing the GPIB controllers.

GPIB bus connections

To connect a 2182A to the GPIB interface, use a cable equipped with standard GPIB connectors, as shown in the following figure.

Figure 4: GPIB connector



To allow many parallel connections to one instrument, stack the connectors. Each connector has two screws on it to ensure that connections remain secure. The following figure shows a typical connection diagram for a test system with multiple instruments.

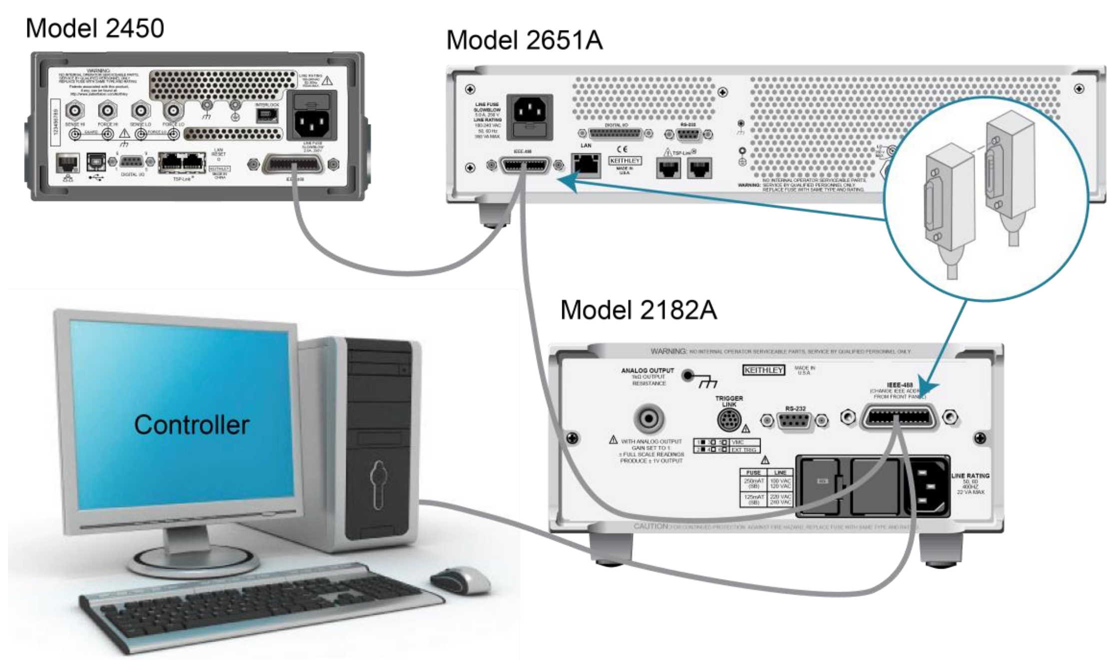
NOTE

The 2182A GPIB connector has metric threads. Do not use GPIB connectors with non-metric screws.

CAUTION

To avoid possible mechanical damage, stack no more than three connectors on any one instrument.

Figure 5: Typical connection diagram for a test system with multiple instruments



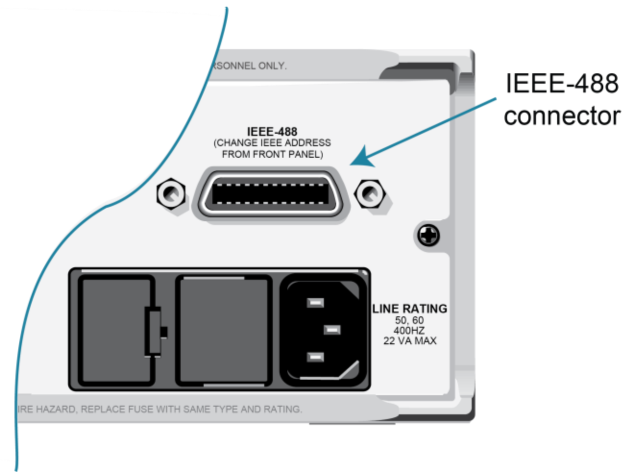
NOTE

To minimize interference caused by electromagnetic radiation, use only shielded IEEE-488 cables, such as the Keithley Instruments Model 7007-1 Shielded GPIB Cable.

To connect the 2182A to the IEEE-488 bus:

1. Line up the cable connector with the connector on the rear panel. The rear-panel connection is shown in the following figure.

Figure 6: IEEE-488 connector location



2. Tighten the screws securely, making sure not to overtighten them.
3. Make sure that the other end of the cable is properly connected to the controller. See the instruction manual for your controller for information about properly connecting to the IEEE-488 bus.

GPIB interface selection and configuration

The GPIB interface is selected and configured from the GPIB menu structure. From this menu, you can enable or disable the GPIB interface, and check or change the following settings:

- Primary address (0 to 30)
- Language (SCPI or 182)

The default GPIB address is 07. You can set the address from 1 to 30 if it is unique in the system. This address cannot conflict with an address that is assigned to another instrument or to the GPIB controller.

NOTE

GPIB controllers are usually set to 0 or 21. To be safe, do not configure any instrument to have an address of 21.

The instrument saves the address in nonvolatile memory. It does not change when you send a reset command or when you turn the power off and on again.

You can use either the SCPI or 182 language to program the instrument when using a GPIB connection.

NOTE

To retain a present GPIB setting, press ENTER with the setting displayed. You can exit from the menu structure at any time by pressing EXIT.

To enable the GPIB interface, change the GPIB address, and change the programming language:

1. Press **SHIFT** and then **GPIB** to access the GPIB menu. The present state (on or off) of the GPIB is displayed.
2. Place the cursor on the on/off selection by pressing the ► key.
3. Press the ▲ or ▼ key to toggle the selection to ON.
4. Press **ENTER**. The present GPIB address is displayed.
5. Use the ◄, ►, ▲, and ▼ keys to display a valid address value.
6. Press **ENTER**. The present programming language selection is displayed.
7. Place the cursor on the present language selection.
8. Press the ▲ and ▼ key to toggle the selection.
9. Press **ENTER**. The instrument returns to the normal display state.

Front-panel GPIB operation

The following paragraphs describe aspects of the front panel that are part of GPIB operation, including messages, status indicators, and the LOCAL key.

GPIB status indicators

The REM (remote), TALK (talk), LSTN (listen), and SRQ (service request) annunciators show the GPIB bus status.

The REM indicator is on when the instrument is in the remote state. The instrument must be addressed to listen with REM true before the REM indicator turns on. When the instrument is in the remote state, all front-panel keys except for the LOCAL key are locked out. When REM is turned off, the instrument is in the local state, and front-panel operation is restored.

The TALK indicator is on when the instrument is in the talker active state. To place the instrument in the talk state, address it to talk with the correct MTA (My Talk Address) command. TALK is off when the instrument is in the talker idle state. To place the instrument in the talker idle state, send an UNT (Untalk) command, address it to listen, or send the IFC (Interface Clear) command.

The LSTN indicator is on when the 2182A is in the listener active state, which is activated by addressing the instrument to listen with the correct MLA (My Listen Address) command. LSTN is off when the instrument is in the listener idle state. To place the instrument in the listener idle state, send UNL (Unlisten), address it to talk, or send the IFC (Interface Clear) command over the bus.

When the SRQ indicator is on, a service request was generated. You can program the instrument to generate a service request (SRQ) when one or more errors or conditions occur. This indicator stays on until the serial poll byte is read or all the conditions that caused the SRQ have ended.

LOCAL key

The LOCAL key cancels the remote state and restores local operation of the instrument. Pressing the LOCAL key also turns off the REM indicator and returns the display to normal if a user-defined message was displayed.

If the `LLO` (Local Lockout) command is in effect, the LOCAL key is inoperative. Refer to LLO (on page 2-15) for additional information.

General bus commands

General commands are commands that have the same general meaning, regardless of the instrument. The following table lists the general bus commands.

Command	Effect on 2182A
DCL	Device clear. Returns the 2182A and all devices on the GPIB to known conditions. See DCL (on page 2-13) for details.
GET	Group execute trigger. Initiates a trigger. See GET (on page 2-13) for details.
GTL	Go to local. Cancel remote; restore 2182A front-panel operation. See GTL (on page 2-14) for details.
IFC	Interface clear. Goes into talker and listener idle states. See IFC (on page 2-14) for details.
LLO	Local lockout. LOCAL key locked out. See LLO (on page 2-15) for details.
REN	Remote enable. Goes into remote operation when next addressed to listen. See REN (on page 2-15) for details.
SDC	Selective device clear. Returns the 2182A to known conditions. See SDC (on page 2-15) for details.
SPE, SPD	Serial polling. Serial polls the 2182A. See SPE, SPD (on page 2-16) for details.

DCL (device clear)

Use the device clear (DCL) command to clear the GPIB interface and return it to a known state. The DCL command is not an addressed command, so all instruments equipped to implement DCL are returned to a known state simultaneously.

When the 2182A receives a DCL command, it:

- Clears the input buffer, output queue, and command queue
- Cancels deferred commands
- Clears any command that prevents the processing of any other device command

The DCL command does not affect instrument settings and stored data.

Program fragment

```
CALL TRANSMIT ("DCL", status%) ' Clears all devices.
```

GET (group execute trigger)

The group execute trigger (GET) command is a GPIB trigger that triggers the instrument to make readings from a remote interface. The 2182A reacts to this trigger if it is the programmed control source. The control source is programmed from the SCPI TRIGger subsystem.

With the instrument programmed and waiting for a GPIB trigger, the following program fragment provides the GET.

Program fragment

```
CALL TRANSMIT ("UNL LISTEN 7 GET", status%)   ' Trigger 2182A over
        the bus.
```

When the command is executed, the trigger event for the 2182A occurs. Any other listeners ignore the trigger.

GTL (go to local)

Use the go to local (GTL) command to put an instrument that is in remote mode instrument into local mode. Leaving the remote state also restores operation of all front-panel controls. The following program fragment demonstrates how to use GTL.

Program fragment

```
CALL TRANSMIT ("UNL LISTEN 7 GTL", status%)' Place 2182A in local mode.
```

IFC (interface clear)

The controller sends the interface clear (IFC) command to place the 2182A in the talker idle state and the listener idle state. The instrument responds to the IFC command by canceling illumination of the front-panel TALK or LSTN lights if the instrument was previously placed in one of these states.

Transfer of command messages to the instrument and transfer of response messages from the instrument are not interrupted by the IFC command. If transfer of a response message from the instrument was suspended by IFC, transfer of the message resumes when the instrument is addressed to talk. If transfer of a command message to the instrument was suspended by the IFC command, the rest of the message can be sent when the instrument is addressed to listen.

This command does not affect the status of the instrument. Settings, data, and event registers are not changed.

To send the IFC command, the controller needs to set the IFC line true for a minimum of 100 µs.

The initialize routine (`CALL INITIALIZE`) uses this command internally. This command does not affect the status of the instrument. Settings, data, and event registers are not changed.

To send the IFC command, the controller must set the IFC line true for a minimum of 100.

Program fragment

```
CALL INITIALIZE (21, 0)   ' Initialize GPIB system (sends IFC) and set
   ' interface card address to 21.
```

LLO (local lockout)

Use the LLO command to prevent local operation of the instrument. After the 2182A receives LLO, all its front-panel controls except the POWER switch are inoperative. In this state, pressing LOCAL does not restore control to the front panel. The GTL command restores control to the front panel.

Program fragment

```
CALL TRANSMIT ("UNL LISTEN 7 LLO", status%)' Lock out front panel.
CALL TRANSMIT ("UNL LISTEN 7 GTL", status%)' Restore front panel.
```

REN (remote enable)

When the instrument detects the remote enable (REN) event, it is set up for remote operation. The instrument is not placed in remote mode when it detects the REN event. The instrument must be addressed to listen after the REN event before it goes into remote mode.

You should place the instrument into remote mode before you attempt to program it using a remote interface.

The instrument does not have to be in remote to be a talker.

NOTE

All front-panel controls except for LOCAL and POWER are inoperative while the instrument is in remote. To restore normal front-panel operation, press the LOCAL key.

Program fragment

```
CALL TRANSMIT ("MTA LISTEN 7 REN", status%)   ' Place 2182A in remote;
      ' turn on REM annunciator.
```

SDC (selective device clear)

The selective device clear (SDC) command is an addressed command that performs essentially the same function as the device clear (DCL) command. However, because each device must be individually addressed, the SDC command provides a method to clear only selected instruments, instead of clearing all instruments simultaneously with the DCL command.

When the 2182A receives an SDC command, it:

- Clears the input buffer, output queue, and command queue
- Cancels deferred commands
- Clears any command that prevents the processing of any other device command

An SDC call does not affect instrument settings and stored data.

Program fragment

```
CALL TRANSMIT ("UNL LISTEN 7 SDC", status%) ' Clears 2182A.
```

SPE, SPD (serial polling)

Use the serial polling sequence to obtain the 2182A serial poll byte. The serial poll byte contains important information about internal functions. Generally, the serial polling sequence is used by the controller to determine which of several instruments has requested service with the SRQ line. The serial polling sequence may be performed at any time to obtain the status byte from the 2182A. For more information, see Status Byte Register overview (on page 16-2).

Program fragment

```
WaitSRQ:
  CALL SPOLL (7, poll%, status%)           ' Serial poll the 2182A.
  IF (poll% AND 64) = 0 THEN GOTO WaitSRQ  ' Loop back if no SRQ.
```

Transmit routines

A transmit routine sends general bus commands. It contains a series of GPIB commands to be carried out. In addition to the general bus commands, there are other commands used in the transmit command string. For detail on all the commands, refer to the documentation for the interface card. Some of the more frequently used commands are:

- **UNL:** Unlisten disables any listeners that may exist.
- **UNT:** Untalk disables the current talker, if any.
- **LISTEN 7:** Listen assigns the device at address 7 (2182A) to be a listener.
- **MTA:** My Talk Address assigns the computer as the talker.

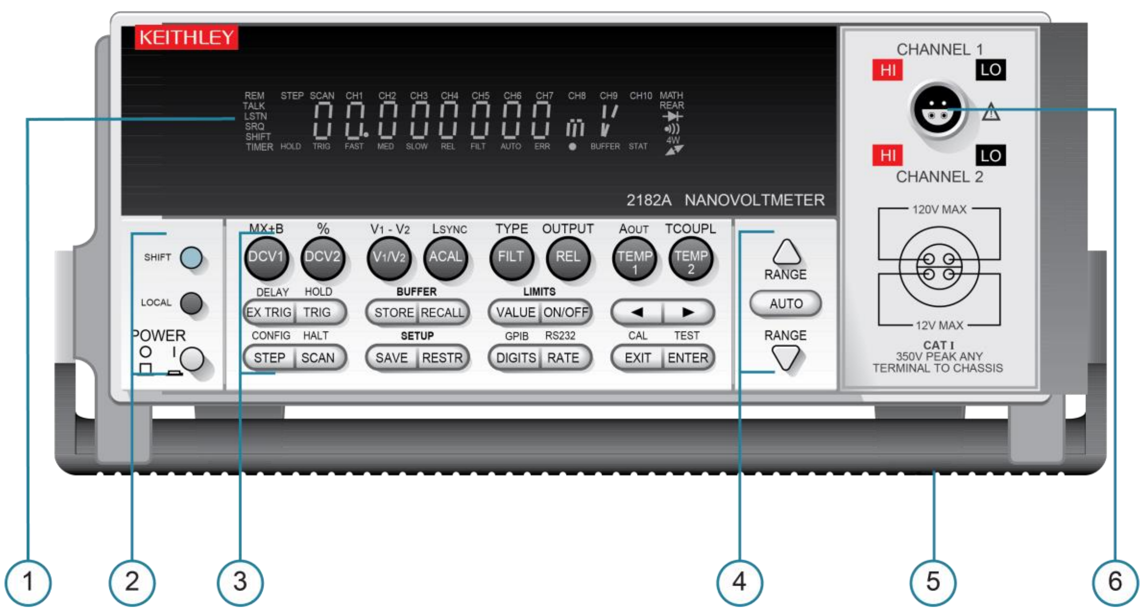
Instrument description

In this section:

Front-panel overview

The front panel of the 2182A is shown in the following figure.

Figure 7: Model 2182A front panel



Most keys perform two functions. The label on a key indicates its normal function, which is selected by pressing the key. The label above a key indicates its shifted function. A shifted function is selected by pressing the SHIFT key and then the function key.

1 Display annunciators

The display of the 2182A is primarily used to display readings with the units and type of measurement. Annunciators are at the top, bottom, left, and right of the reading or display message. The annunciators indicate various states of operation, as described in the following table.

*	Readings are being stored in the buffer
	Additional selections are available
•)))	Beeper is on for limit testing
AUTO	Autorange is enabled
BUFFER	Readings that are stored in the buffer are being recalled
CH1	Channel 1 input is displayed
CH2	Channel 2 input is displayed
CH1 and CH2	Ratio (V1/V2) reading is displayed
ERR	Questionable reading or invalid calibration step
FAST	Fast (0.1 PLC) reading rate is selected
FILT	Filter is enabled
HOLD	Instrument is in hold mode
LSTN	Instrument is addressed to listen over GPIB
MATH	Calculation (mx+b or percent (%)) is enabled
MED	Medium (1 PLC) reading rate is selected
REAR	Analog output is on
REL	Relative offset is enabled for the present measurement function
REM	Instrument is in GPIB remote mode
SCAN	Scan mode is selected
SHIFT	SHIFT is selected
SLOW	Slow (5 PLC) reading rate is selected
SRQ	Service request over GPIB
STAT	Buffer statistics are displayed
STEP	Step mode is selected
TALK	Instrument addressed to talk over GPIB bus
TIMER	Timer controlled scans are in use
TRIG	External triggering (front panel, bus, or Trigger Link) is selected

2 SHIFT key, LOCAL key, and POWER switch

SHIFT	Use to select a shifted function or operation
LOCAL	Cancels GPIB remote mode Controls keyclicks; press SHIFT then LOCAL to enable or disable keyclicks
POWER	Power switch; when pressed in, 2182A is on; when out, 2182A is off

3 Function keys

Top row of keys

Unshifted	
DCV1	Selects the Channel 1 voltage measurement function
DCV2	Selects the Channel 2 voltage measurement function
V_1/V_2	Selects ratio (Channel 1 voltage reading / Channel 2 voltage reading)
ACAL	Selects automatic gain calibration
FILT	Enables or disables the filter for the selected measurement function
REL	Enables or disables the relative offset for the selected measurement function
TEMP 1	Selects the Channel 1 temperature measurement function; when the internal sensor is selected, TEMP 1 measures and displays the internal temperature of the 2182A
TEMP 2	Selects the Channel 2 temperature measurement function; when the internal sensor is selected, TEMP 2 measures and displays the internal temperature of the 2182A
Shifted	
MX+B	Multiplies a scale factor (M) to the reading (X) and then adds an offset (B)
%	Calculates the percent deviation from a specified reference
$V_1 - V_2$	Selects the delta; $(V_1t1 - V_1t2)/2$
LSYNC	Enables or disables line cycle synchronization; when enabled, noise induced by the power line is reduced at the expense of speed
TYPE	Selects the filter (analog or digital) and configures the digital filter (window, count and type)
OUTPUT	Enables or disables the relative offset for the analog output
AOUT	Enables or disables the analog output
TCOUPL	Configure the temperature measurement (units, junction type, thermocouple type, and sensor type)

Middle row of keys

Unshifted	
EX TRIG	Selects an external trigger source (front panel, bus, or Trigger Link)
TRIG	Triggers a measurement from the front panel
STORE	Sets the reading count for the buffer and enables the buffer
RECALL	Displays the stored readings (including maximum, minimum, peak-to-peak, average, and standard deviation) The ▲ and ▼ range keys scroll through the buffer and the ◄ | ► key toggles between reading number and reading
VALUE	Sets the upper and lower limits for limit testing
ON/OFF	Enables or disables limit testing and selects the beeper mode for limit testing
◄ | ►	Controls the cursor position for making selections or editing values
Shifted	
DELAY	Sets the user delay between the trigger and the measurement
HOLD	Holds reading when the specified number of samples is in the selected tolerance

Bottom row of keys

Unshifted	
STEP	Steps through the channels; sends a trigger after each channel
SCAN	Scans through the channels; sends a trigger after last channel
SAVE	Saves the present configuration for the power-on user default
RESTR	Restores the factory or user default configuration
DIGITS	Changes the number of digits of reading resolution
RATE	Changes the reading rate (the number of power line cycles (PLC))
EXIT	Cancels the selection or returns to the measurement display
ENTER	Accepts the selection, moves to the next choice, or returns to the measurement display
Shifted	
CONFIG	Configures a scan (type, timer, channel count, and reading count)
HALT	Turns off step/scan operation
GPIB	Enables or disables the GPIB interface, sets the GPIB address, and selects the language
RS232	Enables or disables the RS-232 interface and selects the baud rate, flow control, and terminator
CAL	Accesses calibration
TEST	Tests the display annunciators, display digit segments, and front-panel keys. For more detail on these options, refer to "Front panel tests" in the *Model 2182A Service Manual*.

4 Range keys

▲	Selects the next higher voltage measurement range
▼	Selects the next lower voltage measurement range
AUTO	Enables or disables autorange

5 Handle

Pull out and rotate to new position.

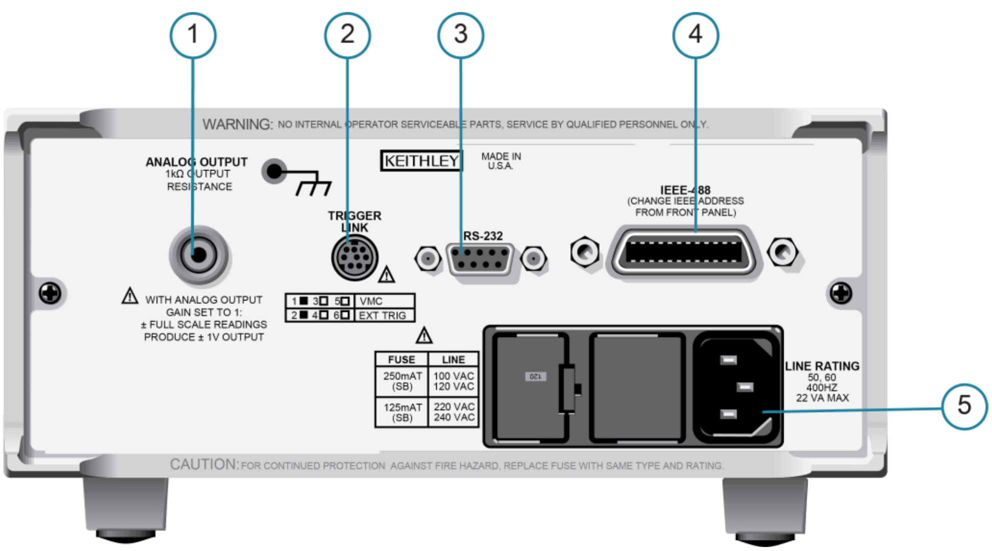
6 Input connector

CHANNEL 1	Measure voltage or temperature; voltage ranges are 10 mV, 100 mV, 1 V, 10 V, and 100 V
CHANNEL 2	Measure voltage or temperature; voltage ranges are 100 mV, 1 V, and 10 V

Rear-panel overview

The rear panel of the 2182A is shown in the following figure.

Figure 8: 2182A rear panel



1 ANALOG OUTPUT

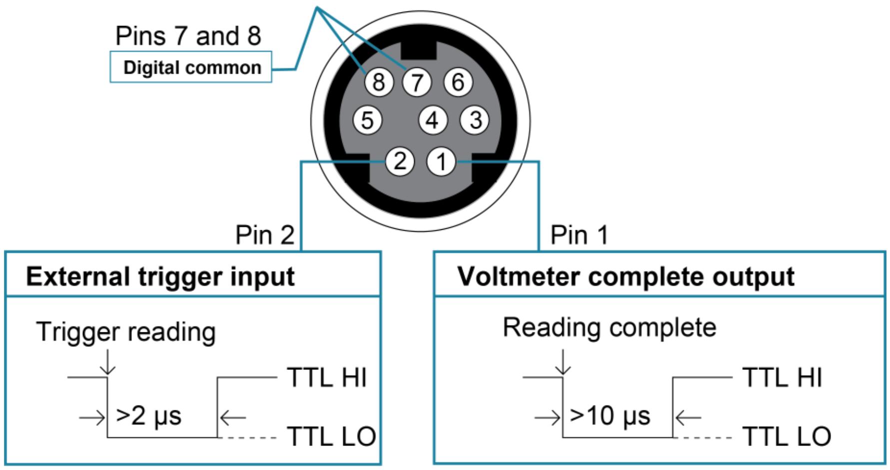
The analog output provides a scaled non-inverting dc voltage that is typically used to drive a chart recorder. With analog output gain set to one, a full range input results in a 1.2 V analog output. For additional detail on using the analog output, refer to Analog output overview (on page 4-34).

2 TRIGGER LINK

An eight-pin micro-DIN connector for sending and receiving trigger pulses among connected instruments. Use a trigger link cable or adapter, such as Keithley Models 8501-1, 8502, or 8503.

The pinouts for the TRIGGER LINK connector are shown in the following figure.

Figure 9: TRIGGER LINK pinouts



3 RS-232

Connector for RS-232 operation. Use a straight-through (not null modem) DB-9 shielded cable.

4 IEEE-488

Connector for IEEE-488 (GPIB) operation. Use a shielded cable, such as the Model 7007-1.

5 Power module

Contains the ac line receptacle, power-line fuse, and the line-voltage setting. You can configure the instrument for line voltages of 100 V ac, 120 V ac, 220 V ac, or 240 V ac at line frequencies of 45 Hz to 66 Hz or 360 Hz to 440 Hz. If you need to change the line voltage, refer to Changing the line voltage (on page 2-2).

Error and status messages

See Status and error messages (on page 11-4) for a list of error and status messages.

You can program the instrument to generate an SRQ and send command queries to check for specific error conditions. Refer to Program examples (on page 15-1).

Default settings

The default setup configurations are factory and user. As shipped from the factory, the 2182A powers up to the factory default settings listed in the following table. You can also set the 2182A to power up to a user default setup. The user default setup contains the last configuration you saved. The SAVE key saves the present configuration as the user default setup. The RESTR key restores the instrument to the factory defaults or the user-saved defaults.

NOTE

To assure that the proper filter state is recalled, set the analog and digital filters before saving the user setup. Refer to Filters (on page 4-20) for more information.

To save the present setup as the power-on default:

1. Configure the instrument for your measurement application.
2. Press the **SAVE** key.
3. Press the ▼ and ▲ keys to display **YES** or **NO**.
4. Press the **ENTER** key to store the settings.

To restore factory or user settings:

1. Press the **RESTR** key.
2. Press the ▼ and ▲ keys to select **FACT** (factory) or **USER**.
3. Press the **ENTER** key to restore the settings.

Default settings

Setting	**Factory Default**
Measure function	Voltage
Analog output	On
Gain (M)	1.0
Offset (B)	0
Relative (REL)	Off
Front Autozero	On
Autozero	On
LSYNC	Off
Buffer	No effect
Delta	Off
Function	DCV1
GPIB	No effect (On at factory)
Address	No effect (7 at factory)
Language	No effect (SCPI at factory)
Key click	On

Default settings

Setting	Factory Default
Limits	Off
Beeper	Never
High limit 1	+1
Low limit 1	−1
High limit 2	+2
Low limit 2	−2
mx+b	Off
Scale factor (M)	1.0
Offset (B)	0.0
Percent (%)	Off
Reference	1.0
Ratio (V1/V2)	Off
RS-232	Off
Baud rate	No effect
Flow control	No effect
Terminator (Tx)	No effect
Scanning	Off
Type	Internal
Timer	Off
Channel 1 count	1
Reading count	2
TEMP 1 and TEMP 2	
Digits	6
Filter	On
Analog filter	Off
Digital filter	On
Count	10
Mode	Moving average
Window	0.01%
Rate	5 PLC (Slow)
Reference junction	Internal
Relative (REL)	Off
Sensor	Thermocouple
Thermocouple type	Type J
Units	C
Triggers	
Continuous	On
Delay	Auto
Control Source	Immediate
DCV1 and DCV2	
Digits	7.5

Default settings

Setting	Factory Default
Filter	On
Analog filter	Off
Digital filter	On
Count	10
Mode	Moving average
Window	0.01%
Hold	Off
Count	5
Window	1%
Range	Auto
Rate	5 PLC (Slow)
Relative (REL)	Off

Reset the instrument

You can reset many of the instrument settings to their default values. For detail on what gets reset, see Default settings (on page 3-7) and the descriptions in SCPI command reference (on page 13-1).

To reset the instrument using a remote interface, send the command `*RST`.

Setting the number of displayed digits

You can change the number of digits that are displayed for voltage and temperature measurement readings on the display. For voltage readings, you can display 3½, 4½, 5½, 6½, or 7½ digits. For temperature readings, you can display 4 to 7 digits.

The number of displayed digits does not affect accuracy or speed of the measurements. It also does not affect the format of readings that are returned from a remote command.

NOTE

The digits setting for a voltage function applies to the other voltage function. For example, if you set DCV1 for 5½ digits, DCV2 is also set for 5½ digits. Similarly, the digits setting for a temperature function applies to the other temperature function. Setting TEMP 1 for 6 digits also sets TEMP 2 for 6 digits.

To set the number of displayed digits from the front panel:

1. Press the **DCV1**, **DCV2**, **TEMP 1**, or **TEMP 2** key.
2. Press the **DIGITS** key until the correct number of digits is displayed.

This setting takes effect the next time you make a measurement.

From a remote interface:

Refer to :SENSe:<function>:DIGits (on page 13-41).

Section 4

Making measurements

In this section:

Measurement overview

The 2182A provides two input channels for dc voltage and temperature measurements, as shown in the following table.

Measurement	Input Channels to Use
Voltage	Channel 1
Temperature	Channel 1
Voltage and Voltage	Channel 1 and Channel 2
Voltage and Temperature	Channel 1 and Channel 2

Channel 1 is used as the fundamental measurement channel, while Channel 2 provides sense measurements. Because of this operational relationship between the two channels, Channel 2 cannot be used as an independent, stand-alone measurement channel. Its inputs must be referenced to Channel 1 LO.

Using the front-panel display

Guidelines for making menu selections:

- The display shows the presently selected option with each menu item. For example, if °C is the present units option, then `UNITS: C` is displayed.
- Blinking characters indicate cursor position. The cursor can be on a menu item name (for example, `UNITS` is blinking) or on a menu item option (for example, `C` is blinking). Cursor position is controlled by the ◄ and ► keys.
- With the cursor on a menu item name, you can use the ▲ or ▼ key to scroll through the other menu items. Press **ENTER** to select the displayed option and move on to the next menu item or exit if you are at the end of the menu.
- With the cursor on a menu item option, you can use the ▲ or ▼ key to display one of the other options for that menu item. Press **ENTER** to select the displayed option and move to the next menu item or exit if you are at the end of the menu.
- Press **EXIT** to leave the menu and return to the normal display state.

Voltage measurements

The 2182A has the voltage measurement functions DCV1 and DCV2. DCV1 is available for input Channel 1 and DCV2 is available for Channel 2.

DCV1 has five measurement ranges (10 mV, 100 mV, 1 V, 10 V, and 100 V) and can measure voltage from 1 nV to 120 V. DCV2 (Channel 2) has three measurement ranges (100 mV, 1 V, and 10 V) and can measure voltage from 10 nV to 12 V. Accuracy for each channel is listed in the specifications, available at tek.com/keithley.

Temperature measurements

The 2182A has the temperature measurement functions TEMP 1 and TEMP 2. TEMP 1 is available for input Channel 1 and TEMP 2 is available for Channel 2.

Depending on which thermocouple type is used (B, E, J, K, N, R, S, or T), the 2182A can measure temperature from −200 °C to 1820 °C. Refer to the 2182A specifications for the measurement range for each thermocouple type. The specifications are available at tek.com/keithley.

To make accurate temperature measurements, the thermocouple connections (reference junctions) must be maintained at a known temperature. You can use the internal reference junction or an external simulated reference junction.

The internal reference junction of the 2182A is the input connector. A temperature sensor is inside the instrument, adjacent to the input connector. The sensor is measured continuously to maintain accuracy. To use the internal reference junction, the thermocouple wires must be soldered directly to a connector that mates to the input connector. A disadvantage of using the internal reference junction is the connection requirement. You cannot use the supplied 2107 input cable as is. You must modify the cable or use a separate connector.

You can use a simulated reference junction, which is an external apparatus, such as an ice bath, as the reference junction. The thermocouple wires are connected to the copper lugs of the supplied input cable. The connection points are then immersed in the ice bath. The temperature of the ice bath must be entered into the 2182A as the simulated reference temperature.

Warm-up

After the 2182A is turned on, it must be allowed to warm up for at least 2½ hours to allow the internal temperature to stabilize. After the warm-up period, an automatic calibration (ACAL) must be performed if the present internal temperature and the last internal temperature reading differ by more than 1°C.

ACAL (calibration)

Autocalibration (ACAL) is a special front-end gain calibration for the 10 mV and 100 V ranges. It needs to be performed whenever the internal temperature and TCAL vary by more than 1 °C. TCAL is the internal temperature reading at the time of the last ACAL. For example, if ACAL was performed at 28 °C and the internal temperature changes to 29.1 °C, another ACAL is required to maintain the specified accuracy.

When the internal temperature and TCAL differ by more than 1 °C, bit 9 in the Questionable Event Condition Register is set to indicate a questionable ACAL. See Status model (on page 16-1) for more information.

You can perform a full or low-level ACAL. FULL ACAL calibrates the 10 mV and 100 V ranges, while LOW-LVL (low-level) ACAL only calibrates the 10 mV range. If you are not going to use the 100 V range, it is recommended that you perform LOW-LVL ACAL.

FULL ACAL requires that no connectors or cables are connected to the input connector on the front panel of the 2182A. Whenever input connections are broken for an extended time, the contacts must be cleaned before reconnecting. See Cleaning input connectors (on page 11-1).

For LOW-LVL ACAL, you do not need to remove the input cable, break any connections, or remove power.

NOTE

ACAL is a partial calibration that can be performed by the user. For complete instrument calibration, which must be performed by a qualified service technician, refer to the *Model 2182A Service Manual*.

It takes about five minutes to complete low-level ACAL and a little more than five minutes to complete full ACAL. When finished, the instrument returns to the normal display state.

To perform LOW-LVL or FULL ACAL from the front panel:

1. Press the **ACAL** key to access the menu.
2. Use the ▼ or ▲ key to display the type of ACAL (LOW-LVL or FULL).
3. Press **ENTER**. The message `ACAL` is displayed while calibration is in process.

To perform LOW-LVL or FULL ACAL using remote commands:

Refer to the CALibration subsystem (on page 13-26) for command descriptions and examples.

Measuring internal temperature

To measure the internal temperature of the 2182A:

1. Press **SHIFT** and then **TCOUPL** to display the present units designator (C, F, or K) for temperature measurements.
2. To change the units designator, press the ► key to place the blinking cursor on the units designator, and press the ▼ or ▲ key to display the units.
3. Press **ENTER**. The present sensor selection (TCOUPLE or INTERNL) is displayed.
4. To change the sensor selection, press the ► key to place the blinking cursor on **TCOUPLE** and press the ▼ or ▲ key to display **INTERNL**.
5. Press **ENTER** to return to the normal display state.
6. Press **TEMP 1** or **TEMP 2** to measure and display the internal temperature of the 2182A. When the internal temperature is displayed, the CH1 and CH2 annunciators are off.

Checking TCAL temperature

To determine the internal temperature at the time of the last ACAL:

1. Press **SHIFT** and then **CAL** to access the calibration menu.
2. Use ▼ or ▲ key to display `CAL: TEMP`.
3. Press **ENTER**. The temperature (in °C) at the time of the last ACAL is displayed.
4. Use the **EXIT** key to return to the measurement display.

Test connections

WARNING

A hazardous voltage condition exists at or above 42 V_{PEAK}. To prevent electric shock that could result in injury or death, never make or break connections while hazardous voltage is present.

CAUTION

Exceeding the following limits may cause instrument damage that will void the warranty:

- **Channel 1 HI and LO inputs have a maximum measurement capability of 120 V_{PEAK}. These inputs are protected to 150 V_{PEAK} to any terminal or 350 V_{PEAK} to chassis.**
- **Channel 2 HI and LO inputs have a maximum measurement capability of 12 V_{PEAK}. Channel 2 HI is protected to 150 V_{PEAK} to any terminal. Channel 2 LO is protected to 70 V_{PEAK} to Channel 1 LO. Both inputs are protected to 350 V_{PEAK} to chassis.**

As a general rule, use Channel 1 whenever possible to measure voltages below 1 V. If you are using Channel 2 to measure voltages below 1 V and the impedance between Channel 2 LO and Channel 1 LO is ≥100 kΩ, pumpout current may be high enough to corrupt measurements. In this case, you can enable the low charge injection mode to reduce pumpout current. Refer to Pumpout current (low charge injection mode) (on page 4-27) for details.

Connection techniques

Use copper-to-copper connections wherever possible in the test circuit to minimize thermal EMFs that could corrupt measurements.

Any solder connections to your test circuit require the use of silver solder to minimize thermal EMFs.

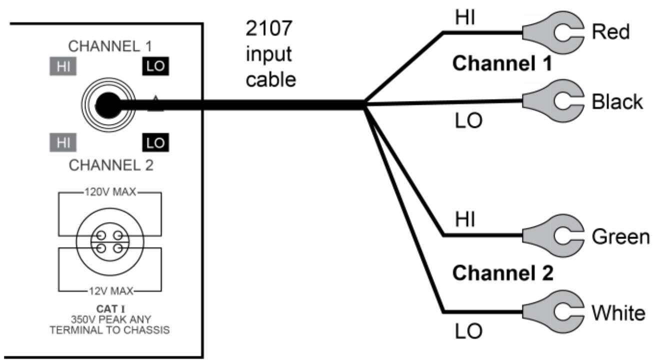
CAUTION

Silver solder has a high melting point. Take care not to damage a CHANNEL connector (or any other device) by applying excessive heat.

Model 2107 input cable

The Model 2107 Input Cable, shown in the following figure, is terminated with a CHANNEL connector on one end and copper lugs on the other end. The cable is shielded to chassis ground when connected to the 2182A. You can use this cable to make voltage measurements and temperature measurements that use an external simulated reference junction.

Figure 10: Model 2107 input cable



To make voltage connections, clamp the cleaned copper lugs of the cable to the cleaned copper connectors of the test circuit. For the test circuit, use clean #10 copper bus wire wherever possible. Clean copper-to-copper connections minimize thermal EMFs, which could corrupt a measurement. See Cleaning test circuit connectors (on page 4-13).

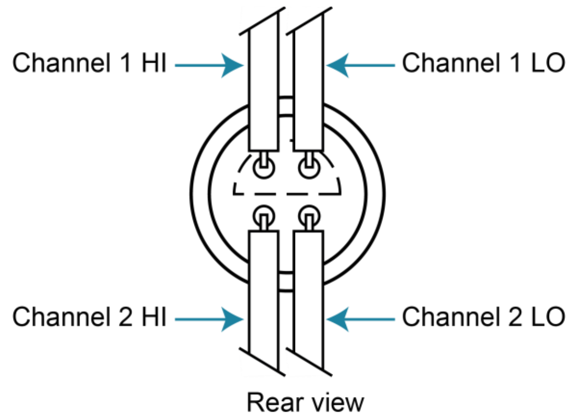
If necessary, you can cut the copper lugs off the Model 2107 and connect the wires directly to your test circuit, as described in Customized connections (on page 4-7).

To make temperature (simulated reference) measurements using an external simulated reference junction, wrap or clamp the thermocouple wires around the copper lugs or bare wires of the input cable.

Customized connections

Temperature measurements using the internal reference junction require that the thermocouple wires be soldered directly to a connector that mates to the input of the 2182A. Use silver solder to minimize thermal EMFs. The following figure shows terminal identification for a connector.

Figure 11: CHANNEL connector - terminal identification



To make these customized connections, you can modify the supplied input cable.

CAUTION

Silver solder has a high melting point. Do not apply excessive heat to the connector. Excessive heat can damage the connector.

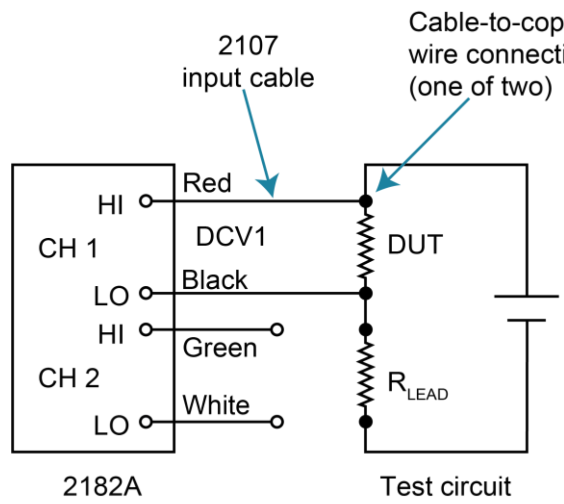
Voltage only connections

You can perform single or dual channel voltage measurements. The dual channel feature of the 2182A allows you to make comparison measurements in a test circuit. The connections for voltage measurements are described in the following topics.

Single channel measurement connections

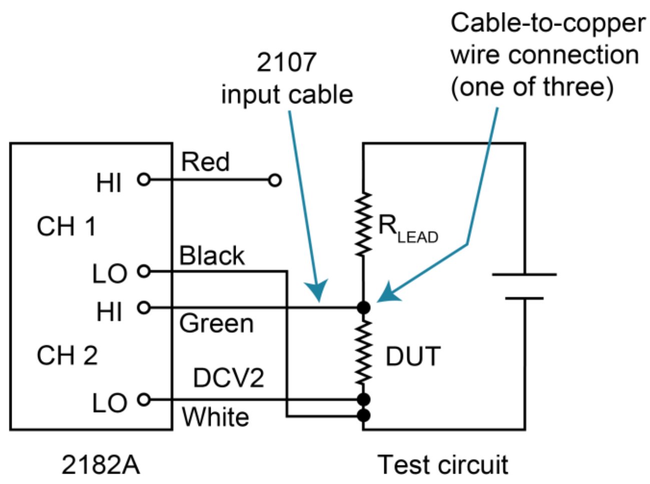
The following figures show typical connections to measure a DUT using a single channel.

Figure 12: Single-channel voltage connections for Channel 1



When using Channel 2, its inputs must be referenced to Channel 1 LO, as shown in the following figure.

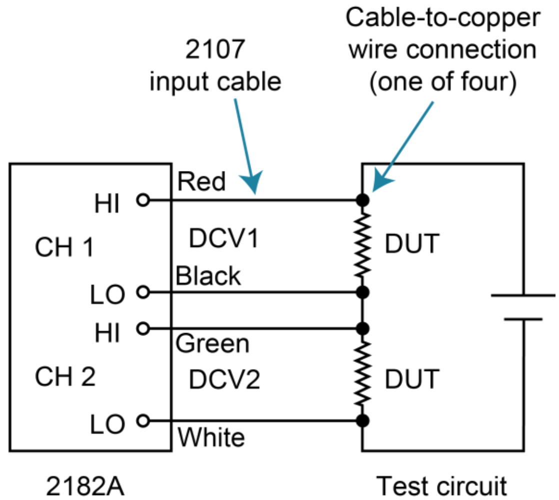
Figure 13: Single-channel voltage connections for Channel 2



Dual-channel measurement connections

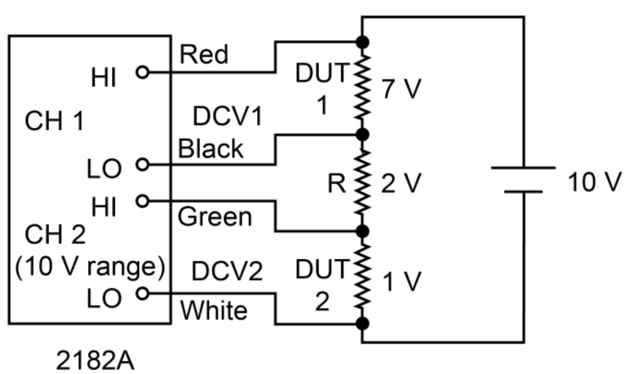
The following figure shows typical connections to make comparison measurements of two devices in a test circuit. For this measurement configuration, there is no voltage differential between the two measurement channels. Channel 2 HI is connected directly to Channel 1 LO.

Figure 14: Typical dual-channel voltage connections



The following figure shows a measurement configuration that has a voltage differential between two channels. The differential is the 2 V drop across R. Channel 1 measures voltage across DUT 1. Channel 2 measures voltage across DUT 2. Internally, the A/D converter references the Channel 2 measurements to Channel 1 LO. For example, if 1 V is being input to Channel 2 and there is a 2 V differential between the two channels, 3 V is applied to the A/D converter. Therefore, if Channel 2 is on the 1 V range, the 3 V applied to the A/D converter causes it to overflow. The 1 V measurement on Channel 2 can only be made on the 10 V range.

Figure 15: Differential dual-channel voltage connections



NOTE

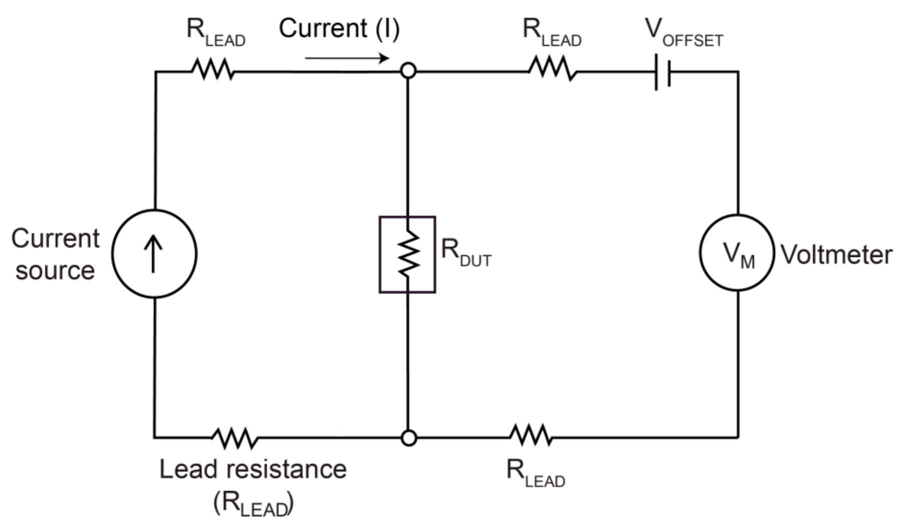
When measuring with a voltage differential between two channels, Channel 2 HI or LO cannot be more than 12 V_{PEAK} from Channel 1 LO.

Channel voltage differential reduces the maximum measurement capability of Channel 2. Normally, Channel 2 can measure up to 12 V. However, a 2 V differential reduces the maximum measurement capability of Channel 2 to 10 V. In the “Typical dual-channel voltage connections” figure above, a >10 V input to Channel 2 causes an overflow condition.

Measurement techniques

Techniques used to measure resistances in the normal range are not generally suitable for making low-resistance measurements because of errors caused by voltage drops across the test leads. To overcome these limitations, low resistance measurements are usually made using the 4-wire (Kelvin) connections shown in the following figure.

Figure 16: 4-wire low-resistance measurement technique



A current source forces the current (I) through an unknown resistance, developing a voltage across that device. Even though the test lead resistance, R_{LEAD}, is present, it does not affect the current through R_{DUT} because I is assumed to be a constant current source with high output impedance. Also, since the voltmeter has a very high input resistance (very low leakage current), the current through the sense leads is negligible and the voltage drop across R_{LEAD} is essentially zero. Thus, the voltage measured by the meter is essentially the same as the voltage across the unknown resistance, R_{DUT}.

Since the current through the measured resistance and the voltage across the device are both known, the value of that resistance can be determined from Ohm's law:

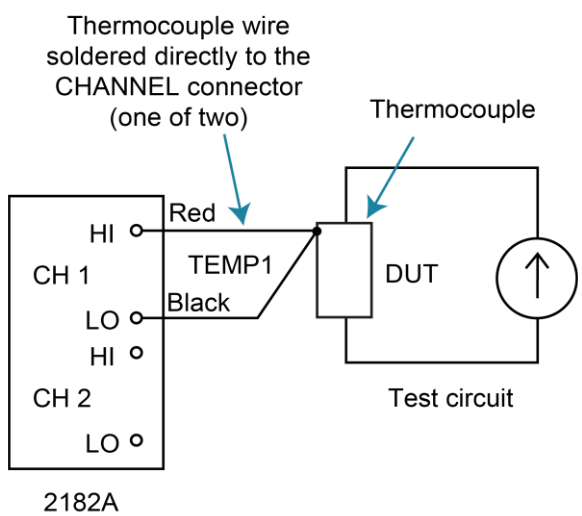
$$R_{DUT} = V_M/I$$

Although the 4-wire measurement method minimizes the effects of lead resistances, other factors can affect low-resistance measurement accuracy. Thermal EMFs and other effects can add an extraneous dc offset voltage (V_{OFFSET}) to the measured voltage. Refer to Minimizing thermal EMFs (on page 5-4) for additional information.

Temperature-only connections

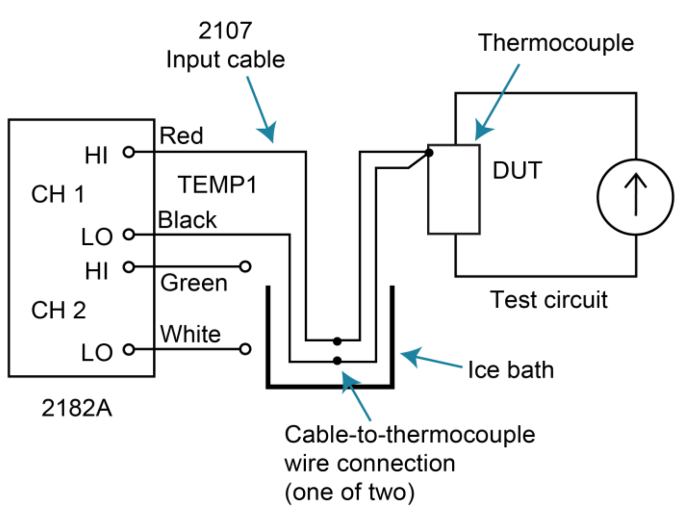
You can use Channel 1 of the 2182A to make temperature measurements. The following figure shows connections using the internal reference junction. The thermocouple wires must be soldered directly to the CHANNEL connector as described in Customized connections (on page 4-7).

Figure 17: Temperature connections (internal reference)



The following figure shows temperature-only connections using an ice bath as a simulated reference junction. The connection points for the input cable and the thermocouple wires are immersed in the ice bath.

Figure 18: Temperature connections (simulated reference)



Voltage and temperature connections

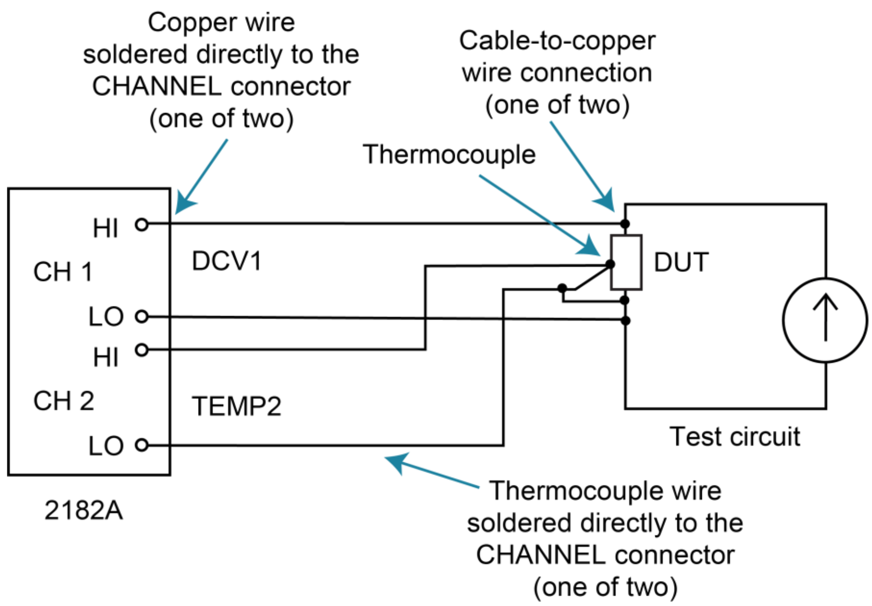
Use Channel 1 for voltage measurements because it supports a wider range of measurements. Use Channel 2 to measure temperature.

A connection example using the internal reference junction for temperature measurements is shown in the following figure. In this example, Channel 1 measures the voltage drop across the DUT and Channel 2 measures the temperature of the DUT.

The jumper wire from the thermocouple to test circuit low is not needed in some test setups, such as:

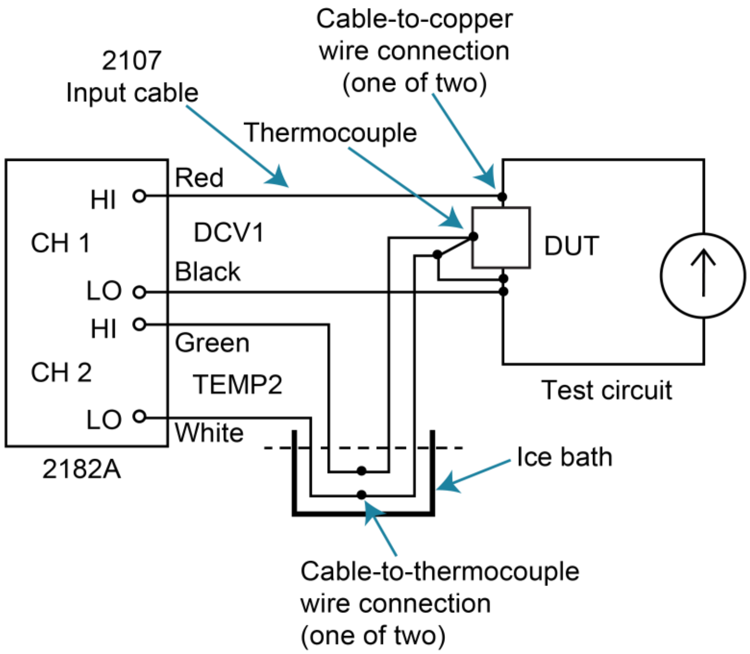
- If the case of the DUT is metal and is connected to test circuit low.
- If there is enough thermal bonding between the DUT and test circuit low. In this setup, you can connect the thermocouple directly to low.

Figure 19: Voltage and temperature connections (internal reference)



The following figure shows the same test with a simulated reference junction (ice bath).

Figure 20: Voltage and temperature connections (simulated reference)



Cleaning test circuit connectors

Wherever possible, use copper-to-copper connections throughout your test circuits to minimize thermal EMFs. However, exposed copper is susceptible to oxidation, which can corrupt the measurement. Make sure that the copper contact surfaces are free of oxidation before making the connection. DeoxIT can be used to clean copper connectors. A small bottle of DeoxIT is supplied with the 2182A.

The Model 2107 Input Cable is terminated with copper lugs.

To clean the copper connectors used in your test circuit:

1. Using a lint-free foam swab (or other applicator), soak up a small amount of DeoxIT.
2. Apply the DeoxIT sparingly to the connectors. Only a thin coating is required.

NOTE

After cleaning, make your test circuit connections in a timely manner to prevent oxidation from forming on exposed connector surfaces.

Measuring voltage and temperature

Any time the internal temperature of the 2182A changes by 1 °C or more, the 10 mV and 100 V ranges need to be calibrated. Refer to ACAL (calibration) (on page 4-3) for details.

Whenever the connector of the Model 2107 Input Cable (or customized cable) is disconnected from the input of the 2182A for a long period, the input connectors must be cleaned to remove oxidation. Refer to Cleaning input connectors (on page 11-1) for details.

Do not use both channels to measure temperature. The electrical connection between the two thermocouples will cause erratic temperature readings.

Clean copper-to-copper connections minimize thermal EMFs. However, when measuring very low voltages, there may still be enough thermal EMFs to corrupt the measurement. In this case, use the relative offset feature of the 2182A to null out that offset. See Minimizing thermal EMFs (on page 5-4) for more information.

Temperature configuration

To make temperature measurements, you must configure the 2182A.

To configure temperature measurements using the front panel:

1. Press **SHIFT** and then **TCOUP** to open the temperature measurement menu.
2. Select **UNITS** to set the temperature units (°C, °F, or K).
3. Select **SENS.**
   - To measure the internal temperature of the 2182A: Select **INTERNL**.
   - To measure temperature at the thermocouple: Select **TCOUPLE**.
4. Select **TYPE** and select the thermocouple type that you are using to measure temperature (B, E, J, K, N, R, S, or T).
5. Select **JUNC** and select the type of reference measurement:
   - To reference measurements to the internal reference junction: Select **INTRNL**.
   - To reference measurements to an external simulated reference: Select **SIM**. You are prompted to enter the simulated reference temperature. Use the arrow keys to display the value and press **ENTER**.

NOTE

The following basic measurement procedure assumes factory defaults. Reset the instrument to the factory default settings when following this step-by-step procedure. Refer to Default settings (on page 3-7) for detail on the defaults.

To measure voltage and temperature:

1. Connect the test circuit to the input of the 2182A. Refer to Test connections (on page 4-5) for details and drawings.
2. For temperature measurements, configure temperature settings as described in Temperature configuration (on page 4-14).
3. Channel 1 is connected to measure voltage, so select **DCV1**.
4. Observe the reading on the display. The CH1 annunciator indicates that Channel 1 is selected.

NOTE

Channel 2 inputs must be referenced to Channel 1 LO.

5. Channel 2 is connected to measure temperature, so select **TEMP 2**.
6. Observe the reading on the display. The CH2 annunciator indicates that Channel 2 is selected.

SCPI commands - voltage and temperature measurements

See the following SCPI command descriptions for descriptions and programming examples:

:SENSe[1]:FUNCtion (on page 13-55)

:SENSe[1]:CHANnel (on page 13-53)

:SENSe[1]:DATA:FRESh? (on page 13-54)

:SENSe[1]:DATA[:LATest]? (on page 13-54)

:SENSe[1]:TEMPerature:RJUNction:RSELect (on page 13-57)

:SENSe[1]:TEMPerature:RJUNction:SIMulated (on page 13-58)

:SENSe[1]:TEMPerature:TCouple[:TYPE] (on page 13-59)

:SENSe[1]:TEMPerature:TRANsducer (on page 13-60)

:UNIT:TEMPerature (on page 13-102)

Range

The voltage measurement range determines the full-scale value of the measurement range. The range also affects the accuracy of the measurements and the maximum voltage that can be measured.

You can allow the 2182A to choose the range automatically or you can select a specific range.

The DCV1 function has five measurement ranges: 10 mV, 100 mV, 1 V, 10 V, and 100 V. The DCV2 function has three measurement ranges: 100 mV, 1 V, and 10 V. The range setting (fixed or automatic) is stored for each voltage function.

NOTE

The available voltage ranges for ratio (V1/V2) depend on which channel is selected when ratio is enabled. If Channel 1 is selected, DCV1 ranges are available when ratio is enabled. If Channel 2 is selected, DCV2 ranges are available when ratio is enabled. Refer to Range settings when using ratio (on page 4-32) for complete information on ranging for ratio.

There is no range selection for temperature (TEMP 1 and TEMP 2) measurements. Temperature measurements are made on a single fixed range. You can change the reading resolution using the DIGITS key.

Maximum range readings

The full scale readings for every voltage range are 20% overrange. For example, on the 10 V range, the maximum input voltage is ±12 V.

The reading ranges for temperature depend on the type of thermocouple that is selected, as shown in the following table.

Thermocouple reading ranges	
Type	**Range**
J	−200 °C to +760 °C
K	−200 °C to +1372 °C
N	−200 °C to +1300 °C
T	−200 °C to +400 °C
E	−200 °C to +1000 °C
R	0 °C to +1768 °C
S	0 °C to +1768 °C
B	+350 °C to +1820 °C

If an input value exceeds the maximum reading, the overflow message (`OVRFLW`) is displayed.

Manual ranging

To select a range, press the RANGE ▲ or ▼ key. The instrument changes one range for each key press. The selected range is displayed for one second.

The manual range keys have no effect on temperature readings.

If the instrument displays `OVRFLW` when a range is selected, select a higher range until an on-range reading is displayed. Use the lowest range possible without causing an overflow to ensure best accuracy and resolution.

Autoranging

To enable autoranging, press the AUTO key. The AUTO annunciator turns on when autoranging is selected. While autoranging is enabled, the instrument automatically selects the best range to measure the applied signal. Each increase in range occurs at 120% of range. Each decrease occurs at 10% of nominal range.

If you require optimum speed, do not use autoranging.

The AUTO key has no effect on temperature (TEMP 1 and TEMP 2).

To disable autoranging, press AUTO, then select the RANGE ▲ or ▼ key. If you press AUTO without selecting another range, the instrument remains on the last range selected by autoranging.

SCPI programming - range

For information on the range SCPI commands, refer to:

- :SENSe[1]:VOLTage[:DC][:CHANnel<X>]:RANGe:AUTO (on page 13-65)
- :SENSe[1]:VOLTage[:DC][:CHANnel<X>]:RANGe[:UPPer] (on page 13-64)

Autozero modes

To ensure the accuracy of readings, the instrument must periodically get new measurements of its internal ground and voltage reference. The time interval between updates to these reference measurements is determined by the integration aperture. The 2182A uses separate reference and zero measurements for each aperture.

An A/D measurement cycle measures the input signal and periodically measures internal voltages that correspond to offsets (zero) and amplifier gains and the internal reference temperature. These measurements help maintain stability and accuracy over time and changes in temperature. The signal, offset, gain, and temperature measurements are then used in an algorithm to calculate the reading of the input signal. This process is called autozeroing.

Internally, the 2182A has two amplifiers that have an impact on speed, noise, drift, and offset. You can control these aspects to some degree by controlling the autozero modes. The front-end amplifier is controlled by Front Autozero, and the second amplifier is controlled by autozero.

Front Autozero

When Front Autozero is enabled, the 2182A performs two A/D measurement cycles for each reading. The first one is a normal measurement cycle, and the second one is performed with the polarity of the front-end amplifier reversed. This two-cycle, polarity-reversal measurement technique cancels internal offsets in the amplifier.

When Front Autozero is disabled, the second A/D measurement cycle is not performed. Measurements are twice as fast and there is lower pumpout current noise. However, there is also high drift (20 µV/°C) and high offset voltage (±500 µV) in normal voltage mode.

NOTE

To increase the speed of delta measurements, disable Front Autozero. The two-measurement cycle, polarity-reversal technique used by Front Autozero is not required for delta. Delta uses its own polarity-reversal technique to cancel offsets. Refer to Delta (on page 6-1) for more detail.

Autozero

When autozero for the second amplifier is disabled, the measurements for offset, gain, and internal reference temperature are not made. This increases measurement speed by a few percentage points at 1 power line cycle (PLC). However, disabling autozero can cause the zero, gain, and temperature reference points to drift, resulting in inaccurate readings for the input signal. It is recommended that autozero only be disabled for short periods.

When autozero is enabled after being off for a long time, the internal reference points are not updated immediately. This initially results in inaccurate measurements, especially if the ambient temperature has changed by several degrees. To force a single rapid update of the internal reference points when autozero is enabled, set the integration rate to FAST (or 0.01 PLC for remote programming), and then return it to the normal rate (such as MED; 1.0 PLC). For more information on integration rates, refer to Rate (on page 4-19).

When autozero is disabled, pressing the front-panel RATE key changes the speed setting and enables autozero. If the rate is changed using remote programming, there is no change to the state of autozero.

To set the autozero mode from the front panel:

1. Press **SHIFT** and then **CONFIG** to display the present state of Front Autozero. `Y` is enabled; `N` is disabled.
2. Press the ► key to display the present state of autozero: `YES` (enabled), `NO` (disabled).
3. To change the AUTOZERO setting, use the ▼ or ▲ key to display `YES` or `NO`.
4. Press **ENTER** to save the settings and exit from the menu structure.

To set the autozero mode using remote commands:

Refer to :SYSTem:AZERo[:STATe] (on page 13-80).

Rate

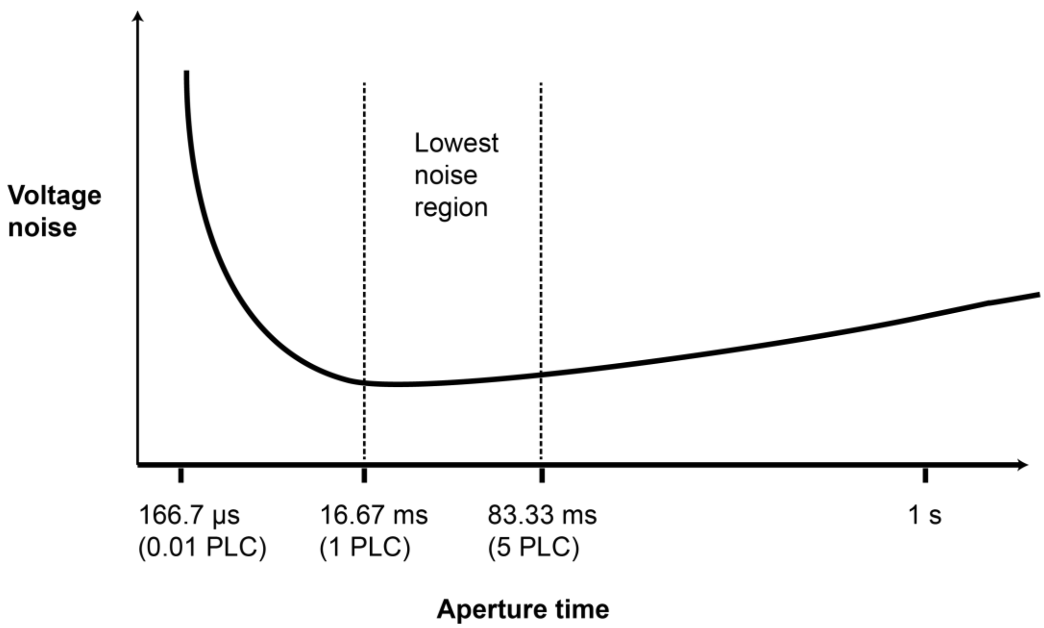
The RATE key selects the integration time of the A/D converter. This is the time the input signal is measured and is also known as aperture. The integration time affects the amount of reading noise and the reading rate of the instrument. The integration time is specified in parameters based on the number of power line cycles (PLCs), where 1 PLC for 60 Hz is 16.67 ms (1/60) and 1 PLC for 50 Hz and 400 Hz is 20 ms (1/50).

The shortest amount of time, or lowest NPLC value, results in the fastest reading rate but increases the reading noise and decreases the number of usable digits.

The longest amount of time, or highest NPLC value, provides the lowest reading noise and more usable digits, but has the slowest reading rate.

The 2182A speed versus noise characteristics are shown in the following figure. The 2182A is optimized for the 1 PLC to 5 PLC reading rate. At these speeds, which are shown in the lowest noise region in the graph, the 2182A makes corrections for its own internal drift and is still fast enough to settle a step response of less than 100 ms.

Figure 21: Speed versus noise characteristics



You can have a separate rate setting for voltage and temperature functions. The rating that is set applies to both functions of that type (DCV1 and DCV2 or TEMP 1 and TEMP 2). For example, if you set DCV1 for 0.1 PLC, DCV2 is also set for 0.1 PLC.

To set the rate from the front panel, select **RATE** and select one of the following options:

- **0.1 PLC:** Selects the fastest front-panel integration time. Select 0.1 PLC if speed is most important.
- **1 PLC:** Selects a medium integration time. Select 1 PLC when a compromise between noise performance and speed is acceptable.
- **5 PLC:** Selects the slowest front-panel integration time. Selecting 5 PLC provides better noise performance at the expense of speed.

NOTE

Pressing the front-panel RATE key enables autozero if it was off. For remote programming, the rate commands have no effect on the state of autozero. For more information, see Autozeroing modes (on page 4-17).

To set the integration rate from the front panel:

1. Select the function.
2. Press the **RATE** key until the number of power line cycles (PLC) is displayed. The appropriate annunciator turns on (FAST, MED, or SLOW).

SCPI programming - rate

For the SCPI commands that control the rate, refer to:

- :SENSe[1]:<function>:APERture (on page 13-42)
- :SENSe[1]:<function>:NPLCycles (on page 13-43)

NOTE

If you are using remote commands, you can set the integration time from 0.01 PLC to 60 PLC (50 PLC for 50 Hz line power). You can also set integration time as an aperture time from 166.67 µs (200 µs for 50 Hz) to 1 second.

Filters

The 2182A has an analog filter and a digital filter. When the filter is enabled, the settings for the analog and digital filters are applied to the present measurement function (DCV1, DCV2, TEMP 1, or TEMP 2).

To enable the filter, press the **FILT** key. When the filter is enabled, the FILT annunciator is on. The filter state and configuration are saved for each measurement function.

Analog filter

When you enable the low-pass analog filter, the normal-mode noise rejection ratio of the instrument is increased at 60 Hz. This filters out noise induced by the power line. The analog filter attenuates frequency at 20 dB/decade starting at 18 Hz.

A primary use of the analog filter is to keep the high-gain input stage of the 2182A from saturating due to the presence of high ac and dc voltage.

NOTE

The filter only attenuates ac voltages for the 10 mV range of the 2182A.

The analog filter adds approximately 125 ms of settling time between A/D conversions. The additional settling time may be required when using a high-impedance (≥100 kΩ) source in the test circuit. The increased settling time causes the reading rate of the 2182A to be greatly reduced. Therefore, if the analog filter is not needed, turn it off.

Digital filter

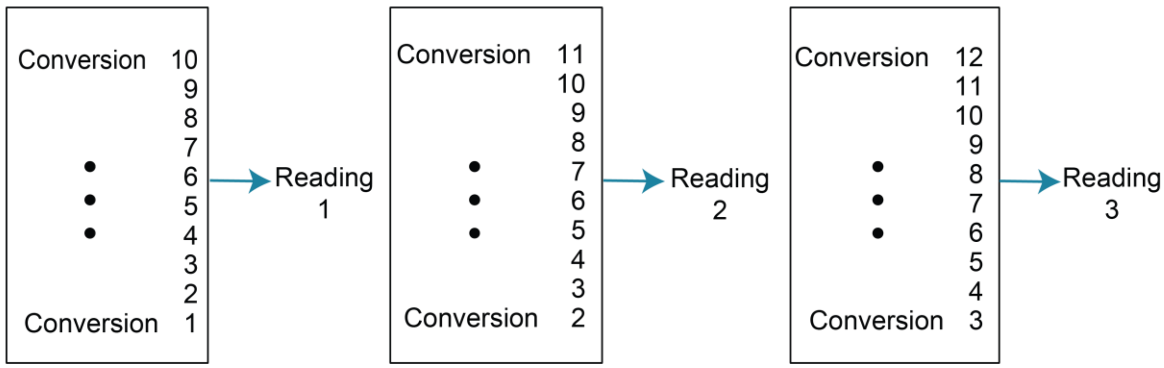
The digital filter stabilizes noisy measurements. The displayed, stored, or transmitted reading is a windowed-average of a number of reading conversions (from 1 to 100).

In general, the digital filter places a specified number of consecutive A/D conversions (filter count) into a memory stack. The readings in the stack are then averaged to yield a single filtered reading. You can set the filter to use a moving or repeating average.

When the moving average filter is selected, the measurements are added to the stack continuously on a first-in, first-out basis. As each measurement is made, the oldest measurement is removed from the stack. A new averaged sample is produced using the new measurement and the data that is now in the stack.

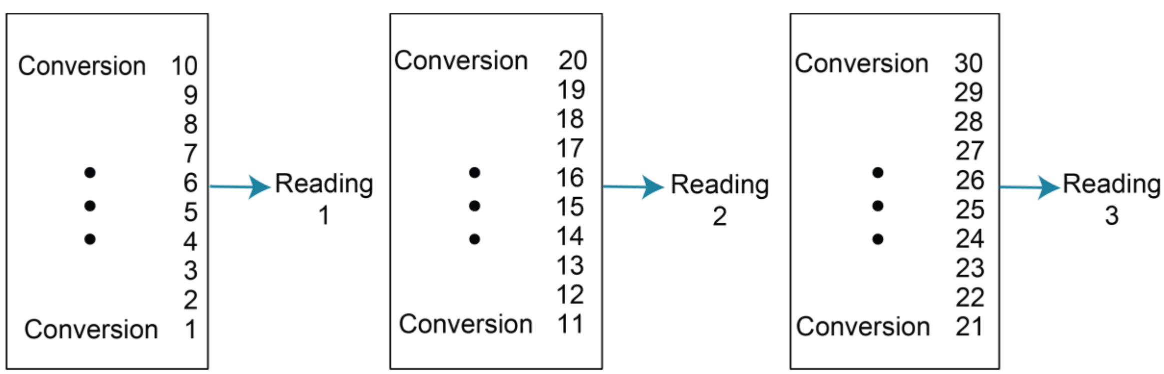
When the moving average filter is first selected, the stack is empty. When the first measurement is made, it is copied into all the stack locations to fill the stack. A true average is not produced until the stack is filled with new measurements. See the following figure for an example of a moving average filter.

Figure 22: Moving average filter



When the repeating average filter is selected, a set of measurements are made. These measurements are stored in a measurement stack and averaged together to produce the averaged sample. Once the averaged sample is produced, the stack is flushed, and the next set of data is used to produce the next averaged sample. This type of filter is the slowest, since the stack must be completely filled before an averaged sample can be produced, but it provides more stable results. See the following figure for an example of a repeating average filter.

Figure 23: Repeating average filter



For either method, the greater the number of measurements that are averaged, the slower the averaged sample rate, but the lower the noise error. Trade-offs between speed and noise are normally required to tailor the instrumentation to your measurement application.

Digital filter characteristics

You can set the filter type, filter count, and filter window for the digital filter.

You can set the filter type to moving or repeating, as described in Digital filter (on page 4-21).

If you are using delta measurements, you must use the moving filter. When delta measurements are selected, if filters are enabled, the 2182A uses a moving filter regardless of the filter type setting. Refer to Delta (on page 6-1) for more information on delta measurements.

If you are stepping or scanning, you must use the repeating filter. When the instrument is stepping or scanning and filters are enabled, the 2182A uses a repeating filter regardless of the filter type setting. Refer to Stepping and scanning (on page 7-1) for more information.

The filter count specifies how many consecutive A/D conversions in the filter window to place in the memory stack. When the stack is full, the A/D conversions are averaged to calculate the final filtered reading. The filter count can be set from 1 to 100. If the filter count is 1, no averaging is done. Only readings in the filter window are displayed, stored, or transmitted.

The digital filter uses a window to control the filter threshold. While the input signal remains in the selected window, A/D conversions continue to be placed in the stack. If the signal changes to a value outside the window, the filter resets, and the filter starts processing again, starting with a new initial conversion value from the A/D converter.

From the front panel, you can select a window of 0.01%, 0.1%, 1%, or 10% of range, or NONE (no window). With remote commands, you can set the window to any value from 0% to 10% or NONE.

For the voltage function, the filter window is expressed as a percent of range. For example, on the 10 V range, a 10% window means that the filter window is ±1 V. For temperature, the filter window is expressed as a percent of the maximum temperature reading. The maximum temperature depends on which thermocouple is used. For example, for a Type J thermocouple, the maximum reading is 760 °C, so a 10% window creates a filter window of ±76 °C.

Filter control and configuration

The FILT key toggles the state of filters. When filter is enabled, the FILT annunciator is on. You can configure the analog and digital filters while filter is enabled or disabled. If filter is enabled, changes to the configuration take effect as soon as they are made. With filter disabled (FILT annunciator off), changes to the configuration take effect the next time the filter is enabled.

If both the analog and digital filters are configured to be off and filter is enabled, the digital filter automatically turns on. This ensures that some type of filtering is applied when the FILT annunciator is on.

While the filtering operation is in progress, the FILT annunciator blinks. Readings continue to be processed, but they may be questionable. When the FILT annunciator stops blinking, the filter has settled.

Filters are reset when the function or range changes. The filter settings change to the state (enabled or disabled) and configuration for that function or range.

When both channels are being measured for ratio, the filter state (enabled or disabled) and configuration for DCV1 is used.

To configure the filter using the front panel:

1. Select the function (DCV1, DCV2, TEMP 1, or TEMP 2).
2. Press **SHIFT** and then **TYPE**. The present state of the analog filter is displayed.
3. To enable or disable the analog filter, use the ◄ and ► keys to move the cursor to **ON** or **OFF**. Press the **RANGE** ▲ or ▼ key to change the setting.
4. Press **ENTER**. The present state of the digital filter (on or off) is displayed.
5. To change the state of the digital filter, use the ◄ and ► keys to move the cursor to **ON** or **OFF**. Press the **RANGE** ▲ or ▼ key to change the setting.
6. Press **ENTER**. The present digital filter window setting is displayed.
7. Use the **RANGE** ▲ or ▼ key to display the window setting.
8. Press **ENTER**. The present digital filter count is displayed.
9. To change the count, use the ◄ and ► keys and ▲ and ▼ keys to display the count setting.
10. Press **ENTER**. The present digital filter type is displayed.
11. To change the digital filter type, place the cursor on the type name and press the **RANGE** ▲ or ▼ key.
12. Press **ENTER**. The instrument returns to the normal measurement display state.

Digital filter example

Filter Count = 10
Filter Window = 0.01% of range
Filter Type = Moving

Ten readings fill the stack to yield a filtered reading. Now assume the next reading (which is the 11th) is outside the window. A reading is processed and displayed; however, the stack is loaded with that same reading. Each subsequent valid reading then displaces one of the loaded readings in the stack. The FILT annunciator flashes until 10 new readings fill the stack.

NOTE

Bit 8 of the Operation Event Status Register sets when the filter window has settled. See Status structure (on page 16-1) for details.

SCPI programming - filter

See the following SCPI command descriptions for descriptions and programming examples:

:SENSe[1]:<function>[:CHANnel<X>]:DFILter:COUNt (on page 13-45)

:SENSe[1]:<function>[:CHANnel<X>]:DFILter:TCONtrol (on page 13-46)

:SENSe[1]:<function>[:CHANnel<X>]:DFILter:WINDow (on page 13-47)

:SENSe[1]:<function>[:CHANnel<X>]:DFILter[:STATe] (on page 13-48)

Relative offset

When making measurements, you may need to subtract an offset value from a measurement.

The relative offset feature subtracts a set value or a baseline reading from measurement readings. When you enable relative offset, all measurements are recorded as the difference between the actual measured value and the relative offset value. The formula to calculate the offset value is:

Displayed value = Actual measured value − Relative offset value

When a relative offset value is established for a measure function, the value is the same for all ranges for that measure function. For example, if 5 V is set as the relative value on the 10 V range for DCV1, the relative value is also 5 V on the 100 V, 1 V, 100 mV, and 10 mV ranges.

On the front panel, when relative offset is enabled, the REL indicator is displayed.

When a relative value is larger than the selected range, the display is formatted to accommodate the reading with the offset applied. However, this does not increase the maximum allowable input for that range. An overrange input signal still causes the display to overflow. For example, on the 10 V range, the 2182A overflows for a 12 V input.

NOTE

For information on using readings with offset applied in ratio calculations. See Ratio (on page 4-31) for more information.

For information on canceling residual thermal offsets, see Canceling residual thermal offsets (on page 5-4).

For information on setting a relative offset value for the analog output, see Analog output (on page 4-34).

For offsets that vary, use the dc current-reversal technique instead of relative offset. This technique uses the delta measurement mode of the 2182A to cancel offsets. Refer to Delta, pulse delta, and differential conductance (on page 6-1) for more information.

You can use the mx+b function to manually establish a relative offset value. To do this, set the scale factor (m) to 1 and set the offset (b) to the offset value. Each subsequent reading is the difference between the actual input and the offset value.

The relative offset value is set for the selected function (DCV1, DCV2, TEMP 1, or TEMP 2). You can set a unique relative offset value for each measurement function and for each channel.

To set a relative offset value from the front panel:

1. Connect the test circuit.
2. Turn on the instrument and leave the source (voltage or current) disconnected or in stand-by.
3. Select the channel.
4. Select the function.
5. Display the reading you want to use for the relative offset value, such as a zero offset reading that you want to null out or an applied level that you want to use as a baseline.
6. If you are not using autorange, select the lowest possible measurement range to display the voltage offset.
7. Press the **REL** key to zero the display. The REL annunciator turns on and subsequent readings are the difference between the actual input and the relative offset value.
8. To disable the relative offset, press the **REL** key again. The REL annunciator turns off.

SCPI programming - relative offset

For information on setting relative offset using SCPI commands, refer to:

:SENSe:<function>[:CHANnel<X>]:REFerence (on page 13-50)

:SENSe:<function>[:CHANnel<X>]:REFerence:ACQuire (on page 13-51)

:SENSe:<function>[:CHANnel<X>]:REFerence:STATe (on page 13-52)

Pumpout current (low charge injection mode)

Pumpout current is a small discharge of current at the meter input. Pumpout current can sometimes be a problem for low-level signals. A symptom of a pumpout current problem is when there appears to be noise injections in the measurement results. This current discharge results from the switching the instrument does between measurements of the input signal and the background measurements made by the analog to digital (A/D) converters, which include measurement of reference and zero points.

Pumpout current for Channel 1 is very low (0.5 µA peak-to-peak) and therefore does not adversely affect instrument performance. The pumpout current for Channel 2 is also very low if Channel 2 LO is connected to Channel 1 LO.

Settling for the transition occurs on the next A/D conversion. Whenever the impedance between Channel 2 LO and Channel 1 LO is >100 kΩ, pumpout current could be high enough to corrupt measurements below 1 V. Above 1 V measurements, pumpout current is not significant.

If you must use Channel 2 for measurements below 1 V and the impedance between Channel 2 LO and Channel 1 LO is >100 kΩ, you can enable the low charge injection mode to reduce the pumpout current. However, this mode increases measurement noise by up to eight times.

Low charge injection mode is only available through a remote interface. To enable low charge injection mode, send the command:

```
CALL SEND(7,":SENS:VOLT:CHAN2:LQM ON",status%)
```

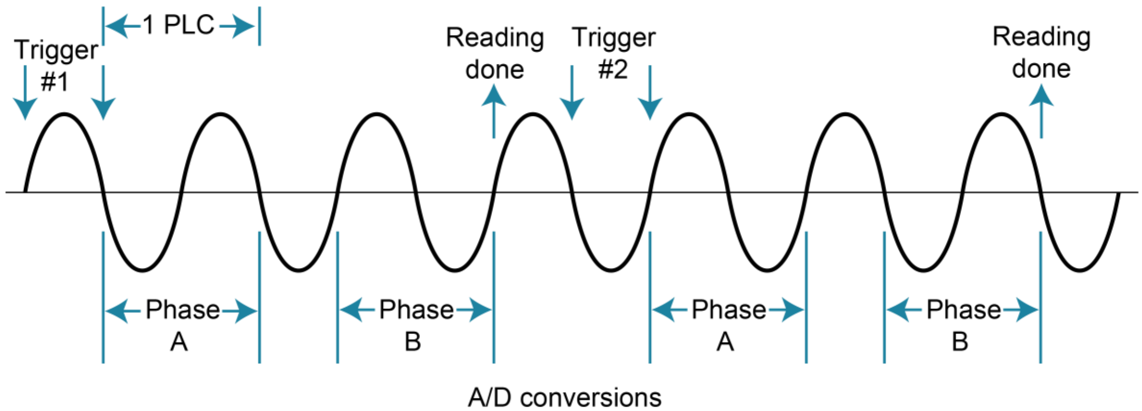
Refer to :SENSe[1]:VOLTage[:DC]:CHANnel2:LQMode (on page 13-61) for more information on low charge injection mode.

Line cycle synchronization

Using line synchronization helps increase common-mode and normal-mode noise rejection. When line cycle synchronization is enabled, measurements are initiated at the first positive-going zero crossing of the power line cycle after the trigger.

The following figure shows the measurement process, which consists of two A/D conversions. If the trigger occurs during the positive cycle of the power line (as shown in the figure), the first A/D conversion starts with the negative-going zero crossing of the power line cycle. If the next trigger (Trigger #2) occurs during the negative cycle, the measurement process starts with the positive-going zero crossing.

Figure 24: Line cycle synchronization



NOTE

Line cycle synchronization is not available for integration rates <1 PLC, regardless of the LSYNC setting.

To enable or disable line cycle synchronization from the front panel:

1. Press **SHIFT** and then **LSYNC** to display the present state of line synchronization (OFF or ON).
2. Use ▼ or ▲ key to display `ON` or `OFF`.
3. Press **ENTER**. The instrument returns to the normal display state.

SCPI programming - Line cycle synchronization

Refer to :SYSTem:LSYNc[:STATe] (on page 13-86).

Calculations that you can apply to measurements

The 2182A allows you to apply the following math operations to the measurement:

- mx+b
- percent

The calculations are applied to all measurement functions.

mx+b

The mx+b math operation lets you manipulate normal display readings (x) mathematically based on the following calculation:

mx + b = y

Where:

- **m** is a user-defined constant for the scale factor
- **x** is the measurement reading (if you are using a relative offset, this is the measurement with relative offset applied)
- **b** is a user-defined constant for the offset factor
- **y** is the displayed result

You can use the mx+b function to manually establish a relative offset value. To do this, set the scale factor (m) to 1 and set the offset (b) to the offset value. Each subsequent reading is the difference between the actual input and the offset value.

NOTE

The mx+b function does not affect the analog output. Analog output has its own gain and offset settings. Refer to Analog output (on page 4-34) for more information.

To configure the mx+b calculation from the front panel:

1. Press the **SHIFT** key and the **MX+B** key to display the present scale factor. The default is 1.
2. Enter the **m** factor. Use ◄ and ► to move the cursor and ▲ and ▼ to increase or decrease the value.
3. To change the range, place the cursor on the multiplier and use the ▲ and ▼ keys to display the range (m = ×0.001, ^ = ×1, K = ×1000, and M = ×1,000,000). With the cursor on the polarity sign, the ▲ and ▼ keys toggle polarity.
4. Press **ENTER** to enter the m value and display the b value:

   `B:  +00.000000 m    (factory default)`
5. Enter the offset value.
6. Press **ENTER** to enter the b value and display the two-character UNITS designator:

   `UNITS:  MX   (factory default)`
7. Use the cursor keys and the ▲ and ▼ key to change the units designator. Each character can be any letter in the alphabet (A through Z), the degrees symbol (°), or the ohms symbol (Ω).
8. Press **ENTER**. The MATH annunciator turns on, and the result of the calculation is displayed.
9. To disable mx+b, press **SHIFT** and then **MX+B**. The MATH annunciator turns off.

Percent (%)

The percent math function determines the percent deviation from a specified reference value. The percent calculation is made as follows.

$$Percent = \frac{Input - Reference}{Reference} x\ 100\%$$

Where:

- Percent is the displayed result
- Input is the normal display reading
- Reference is the user-entered constant

NOTE

The result of the percent calculation is positive when the input exceeds the reference and negative when the input is less than the reference. The result of the percent calculation may be displayed in exponential notation. For example, a displayed reading of + 2.500E+03% is equivalent to 2500%.

To configure and control the percent calculation:

1. Press **SHIFT** and then **%** to display the present reference value.

   `REF: +1.000000 ^    (factory default)`

2. To change the range, place the cursor on the multiplier and use the ▲ and ▼ keys (m = ×0.001, ^ = ×1, K = ×1000, and M = ×1,000,000). With the cursor on the polarity sign, the ▲ and ▼ keys toggle polarity.
3. Enter a reference value. The ◄ and ► keys control cursor position and the ▲ and ▼ range keys increment and decrement the digit value.
4. Press **ENTER**. The MATH annunciator turns on and the result of the calculation is displayed.
5. To disable percent, press **SHIFT** and then **%**. The MATH annunciator turns off.

SCPI programming - mx+b and percent

Refer to CALCulate[1] subsystem (on page 13-6) for detail on programming mx+b and percent.

Ratio

Ratio (V1/V2) displays the proportional relationship between the two voltage input channels (DCV1 and DCV2). Ratio is calculated as follows:

Ratio = V1/V2

Where:

- V1 is the voltage reading for Channel 1 (DCV1)
- V2 is the voltage reading for Channel 2 (DCV2)

When the ratio feature is selected, one of the channel annunciators (CH1 or CH2) turns on briefly. This indicates the channel that you can control with the manual range key. After the brief display, the CH1 and CH2 annunciators are active. You can press the AUTO range key to either enable autorange for both channels or disable autorange for both channels. Refer to Range settings when using ratio (on page 4-32) for additional information.

Line cycle integration (LSYNC) must be enabled when ratio is selected. LSYNC turns on automatically when ratio is enabled and turns off automatically when ratio is disabled.

If an overflow condition (OVRFLW) occurs, the range that overflowed formats the display.

Ratio readings can be stored in the buffer.

NOTE

You cannot use reading hold with ratio. Selecting ratio disables reading hold.

WARNING

A hazardous voltage condition exists at or above 42 V_{PEAK}. To prevent electric shock that could result in injury or death, never make or break connections while hazardous voltage is present.

CAUTION

Channel 1 HI and LO terminals have a maximum measurement capability of 120 V_{PEAK}. These inputs are protected to 150 V_{PEAK} to any terminal or 350 V_{PEAK} to chassis. Exceeding these limits may cause instrument damage not covered by the warranty.

CAUTION

Channel 2 HI and LO terminals have a maximum measurement capability of 12 V_{PEAK} to Channel 1 LO. Channel 2 HI is protected to 150 V_{PEAK} to any terminal. Channel 2 LO is protected to 70 V_{PEAK} to Channel 1 LO. Both inputs are protected to 350 V_{PEAK} to chassis.

To set ratio from the front panel:

1. Connect the 2182A for voltage as described in Test connections (on page 4-5).
2. Configure each channel (DCV1 and DCV2) for the voltage measurement.
3. Verify that DCV1 and DCV2 are displaying on-scale readings. If an `OVRFLW` message is displayed for any channel, select a higher range until an on-scale reading is displayed. You can also press **AUTO** to enable the autorange feature.
4. If you want the manual range keys to control Channel 1 ranges, press **DCV1**. If you want the manual range keys to control Channel 2 ranges, select **DCV2**.
5. To enable ratio, press the **V1/V2** key. While in ratio, the `RA` message is displayed.
6. Make ratio readings from the display.
7. To disable ratio, select a single measurement function (**DCV1**, **DCV2**, **TEMP 1**, or **TEMP 2**) or **Delta**.

SCPI command - ratio

Refer to :SENSe:VOLTage[:DC]:RATio (on page 13-63) for the command description.

Range settings when using ratio

You can use a separate range setting (fixed or automatic) for each voltage channel. When ratio is enabled, the range setting for each channel is retained. For example, you can set Channel 1 to autorange and you can set Channel 2 to the 10 V range.

The manual range keys can only control one of the two channels. If the instrument is on DCV1, TEMP 1, or TEMP 2 when ratio is enabled, the manual range keys control Channel 1 (DCV1). If the instrument is on DCV2 (Channel 2) when ratio is enabled, range control applies to Channel 2 (DCV2). When a range key is pressed, the channel number annunciator for the range-controlled channel remains on. The other channel annunciator turns off for a moment.

The state (on or off) of the AUTO range annunciator indicates the state of autorange for the voltage channel that is under range control. When ratio is enabled, pressing the AUTO range key either disables autorange for both channels or enables autorange for both channels (AUTO annunciator turns on).

When ratio is enabled, the range control channel is displayed while the `CH1 / CH2` message is displayed.

Relative offset when using ratio

You can set a separate relative offset value for each voltage channel. When ratio is enabled, the relative offset values are applied to the respective channels before the calculation is made.

Ratio with relative off set applied is calculated as follows:

Ratio = (Filt V1 − V1 Rel) / (Filt V2 − V2 Rel)

Where:

- Filt V1 is the filtered reading for Channel 1 voltage input
- V1 Rel is the relative offset value established for Channel 1
- Filt V2 is the filtered reading for Channel 2 voltage input
- V2 Rel is the relative offset value established for Channel 2

NOTE

The previous calculation shows Filter enabled. If Filter is not used, remove the "Filt" component from the calculation.

When ratio is enabled, the state (on or off) of the REL annunciator depends on which measurement function was last selected. If the function is DCV1 when ratio is enabled, the REL annunciator indicates the state for DCV1. If the function is DCV2 when ratio is enabled, the REL annunciator indicates the state of relative offset for DCV2.

The REL key is operational while in ratio. When you press REL, it either disables relative offset for both channels or enables relative offset for both channels. When relative offset is enabled, the instrument acquires the input signal from each of the two channels as relative offset values. Each relative offset value is then applied to the respective channel.

NOTE

Relative offset operations are performed on the input channels, not on the result of ratio.

For more information on relative offset, see Relative offset (on page 4-25).

Analog output

The analog output is a scaled, non-inverting voltage output up to ±1.2 V. It is typically used to drive a chart recorder. The analog output voltage is calculated as follows:

Analog output = (Gain × Reading/Range) − Offset

Where:

- Gain is the user-entered gain factor
- Reading is the reading on the 2182A
- Range is the measurement range
- Offset is a user-entered offset value

NOTE

Gain and offset for the analog output are not related to gain and offset for the mx+b calculation.

Gain provides amplification for small analog output voltage signals, while offset allows you to adjust the analog output to keep it between the maximum output of ±1.2 V or reference the voltage output to a specific value, such as zero.

For example, assume you are measuring 100 mV on the 1 V range. With gain set to 1, the analog output is 100 mV. You can increase analog output sensitivity by setting the gain to 10. This increases the analog output to 1 V. You can then set the offset to 1 V to reference the 1 V analog output to zero. The analog output calculation for this example is:

Analog output = (10 × 100 mV/1 V) − 1 V = 0 V

The factory default for gain is 1 and the factory default for offset is 0. Therefore, when using the factory defaults, gain and offset drop out of the equation:

Analog Output = Reading/Range

The following table shows analog output examples with the gain set to 1 and offset set to 0.

Reading	Range	Analog output voltage
1 V	1 V	+1 V
−1 V	1 V	−1 V
1 V	10 V	0.1 V
12 V	10 V	1.2 V
50 mV	100 mV	0.5 V
−1 mV	1 V	−1 mV

Temperature

The analog output voltage for temperature measurements depends on thermocouple type and the selected units (°C, °F, or K). The 1.2 V analog output is scaled to the maximum positive temperature reading. For example, the measurement range for the type J thermocouple is −200°C to +760°C. For a 760°C reading, the analog output voltage is 1.2 V. For a −200°C reading, the analog output voltage is −0.316 V. The measurement ranges (in °C) for the various thermocouple types are listed in the specifications, available at tek.com/keithley.

With the gain set to 1 and offset set to 0, analog output voltage for temperature measurements is calculated as follows:

Analog output = 1.2 × Reading/Range

In this formula, range is a magnitude, so it is always a positive value.

Example calculations:

Type J, 100°C reading

Analog output = 1.2 × 100/760
= 158 mV

Type J, –100°C reading

Analog output = 1.2 × (−100/760)
= −158 mV

When using gain and offset, the calculation is expanded as follows:

Analog output = (Gain × 1.2 × Reading/Range) – Offset

Ratio

You can use analog output with the ratio feature. When enabled, the analog output voltage is scaled to a ratio value of 1. That is, the analog output is 1 V for a ratio result of 1. If, for example, the ratio is 0.4, the analog output voltage is 0.4 V. You can also use gain, offset, and analog output relative offset.

The maximum analog output is ±1.2 V.

Refer to Ratio (on page 4-31) for more information on using the ratio feature.

Analog output connections

The analog output is accessed from the rear-panel BNC connector labeled ANALOG OUTPUT. This connector requires a cable that is terminated with a standard male BNC connector.

The output resistance of the analog output is 1 kΩ ±5%. To minimize the effects of loading, the input resistance of the device connected to ANALOG OUTPUT should be as high as possible. For example, for a device with an input resistance of 10 MΩ, the error due to loading is approximately 0.01%.

Configure the analog output

To configure the analog output:

1. Press **SHIFT** and then **AOUT** to display the Gain (M) factor. The default is 1, with a range of −100e6 to 100e6.
2. Use the ◄ and ► keys to move the cursor position and the ▲ and ▼ range keys to increment and decrement the digit value. To change the range, place the cursor on the multiplier and use the ▲ and ▼ keys (m = ×0.001, ^ = ×1, K = ×1000, and M = ×1,000,000). With the cursor on the polarity sign, the ▲ and ▼ keys toggle polarity.
3. Press **ENTER** to enter the gain value and display the Offset value B. The default is 0.
4. Enter the offset value.
5. Press **ENTER** to enter the offset value and enable the analog output. The instrument returns to the normal display state.
6. To disable the analog output, press **SHIFT** and then **AOUT**.

Analog output relative offset

When analog output is enabled, the analog output relative offset automatically references the analog output voltage to zero.

If you enable relative offset when the analog output is enabled, the present analog output voltage is used as the relative offset value. This sets the analog output voltage to zero. Subsequent analog output readings are the difference between the actual analog output and the relative offset value.

To enable the analog output relative offset from the front panel, press **SHIFT** and then **OUTPUT**. The message `AOUT REL ON` is displayed briefly. To disable the analog output relative offset, press **SHIFT** and then **OUTPUT** again. The message `AOUT REL OFF` is displayed briefly.

You can establish a new relative offset value for the analog output by disabling the analog output relative offset and enabling it again.

SCPI programming - analog output

Commands and programming examples for analog output are described in OUTPut subsystem (on page 13-36).

Applications

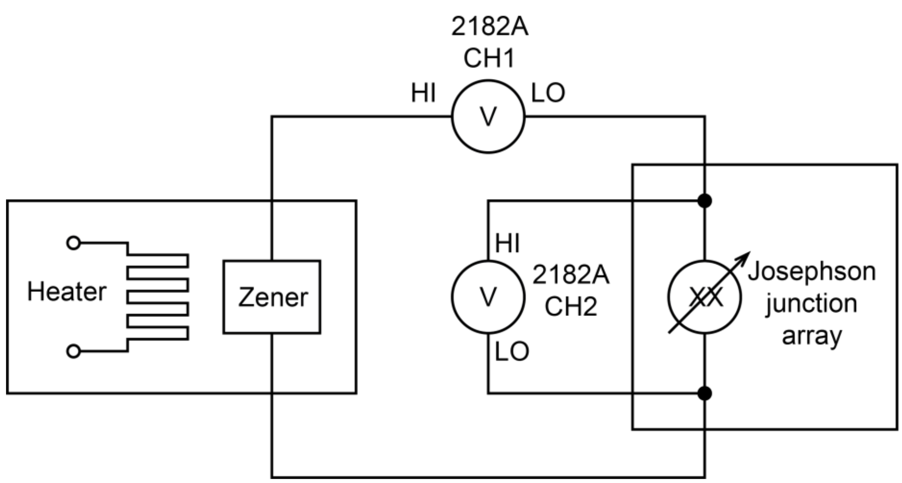
The following examples demonstrate common 2182A applications.

Heated Zener reference and Josephson Junction Array comparisons

You can analyze the performance of a heated Zener reference by comparing it to a Josephson Junction (JJ) Array using both channels of the 2182A. In a cryogenic environment, the JJ Array provides an output voltage in precise, stable 175 µV steps.

The test circuit for this application is shown in the following figure. The JJ Array is adjusted until Channel 1 of the 2182A measures 0 V ±10 µV. The null condition indicates that the heated Zener reference voltage is the same as the JJ Array voltage. Channel 2 of the 2182A determines the exact step that the JJ Array is on. You can then monitor Channel 1 to study noise and drift characteristics of the heated Zener reference.

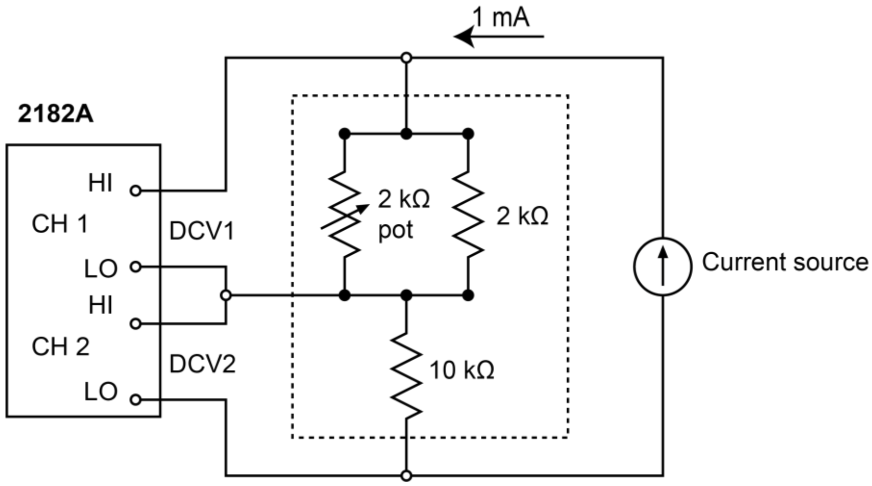
Figure 25: Heated Zener characterization



Calibrating resistor network dividers

You can use the ratio feature to calibrate resistor network dividers. The 1:10 divider network in the following figure is made up of nominal resistances of 1 kΩ and 10 kΩ. The 1 kΩ resistance is the result of the parallel combination of the 2 kΩ pot and the 2 kΩ resistor. The pot provides fine-tuning of the network.

Figure 26: Calibrating 1:10 divider



Use the Keithley Model 6220 to source a constant current of 1 mA. Channel 1 is set to the 1 V range (1 kΩ × 1 mA = 1 V) or autorange. Channel 1 measures the 1 kΩ resistance component of the network. Channel 2 is set to the 10 V range or autorange. Channel 2 measures the 10 kΩ resistance component.

When ratio is enabled, the 2182A displays the result of V1 (approximately 1 V) divided by V2 (approximately 10 V):

$$\text{Ratio} = \frac{\text{FiltV1}}{\text{FiltV2}} = \frac{\approx 1\text{ V}}{\approx 10\text{ V}} = \approx 0.1$$

This calculation includes the Channel 1 Filter. If a filter is not used, remove the Filt component from the calculation.

The network can then be calibrated by adjusting the network pot until a reading of 0.10000 is displayed.

For even greater precision, you can use the relative offset feature of the 2182A to cancel thermal EMFs, which can corrupt low voltage measurements.

To use relative offset with ratio:

1. While displaying the ratio result, disconnect the current source from the network.
2. Press the **REL** key on the 2182A. The voltages at each input, which are thermal EMFs, are cancelled.
3. Reconnect the current source and take the result of ratio from the display.

When using relative offset, the ratio is calculated as follows:

$$\text{Ratio} = \frac{\text{Filt V1} - \text{V1 Rel}}{\text{Filt V2} - \text{V2 Rel}}$$

This calculation includes the Channel 1 Filter. If a filter is not used, remove the Filt component from the calculation.

Testing superconductor materials

A superconductor sample is typically tested by either varying the current through it or varying the magnetic field that surrounds it.

NOTE

The following applications use magnetic field (H) as one of the test parameters. You can modify the applications to substitute temperature (T) for H as a test parameter.

When varying the magnetic field, the current (I) that flows through the DUT is fixed. When varying the current through the superconductor material (DUT), the magnetic field that surrounds it is held constant.

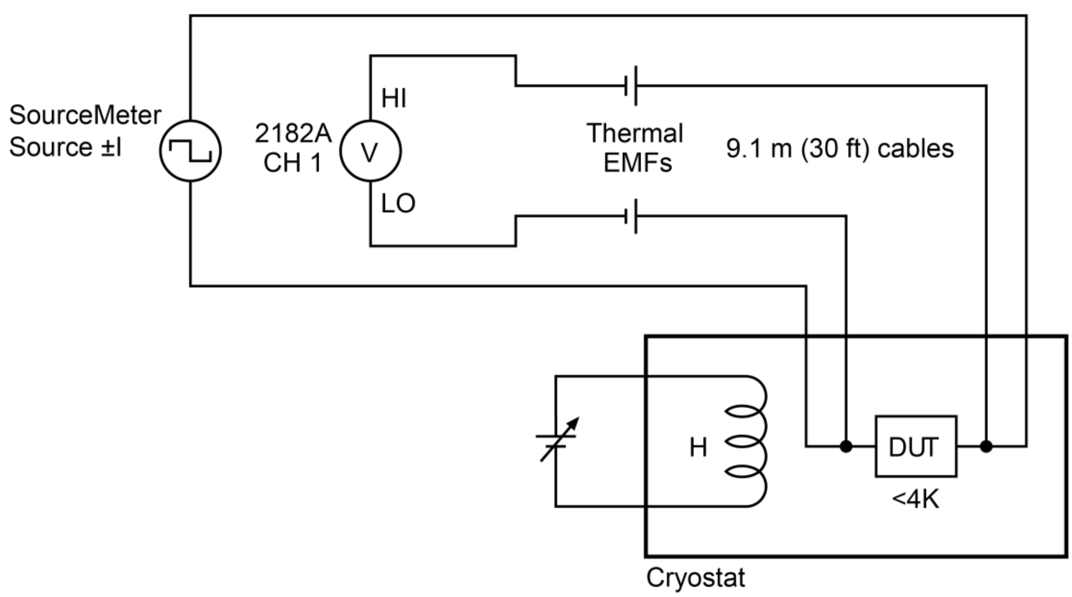
For the following applications, the 2182A measures voltage and a Keithley SourceMeter Model 2400, 2410, or 2420 sources a known current. The actual resistance of the DUT can be calculated using Ohm's Law.

Test leads that connect the 2182A to the superconductor sample (DUT) in a cryostat are typically 30 feet or longer. The test lead connections and the wide temperature range (from ~0 K at the DUT to the ambient temperature of the lab) create substantial thermal EMFs in the test leads. The effects of these thermal EMFs must be canceled to achieve accurate voltage measurements.

Superconductor Application 1: Fixed current

A typical test on a superconductor sample (DUT) is to vary the magnetic field (H) while maintaining a fixed current (I) through the DUT, as shown in the following figure. The current source of the SourceMeter is a constant current source. Therefore, the current through the DUT remains constant as the resistance of the DUT increases.

Figure 27: Test fixture with fixed current (Vary H)



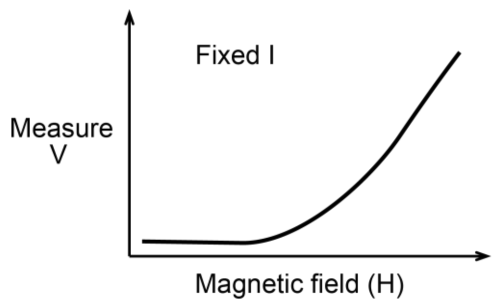
After measuring the DUT voltage (V) at a series of increasing magnetic field values, you can graph H versus V. The example H-V curve in the following figure shows that the measured voltage across the DUT remains at ~0 V in low magnetic fields. This is the flat portion of the curve where the DUT remains at 0 Ω. At some point, the voltage drop across the DUT starts to increase as the magnetic field increases. The actual resistance of the DUT can be calculated at any magnetic field point using Ohm's law:

R = V/I

where:

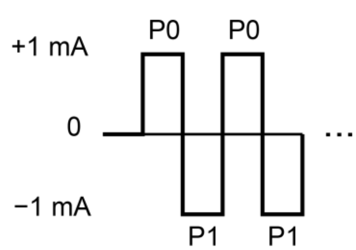
- R is actual resistance of the DUT.
- V is the measured voltage across the DUT.
- I is the known current that flows through the DUT.

Figure 28: H-V curve (fixed current)



To cancel the effects of thermal EMFs in the test leads, you must use the dc current reversal measurement technique. By configuring a custom sweep, the SourceMeter can function as a fixed bipolar amplitude source. For example, if the test requires a fixed current of 1 mA, the custom sweep can be configured to alternate between +1 mA and −1 mA, as shown in the following figure. By enabling delta measurements on the 2182A, the effects of thermal EMFs in the test leads are automatically canceled during the source-measure process.

Figure 29: SourceMeter output: Two-point custom sweep



The procedure to use the SourceMeter and 2182A to perform delta measurements is provided in Delta measurement procedure using a SourceMeter (on page 6-5). That procedure uses a 2-point custom sweep, which is required for this application.

Superconductor Application 2: Fixed magnetic field

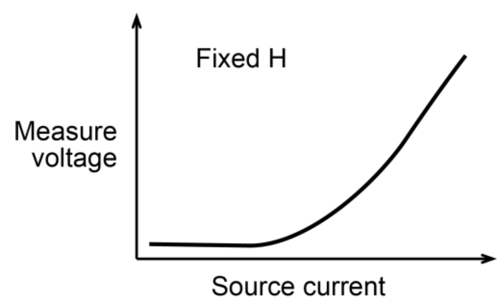
Another typical test on a superconductor sample (DUT) is to source an increasing-amplitude current (I) through the DUT, while maintaining the magnetic field (H) at a fixed level. The I-V curve in the following figure shows that the measured voltage across the DUT remains at ~0V for low currents. This is the flat portion of the curve where the DUT remains at 0 Ω. At some point, the voltage drop across the DUT starts increasing as current through the DUT increases. The actual resistance of the DUT can be calculated at any current source point using Ohm's law:

R = V/I

where:

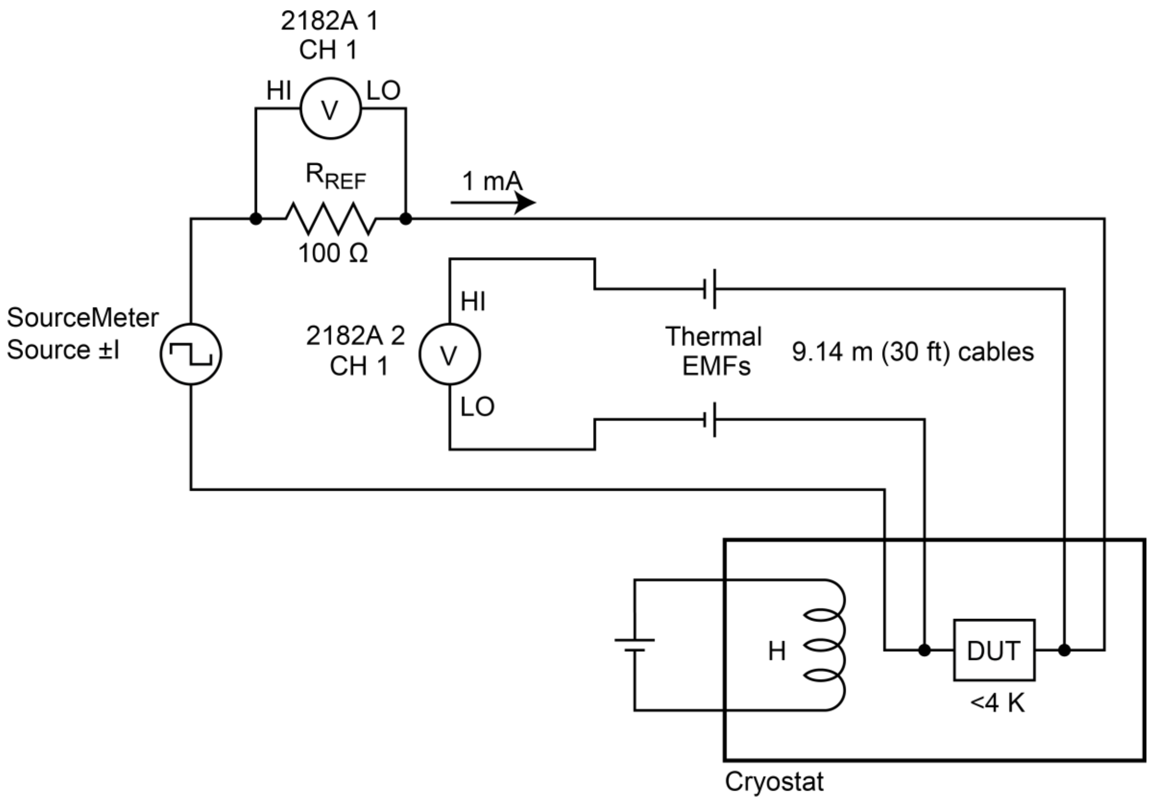
- R is actual resistance of the DUT.
- V is the measured voltage across the DUT.
- I is the known current that flows through the DUT.

Figure 30: Current-voltage curve (fixed H)



An example of this test system is shown in the following figure. A Keithley SourceMeter (Model 2400, 2410, or 2420) sources current through the DUT and two 2182As provide simultaneous voltage measurements.

Figure 31: Test circuit: Fixed H (Vary I)



2182A 1 measures the voltage across a precision reference resistor (R_{REF}) and stores the readings in its buffer. These stored readings allow you to reference current amplitudes to the voltage measurements of the DUT. 2182A 2 measures voltage across the DUT and stores the readings in its buffer.

For example, assume you want to measure DUT voltage at current sweep values of 10 µA, 20 µA, and 50 µA. When the sweep is started (10 µA output), 2182A 1 measures 1 mV (10 µA × 100 Ω = 1 mV) and stores the reading in its buffer at location 1. At the same time, 2182A 2 measures the DUT and stores that reading in its buffer at location 1. At the next sweep point (20 µA), 2182A 1 measures 2 mV and stores the reading in its buffer at location 2, and 2182A 2 measures the DUT and stores the reading in its buffer at location 2. At the last sweep point (50 µA), 2182A 1 measures 5 mV and stores the reading in its buffer at location 3, and 2182A 2 measures the DUT and stores the reading in its buffer at location 3.

The readings in the buffer of 2182A 1 correspond to the current sweep values. You can use the buffer location numbers to reference DUT readings to current amplitudes, shown in the following table.

2182A 1 buffer	**2182A 2 buffer**
RDG NO. 1 = 1 mV: 10 µA	RDG NO. 1 = DUT measurement
RDG NO. 2 = 2 mV: 20 µA	RDG NO. 2 = DUT measurement
RDG NO. 3 = 5 mV: 50 µA	RDG NO. 3 = DUT measurement

To cancel the effects of thermal EMFs in the test leads, the dc current reversal measurement technique must be used. By configuring a custom sweep, the SourceMeter can function as a bipolar growing amplitude source.

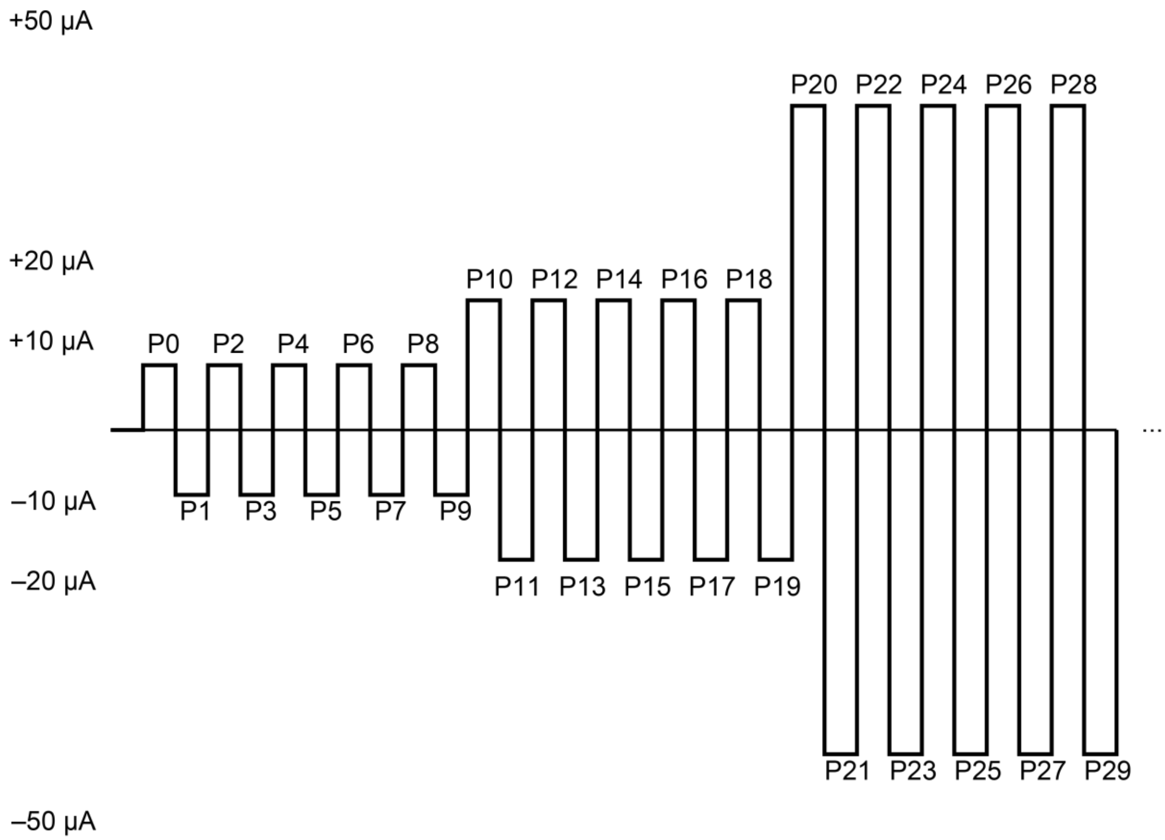
For example, if the test requires current steps of 10 µA, 20 µA, and 50 µA, the 6-point custom sweep would be configured as follows:

P0000 = +10 µA	P0001 = −10 µA
P0002 = +20 µA	P0003 = −20 µA
P0004 = +50 µA	P0005 = −50 µA

By enabling delta measurements on the 2182A, the effects of thermal EMFs in the test leads are automatically canceled during the source-measure process.

To check measurement repeatability, you can make more than one delta measurement at each current amplitude. In the following figure, the SourceMeter outputs five bipolar steps for each amplitude. The result is five delta measurements for each amplitude. When configuring the custom sweep, ±10 µA is assigned to the first 10 points of the sweep, ±20 µA is assigned to the next 10 points, and ±50 µA is assigned to the last 10 points. Therefore, the custom sweep is made up of 30 points (P0 through P29).

Figure 32: SourceMeter output: 30-point custom sweep



The procedure to use a SourceMeter and 2182A to make delta measurements is provided in Delta measurement procedure using a SourceMeter (on page 6-5). That procedure uses a 2-point custom sweep and must be modified for this application as follows:

- This application uses two 2182As that must be configured exactly the same.
- Change the trigger count to equal the number of points in the custom sweep. For example, if using the 30-point custom sweep, set the trigger count to 30.
- Assign current values to the sweep points. For the example 30-point sweep, the current values for points P0 through P29 are shown in the figure above.
- Set the buffers of both 2182As to store the delta readings. If performing a 30-point sweep, set both buffers to store 30 delta readings.
- Enable line synchronization on both 2182As. To access control, press **SHIFT** and then **LSYNC**.

NOTE

Optimum synchronization among all instruments is achieved when 2182A line synchronization is enabled and autozero is disabled. Autozero cannot be disabled from the front panel of the 2182A. When controlling this application over the bus, use the following commands for the 2182As:

```
:SYSTem:LSYNc:STATe ON    'Enable 2182A Line Synchronization
:SYSTem:AZERo:STATe OFF    'Disable 2182A Autozero
```

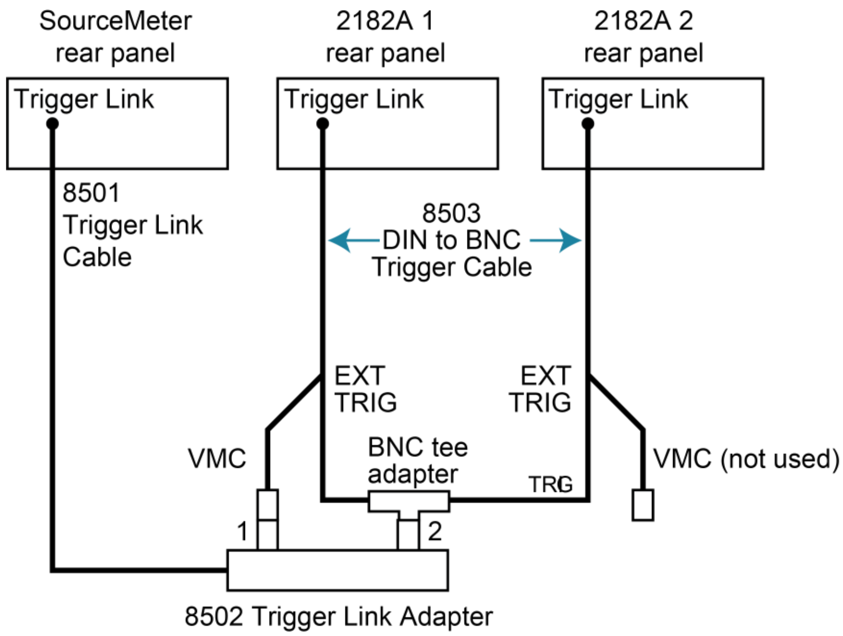
Autozero should only be disabled for short periods. After performing a sweep, re-enable Autozero. Refer to Autozero modes (on page 4-17) for details.

If your test requires currents that span two or more decades, you can configure the SourceMeter to output a log sweep. The 30-point sweep in the figure above is confined to the 10 µA to 100 µA decade. If you want to expand the sweep to span three current decades, use the next 30 sweep points for the 100 µA, 200 µA, and 500 µA amplitudes. The last 30 sweep points are used for the 1 mA, 2 mA, and 5 mA amplitudes. The I-V data points can then be graphed using a logarithmic/linear scale.

External triggering synchronizes source-measure operations among the instruments. The SourceMeter must trigger both 2182As to achieve simultaneous measurements. In turn, only one of the 2182As must trigger the SourceMeter to output the next source value.

The trigger link connections required for this application are shown in the following figure. The output trigger (VMC) is provided by 2182A 1. Do not use VMC from 2182A 2.

Figure 33: Trigger Link connections using two 2182A instruments



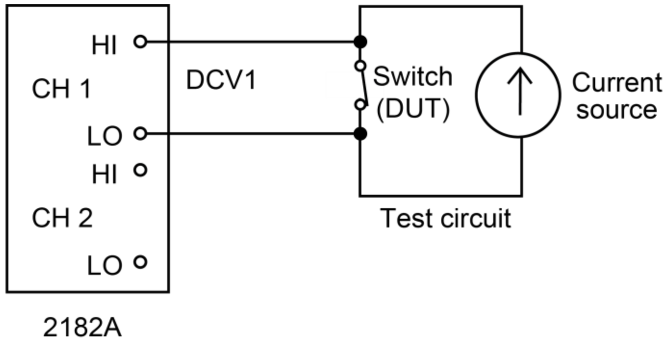
Low-resistance measurements

You can use the 2182A with a current source to measure resistances at levels below the capabilities of most conventional instruments. The following paragraphs discuss low-resistance measurement techniques and include some applications to test switches.

Low power switches

The following figure shows how to use the 2182A to measure the resistance of a switch contact. The constant current is provided by the current source, which can source up to 100 mA. To avoid oxide puncture, the voltage across the switch contact should be ≤20 mV. Voltage is limited by choosing a current that will not result in a voltage drop that is larger than 20 mV. For example, with a contact resistance specified at 500 mΩ, the current should be no larger than 40 mA.

Figure 34: Measuring switch contact resistance



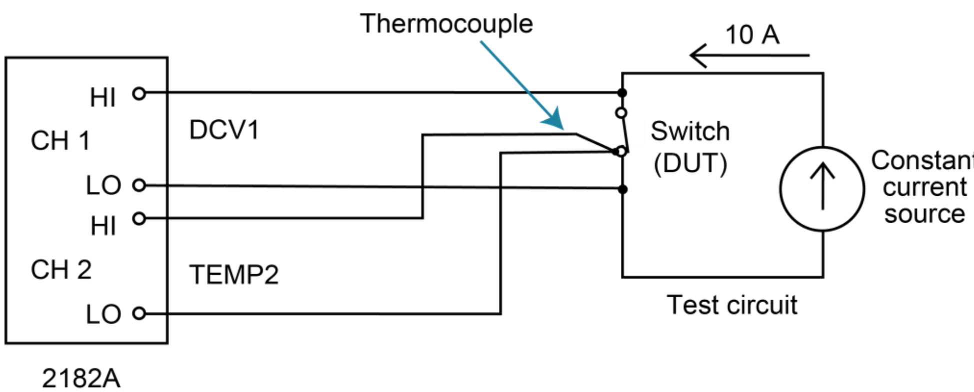
With current known and voltage measured, you can calculate resistance using Ohm's Law:

R = V/I

High power switches

Heat is a factor in high power switching. As the temperature of the switch increases, so does the contact resistance. In the following figure, heat is generated in the switch by sourcing a constant high current, such as 10 A, through it.

Figure 35: Measuring switch contact resistance and temperature



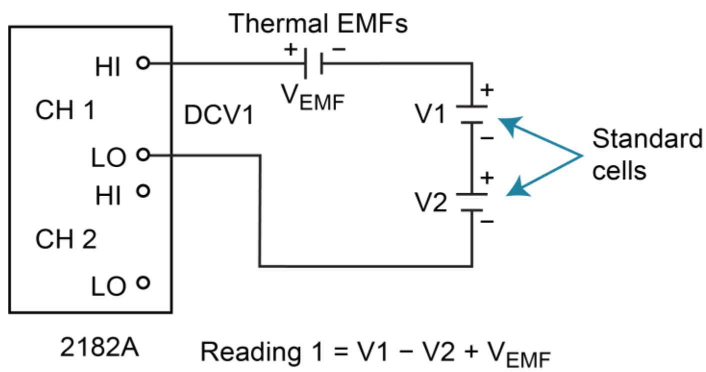
The 2182A measures both the voltage across the switch contact and the temperature. These measurements allow you to develop a resistance versus temperature profile. With current known and voltage measured, you can calculate resistance using Ohm's Law:

R = V/I

Standard cell comparisons

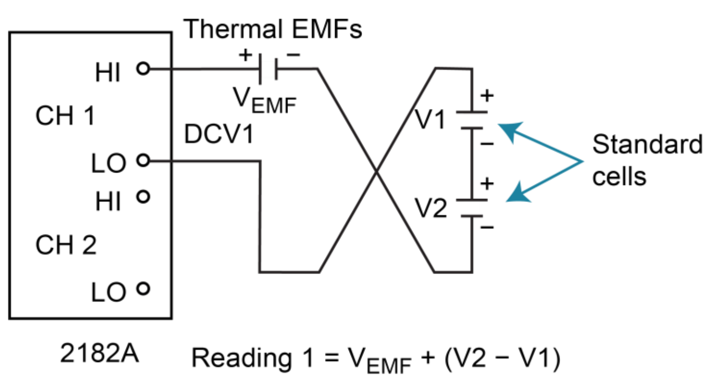
Standard cell comparisons are conducted by measuring the potential difference between a reference and an unknown standard cell. All cell differences are determined in series opposition configuration. The positive terminals of the standard cells (V1 and V2) are connected to the HI and LO inputs of the nanovoltmeter, as shown in the following figure. Use the Model 2107 Input Cable to connect the cells to the nanovoltmeter to minimize errors caused by thermal EMFs (V_{EMF}).

Figure 36: Standard cell comparison measurements - Reading 1



Once the measurement connections are made, care must be taken to avoid errors from thermally generated potentials. To minimize the effects of thermal EMFs, a second measurement is made with the nanovoltmeter leads reversed, as shown in the following figure. The small voltage difference is calculated by averaging the absolute values of the two readings. Calculation of standard deviation across several redundant readings helps provide this assurance.

Figure 37: Standard cell comparison measurements - Reading 2



Once stability is achieved, the actual voltage difference between the cells is measured. For each comparison, several readings are usually averaged. This process of comparing is then repeated each week, month, or year, depending upon the standards laboratory. The results can then be plotted and compared over time.

Section 5

Measurement considerations

In this section:

Introduction

For sensitive measurements, external considerations affect accuracy. Effects that are not noticeable when working with higher voltages are significant in nanovolt signals.

The 2182A reads only the signal received at its input. Therefore, it is important that this signal be properly transmitted from the source.

Some of the phenomena that can cause unwanted noise include thermoelectric effects (thermocouple action), source resistance noise, magnetic fields, and radio frequency interference. The following paragraphs discuss the most important of these effects and ways to minimize them.

NOTE

For comprehensive information on low-level measurements, see the *Low Level Measurements Handbook*, which is available from tek.com/keithley.

Thermoelectric potentials

Thermoelectric potentials, or thermoelectric EMFs, are the most common source of errors in low-voltage measurements. These small electric potentials are generated when different parts of the circuit are at different temperatures and when conductors made of dissimilar metals are joined.

Thermoelectric EMFs can cause the following conditions:

- Instability or zero offset is much higher than expected.
- The reading is sensitive to and responds to temperature changes. This effect can be demonstrated by placing a heat source near the circuit or by observing changes because of sunlight or the activation of heating and air conditioning systems.

The following paragraphs discuss how thermoelectric potentials are generated and ways to minimize their effects.

Thermoelectric coefficients

The following table shows the magnitude of thermoelectric EMFs that are generated for different materials.

Material thermoelectric coefficients	
Material	**Thermoelectric potential**
Copper-to-copper	0.2 µV/°C
Copper-to-silver	0.3 µV/°C
Copper-to-gold	0.3 µV/°C
Copper-to-cadmium/tin	0.3 µV/°C
Copper-to-lead/tin	1 µV/°C to 3 µV/°C
Copper-to-Kovar®	40 µV/°C to 75 µV/°C
Copper-to-silicon	400 µV/°C
Copper-to-copper oxide	1000 µV/°C

Thermoelectric generation

The following figure shows a representation of how thermal EMFs are generated. The test leads are made of the A material, while the source under test is the B material. The temperatures between the junctions are shown as T1 and T2. To determine the thermal EMF generated, the following relationship may be used:

$E_T = Q_{AB} (T_1 - T_2)$

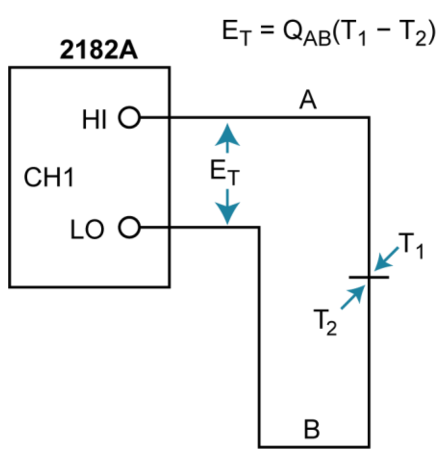
Where:

- E_T = Generated thermal EMF
- Q_{AB} = Thermoelectric coefficient of material A with respect to material B (µV/°C)
- T_1 = Temperature of B junction (°C or K)
- T_2 = Temperature of A junction (°C or K)

In the unlikely event that the two junction temperatures are identical, no thermal EMFs are generated. More often, the two junction temperatures differ and considerable thermal EMFs are generated.

A typical test setup may have several copper-to-copper junctions. Each junction can have a thermoelectric coefficient as high as 0.2 µV/°C. Since the two materials frequently have a several degree temperature differential, several microvolts of thermal potentials can be generated even if reasonable precautions are taken.

Figure 38: Thermal EMF generation



Minimizing thermal EMFs

To minimize thermal EMFs, use only copper wires, lugs, and test leads for the entire test setup. Also, it is imperative that all connecting surfaces are kept clean and free of oxides. As noted in Thermoelectric generation (on page 5-3), copper-to-copper oxide junctions can result in thermal EMFs as high as 1 mV/°C.

Even when low-thermal cables and connections are used, thermal EMFs can still be a problem in some cases. It is especially important to keep the materials that form the junction at the same temperature. One way to minimize thermal problems is to keep the two junctions close together. Also, keep all junctions away from air currents; in some cases, it may be necessary to thermally insulate sensitive junctions to minimize temperature variations. When a copper-to-copper connection is made, sufficient pressure must be applied to ensure the connection is gastight to prevent future oxidation.

In some cases, connecting the two thermal junctions together with good thermal contact to a common heat sink may be required. Unfortunately, most good electrical insulators are poor conductors of heat. In cases where such low thermal conductivity may be a problem, special insulators that combine high electrical insulating properties with high thermal conductivity may be used. Some examples of these materials include hard anodized aluminum, sapphire, and diamond.

Canceling residual thermal offsets

Even if all reasonable precautions are taken, some residual thermal offsets may still be present. These offsets can be minimized by using the 2182A relative offset feature to cancel them out. For additional information on using the relative offset feature, refer to Relative offset (on page 4-25).

If the offset voltage varies, use the dc current-reversal technique instead of relative offset. The dc current-reversal technique requires a source that can output currents equal in magnitude but opposite in polarity. In general, a voltage measurement is made on the positive and negative alternations of the current source. The averaged difference of those two readings cancels the thermal EMF component of the measurements. The 2182A can automatically make the measurements and calculate and display the result by using the delta measurement mode.

For more information on using relative offset, refer to Relative offset (on page 4-25). For information on using the dc current-reversal technique, refer to Delta (on page 6-1).

To use relative offset to cancel the residual thermal offsets:

1. Place the instrument on the 3 mV range.
2. Disconnect the cable from the source to avoid shorting out the source.
3. Short the end of the connecting cable nearest the measured source.
4. Allow the reading to settle.
5. Press the **REL** key.
6. Select the appropriate range and make your measurement as usual.

AC pick-up noise

AC voltages that are extremely large compared with the dc signal to be measured may be induced into the input of the 2182A and corrupt the measurement. AC interference can cause the 2182A to behave in one or more of the following ways:

- Unexpected offset voltages
- Inconsistent readings between ranges
- Sudden shifts in a reading

To minimize ac pick-up, keep the test circuit source and the 2182A away from strong ac magnetic sources. The voltage induced due to magnetic flux is proportional to the area of the loop formed by the input leads. Therefore, minimize the loop area of the input leads and connect each signal at only one point.

Shielding also helps minimize ac interference. Refer to Shielding (on page 5-9) for more information.

Source resistance noise

Noise present in the source resistance is often the limiting factor in the ultimate resolution and accuracy of 2182A measurements. The following topics discuss the generation of Johnson noise and ways to minimize such noise.

Johnson noise equation

The amount of noise present in a given resistance is defined by the Johnson noise equation as follows:

$$E_{RMS} = \sqrt{4kTRF}$$

Where:

- E_{RMS} = The RMS value of the noise voltage
- k = Boltzmann constant (1.38×10^{-23} joules/Kelvin)
- T = Temperature (K)
- R = Source resistance (ohms)
- F = Noise bandwidth (Hz)

At a room temperature of 293 K (20 °C), the equation simplifies to:

$$E_{RMS} = 1.27 \times 10^{-10} \sqrt{RF}$$

Since the peak-to-peak noise is five times the RMS value 99% of the time, the peak-to-peak noise can be equated as follows:

$$E_{p\text{-}p} = 6.35 \times 10^{-10} \sqrt{RF}$$

For example, with a source resistance of 10 kΩ, the noise over a 0.5 Hz bandwidth at room temperature is:

$$E_{p\text{-}p} = 6.35 \times 10^{-10} \sqrt{(10 \times 10^{3})(0.5)}$$
$$E_{p\text{-}p} = 45 \text{ nV}$$

Minimizing source resistance noise

You can reduce the noise in several ways:

- Reduce the source resistance
- Lower the temperature
- Narrow the bandwidth

Of these, lowering the resistance is the least practical because the signal voltage is reduced more than the noise. For example, decreasing the resistance of a current shunt by a factor of 100 also reduces the voltage by a factor of 100, but the noise is decreased only by a factor of 10.

Cooling the source is often the only practical method available to reduce noise. Again, however, the available reduction is not as large as it might seem because the reduction is related to the square root of the change in temperature. For example, to cut the noise in half, the temperature must be decreased from 293 K to 73.25 K.

Magnetic fields

When a conductor loop cuts through magnetic lines of force, a very small current is generated. This phenomenon can cause unwanted signals to occur in the test leads of a test system. If the conductor has sufficient length or cross-sectional area, even weak magnetic fields can create signals that affect low-level measurements.

To reduce these effects:

- Reduce the lengths of the connecting cables.
- Minimize the exposed circuit area.
- Change the orientation of the leads or cables.
- Minimize cable loop area or introduce cable twisting.

In extreme cases, you may require magnetic shielding. Special metal with high permeability at low flux densities (such as mu metal) is effective at reducing these effects.

Even when the conductor is stationary, you may have problems with magnetically-induced signals. Fields can be produced by sources such as the ac power line voltage and large inductors, such as power transformers. Keep the 2182A voltage source and connecting cables away from these potential noise sources.

Radio frequency interference

Radio frequency interference (RFI) is a general term used to describe electromagnetic interference over a wide range of frequencies. RFI can be particularly troublesome at low signal levels, but it can also affect measurements at high levels if the fields are of sufficient magnitude.

Some causes of RFI are:

- Steady-state sources, such as radio or TV signals.
- Some types of electronic equipment, such as microprocessors and high-speed digital circuits.
- Impulse sources, such as arcing in high-voltage environments.

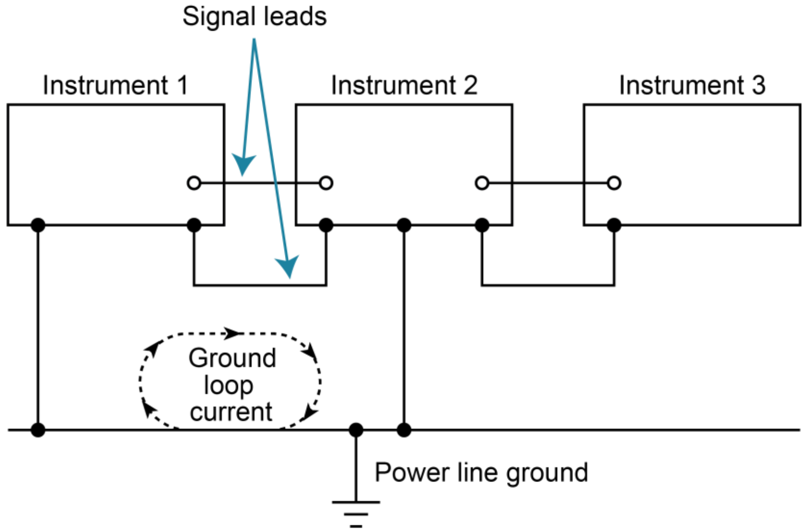
RFI can be minimized in several ways. The most obvious method is to keep the 2182A voltage source and signal leads as far away from the RFI source as possible. Additional shielding of the instrument, signal leads, sources, and other measuring instruments will often reduce RFI to an acceptable level. In extreme cases, a specially-constructed screen room may be required to sufficiently attenuate the RFI signal.

The 2182A digital filter may help to reduce RFI effects in some situations. In other cases, additional external filtering may also be required. However, filtering may have detrimental effects on the signal, such as increasing settling time.

Ground loops

When two or more instruments are connected, avoid unwanted signals caused by ground loops. Ground loops usually occur when sensitive instrumentation is connected to other instrumentation with more than one signal return path, such as power line ground. As shown in the following figure, the resulting ground loop causes current to flow through the instrument LO signal leads and then back through power line ground. This circulating current develops a small voltage between the LO terminals of the two instruments. This voltage is added to the source voltage, affecting the accuracy of the measurement.

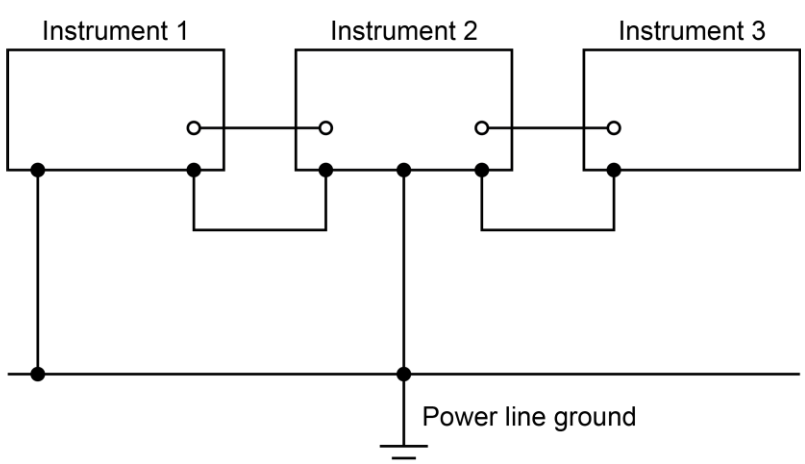
Figure 39: Power-line ground loops



The following figure shows how to connect several instruments together to eliminate this type of ground loop problem. Here, only one instrument is connected to power line ground.

Ground loops are not normally a problem with instruments like the 2182A that have isolated LO terminals. However, other instruments in the test setup may not have isolated terminals. Consult the documentation for each instrument in the test setup.

Figure 40: Eliminating ground loops



Shielding

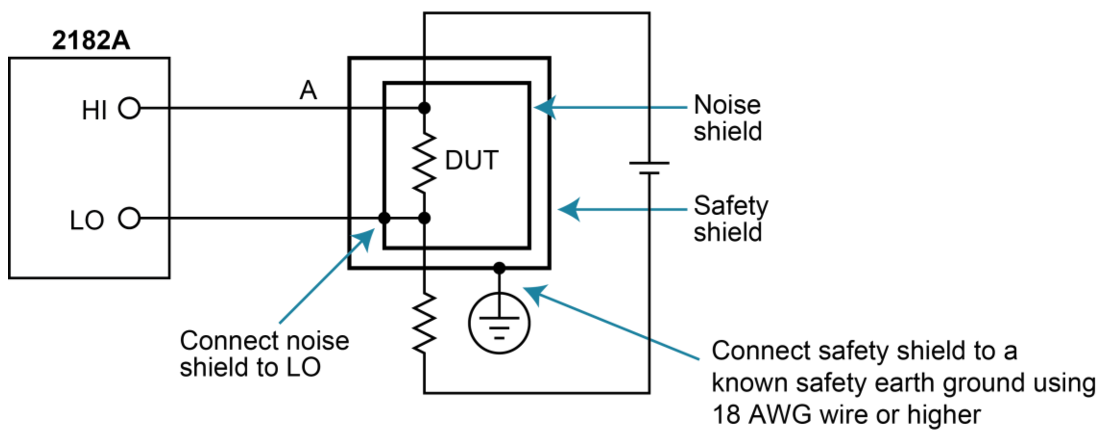
Proper shielding of all signal paths and sources being measured is important to minimize noise pickup in virtually any low-level measurement situation. Otherwise, interference from noise sources such as line frequency and RF fields can seriously corrupt measurements.

To minimize noise, a closed metal shield surrounding the source may be necessary, as shown in the following figure. In most cases, this shield should be connected to input LO. In some situations, better noise performance may result with the shield connected to chassis ground. The metal shield should enclose the test circuit.

WARNING

Do not float input LO more than 30 V_{RMS}, 42.4 V_{PEAK} above earth ground with an exposed shield connected to input LO. To avoid a possible shock hazard, surround the LO shield with a second safety shield that is insulated from the inner shield. Connect this safety shield to safety earth ground using #18 AWG minimum wire before use. Failure to adhere to these guidelines can result in personal injury or death due to electric shock.

Figure 41: Shielding example

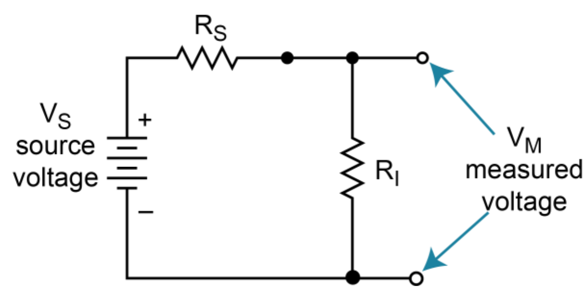


Meter loading

Loading of the voltage source by the 2182A becomes a consideration for high-source resistance values. As the source resistance increases, the error caused by meter loading increases.

The following figure shows the method used to determine the percent error due to meter loading. The voltage source, V_S, has a source resistance, R_S, while the input resistance of the 2182A is R_I, and the voltage measured by the nanovoltmeter is V_M.

Figure 42: Meter loading



The voltage actually measured by the meter is attenuated by the voltage divider action of R_S and R_I, and it can be calculated as follows:

$$V_M = \frac{V_S R_L}{R_I + R_S}$$

This relationship can be modified to directly compute for percent error:

$$\text{Percent error} = \frac{100R_S}{R_I + R_S}$$

From this equation, the input resistance of the 2182A must be at least 999 times the value of source resistance to keep loading error within 0.1%.

Delta, pulse delta, and differential conductance

In this section:

Overview

The delta measurement mode provides the measurements and calculation for the dc current-reversal technique. In this technique, a voltage measurement is made on the positive and negative alternations of the current source. The averaged difference of those two readings cancels the effects of thermal EMFs in the test leads.

Each delta reading is calculated from two voltage measurements on Channel 1.

Basic delta calculation:

$$\text{Delta} = \frac{\text{V1t1} - \text{V1t2}}{2}$$

Where:

- V1t1 is the voltage measurement on the positive phase of the current source.
- V1t2 is the voltage measurement on the negative phase of the current source.

Delta calculation using filter and relative offset:

$$\text{Delta} = \frac{\text{FiltV1t1} - \text{FiltV1t2}}{2} - \text{RelV1}$$

Where:

- Filt V1t1 and Filt V1t2 are filtered voltage measurements on the positive and negative phases of the current source. The `FILT` annunciator is on when the filter is enabled.
- RelV1 is the relative offset value established for DCV1. The `REL` annunciator is on when the relative offset is enabled.

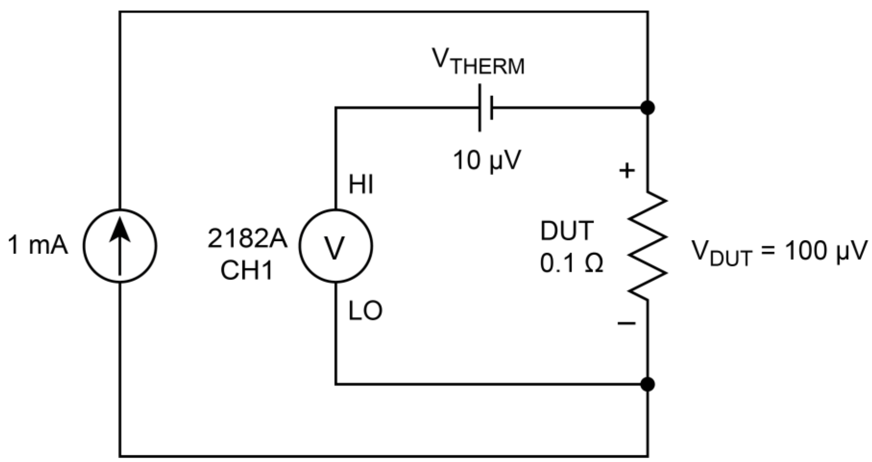
The 2182A is optimized to provide low-noise readings when measurement speed is set from 1 PLC to 5 PLC. At 1 PLC, current can be reversed after 100 ms. At 5 PLC, current can be reversed after 333 ms. At these reading rates, noise induced by the power line should be insignificant. You can use filtering to reduce peak-to-peak reading variations. For more information on the filter in regard to delta measurements, see Filter considerations (on page 6-5).

The following example shows how to use a bipolar current source and delta to cancel the effects of thermal EMFs.

In the following figure, a constant 1 mA is sourced to a 0.1 Ω DUT. Under ideal conditions, the 2182A measures 100 µV across the DUT (1 mA × 0.1 Ω = 100 µV). However, connection points and temperature fluctuations may generate thermal EMFs in the test leads (thermal EMFs drift with temperature). The figure shows 10 µV of thermal EMF (V_{THERM}). Therefore, the 2182A measures 110 µV instead of 100 µV:

$$
\begin{aligned}
V_{2182A} &= V_{THERM} + V_{DUT} \\
&= 10\ \mu V + 100\ \mu V \\
&= 110\ \mu V
\end{aligned}
$$

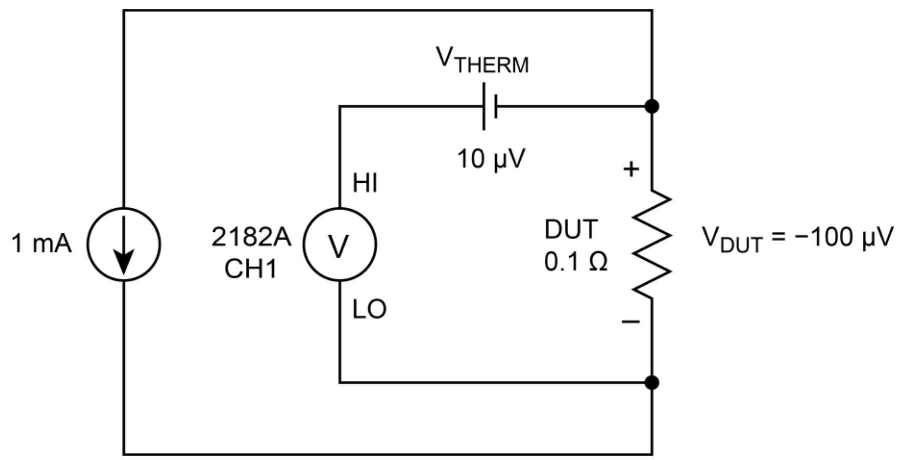
Figure 43: Test circuit using constant positive current source



The following figure shows what happens when the current is reversed. The measurement by the 2182A still includes the 10 µV of thermal EMF, but the voltage across the DUT is now negative. Therefore, the 2182A measures 90 µV:

$$
\begin{aligned}
V_{2182A} &= V_{THERM} - V_{DUT} \\
&= 10\ \mu V - 100\ \mu V \\
&= -90\ \mu V
\end{aligned}
$$

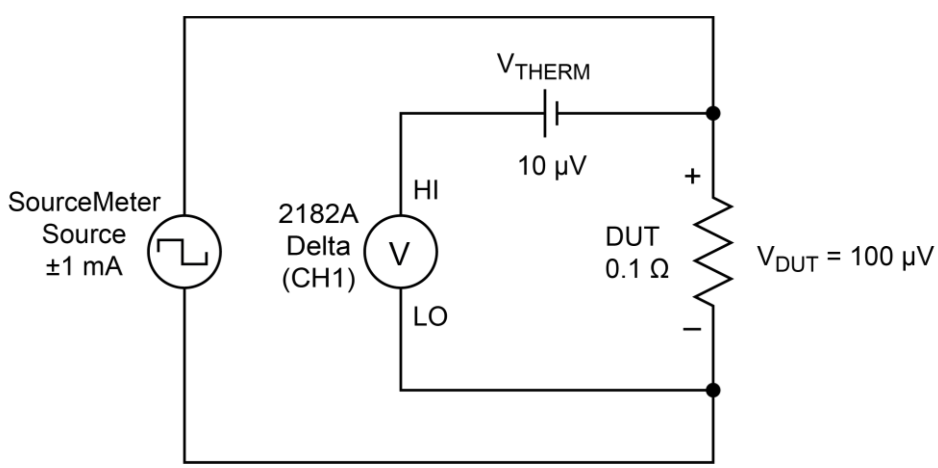
Figure 44: Test circuit using constant negative current source



Neither measurement by the 2182A accurately measured the voltage across the DUT. However, if you take a simple average of the magnitudes of the two readings (110 µV and 90 µV), the result is 100 µV, which is the actual voltage drop across the DUT. This is what the calculation for delta does.

To use the dc current-reversal technique, replace the constant current source with a bipolar current source, as shown in the following figure. The RATE must be set to 1 PLC or 5 PLC to optimize measurement performance. At 1 PLC or 5 PLC, delta measurements cancel thermal EMFs to a <50 nV level.

Figure 45: Delta measurement using bipolar source



The current source will alternate between +1 mA and −1 mA. When using delta, the 2182A makes the first voltage measurement (V1t1) while sourcing +1 mA. The second voltage measurement (V1t2) is performed while sourcing −1 mA:

$$
\begin{aligned}
V1t1 &= V_{THERM} + V_{DUT} \\
&= 10\ \mu V + 100\ \mu V \\
&= 110\ \mu V \\
V1t2 &= V_{THERM} - V_{DUT} \\
&= 10\ \mu V - 100\ \mu V \\
&= -90\ \mu V
\end{aligned}
$$

Delta is then calculated as follows:

At +1 mA:

$V1t1 = 10\ \mu V + 100\ \mu V = 110\ \mu V$

At −1 mA:

$V1t2 = 10\ \mu V - 100\ \mu V = -90\ \mu V$

$$V_{DELTA} = \frac{V1t1 - V1t2}{2} = \frac{110\ \mu V - (-90\ \mu V)}{2} = 100\ \mu V$$

$$V_{DELTA} = V_{DUT}$$

Using delta with a bipolar source effectively canceled the 10 µV thermal EMF.

External triggering is required to control the timing between voltage measurements and current source reversals. Trigger synchronization between the source and the 2182A is explained in Model 2182A and SourceMeter trigger synchronization (on page 6-9).

Enabling the Delta measurement mode

Delta measurements by the 2182A require the use of an alternating polarity source. The source must have external triggering capabilities that are compatible with the external triggering capabilities of the 2182A. See Delta measurement procedure using a SourceMeter (on page 6-5) for an example of how to use a Keithley SourceMeter with the 2182A to perform delta measurements.

When the Delta measurement mode is enabled:

- The reading is followed by a `d`.
- If the 2182A is on Channel 2, it is automatically set to Channel 1.

Delta is not available when stepping or scanning is enabled.

Reading hold cannot be used with the delta feature.

You can store delta readings in the buffer. Refer to Reading buffers (on page 8-1) for more information.

NOTE

To increase the speed of delta measurements, disable Front Autozero. See Autozero modes (on page 4-17) for more detail.

To enable delta:

Press the **SHIFT** key and then the **V1−V2** key. The display shows `(Vt1-Vt2) / 2` briefly before displaying the result of the calculation.

To enable a delta reading using a SCPI command, refer to :SENSe:VOLTage[:DC]:DELTa (on page 13-61).

To disable the Delta feature:

Select a single measurement function (**DCV1**, **DCV2**, **TEMP 1**, or **TEMP 2**) or select **Ratio**.

Filter considerations for delta measurements

The filter configuration for DCV1 is applied separately to each measurement phase (V1t1 and V1t2) of the delta process. When the delta feature is selected, the moving filter is automatically selected and cannot be changed until the delta feature is disabled.

After filtering yields a reading for V1t1, an output trigger is sent. After filtering yields a reading for V1t2, another output trigger is sent. The delta calculation is performed and the reading is displayed.

For example, assume the filter count for the moving filter is 5. Two filter stacks are used, one for V1t1 readings and one for V1t2 readings. The filter stack for the V1t1 readings is filled with five measurement conversions. The five readings are averaged to yield the V1t1 value and an output trigger is sent on each V1t1 A/D conversion. The filter stack for V1t2 readings is then filled with five measurement conversions. The five readings are averaged to yield the V1t2 value and an output trigger is sent. On each V1t2 A/D conversion, the delta calculation is performed using the filtered V1t1 and V1t2 values and the result is displayed on the 2182A.

For every subsequent delta measurement, the operation is basically the same except that each stack only requires one reading to fill it. The oldest reading in each stack is discarded.

Delta measurement procedure using a SourceMeter

You can use a Keithley SourceMeter (2400, 2410, or 2420) as a bipolar source by configuring it to perform a custom sweep. In general, a custom sweep is made up of a number of specified source points. To provide current reversal, the positive current values are assigned to the even-numbered points and the negative current values are assigned to the odd-numbered points. For details on custom sweep, see the documentation for the SourceMeter.

Applications that use delta measurements require either a fixed current or a growing amplitude current. When a fixed current is required, you can configure the SourceMeter to output a bipolar 2-point custom sweep. That sweep can be run a specified number of times or it can run continuously. For example, if a fixed current of 1 mA is required for the test, the two bipolar sweep points for the custom sweep are +1 mA and −1 mA.

When a growing-amplitude current is required, the custom sweep can be configured to include all the current values required for the test. For example, assume the test requires two delta measurements at each of three current levels: 1 mA, 2 mA, and 5 mA. That test requires the following 12-point custom sweep to produce the six delta measurements:

P0000 = +1 mA	P0001 = −1 mA
P0002 = +1 mA	P0003 = −1 mA
P0004 = +2 mA	P0005 = −2 mA
P0006 = +2 mA	P0007 = −2 mA
P0008 = +5 mA	P0009 = −5 mA
P0010 = +5 mA	P0011 = −5 mA

The following procedure uses the SourceMeter as a bipolar fixed-amplitude current source. It outputs a 2-point custom sweep to provide current reversal that is required for delta measurements by the 2182A.

When making delta measurements:

- Set the 2182A RATE to 1 PLC or 5 PLC to optimize measurement performance. At 1 PLC or 5 PLC, delta measurements cancel thermal EMFs to a <50 nV level.
- Set the SourceMeter SPEED to FAST (0.01 PLC). Using a slower speed results in trigger synchronization problems with the 2182A.

NOTE

This procedure assumes that the 2182A is using the factory default Trigger Link line configuration: Line 1 is VMC (output), Line 2 is EXT TRIG (input).

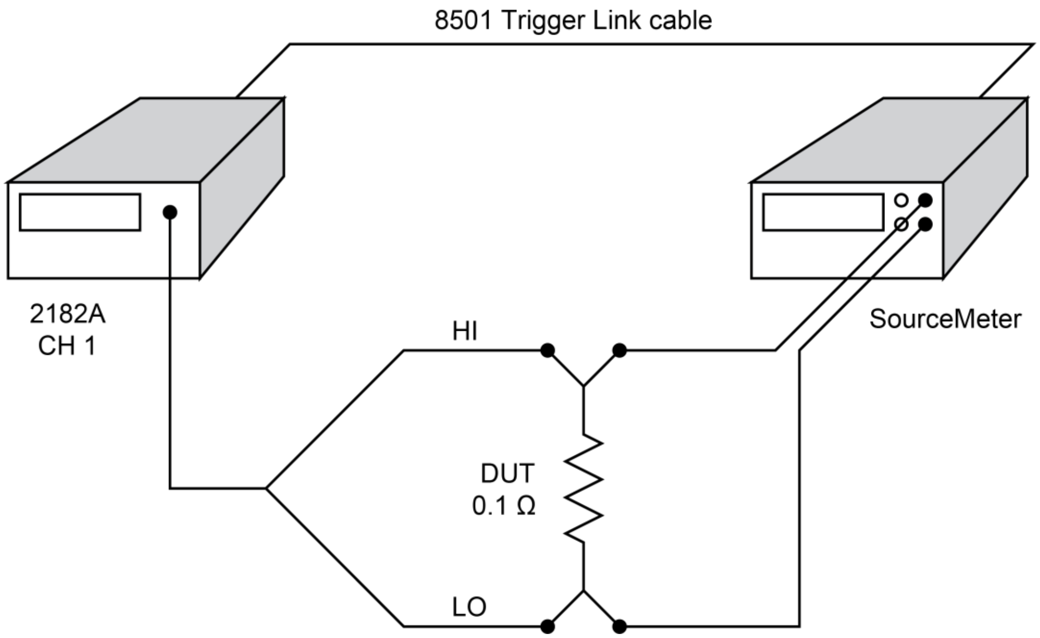
Setting up a delta measurement

This procedure assumes that the 2182A is using the factory default Trigger Link line configuration, where Line 1 is VMC (output) and Line 2 is EXT TRIG (input).

To set up a delta measurement on the 2182A:

1. Connect the SourceMeter and 2182A to the DUT as shown in the following figure.
2. Connect an 8501 Trigger Link cable from the 2182A to the SourceMeter.

Figure 46: Delta measurement connections



3. Reset the 2182A by selecting **MAIN MENU > SAVESETUP > GLOBAL > RESET > BENCH**.
4. Set the **CONFIG** and then **TRIG**.
5. Configure the arm layer of the trigger model as follows:
   - Arm-In Event: **IMMEDIATE**
   - Arm-Out TLink Line: **#3**
   - Arm-Out Events: **OFF**
   - Arm Count: **INFINITE**
6. Configure trigger-in of the trigger layer as follows:
   - Trigger-In Source: **TRIGGER LINK**
   - Trig-In TLink Line: **#1**
   - Event Detect Bypass: **NEVER**
   - Trigger In Events: **SOURCE=ON DELAY=OFF MEAS=OFF**
7. Configure trigger-out of the trigger layer as follows:
   - Trig-Out TLink Line: #2
   - Trigger Out Events: **SOURCE=ON DELAY=OFF MEAS=OFF**
   - Delay: **000.0000 s**
   - Trig-Count: **0002** (trigger count must equal the number of sweep points)

NOTE

Do not set a delay on the SourceMeter. This may adversely affect trigger synchronization between the SourceMeter and the 2182A.

To set up the SourceMeter for a two-point custom sweep:

1. On the SourceMeter, select **Source I and Measure V**.
2. Select an appropriate current source range. For example, if your current reversal values are going to ±1 mA, select the 1 mA source range.
3. Press **SPEED** and select **FAST**. The SourceMeter must run as fast as possible to avoid synchronization problems with the 2182A.
4. On the SourceMeter, set up a two-point custom sweep. Select **CONFIG** and then **SWEEP**.
5. From the menu, select **TYPE** and then select **CUSTOM**.
6. Set the **#-POINTS** to **2**, and set **ADJUST-POINTS** (P0000 and P0001) to the positive and negative current source values. For example, if the test requires 1 mA, set P0000 to +1 mA and set P0001 to −1 mA.
7. Also from the sweep configuration menu, specify the number of two-point sweeps to perform:
   - To continuously source the current reversal sweep: Select **INFINITE**.
   - To perform a specific number of two-point sweeps: Select **FINITE**.

To set up the 2182A:

1. Press **RESTR** and select **FACT** to return the nanovoltmeter to its factory default conditions.
2. Press **RATE** and select **1 PLC** (MED annunciator on) or **5 PLC** (SLOW annunciator on).
3. Enable delta measurements by pressing **SHIFT** and then **V1−V2**.
4. Disable Front Autozero by setting **SHIFT > CONFIG > FRONT AUTOZERO** to **N**.
5. Press **EX TRIG** to place the instrument in the external trigger mode. This halts measurements.
6. If a longer settling time is required before making each measurement, set a manual delay from the 2182A. Press **SHIFT** and then **DELAY** to set the delay.
7. Press **STORE** and set the number of delta readings to store in the buffer. Press **ENTER** to enable the buffer.

Run the sweep from the SourceMeter:

1. Turn on the SourceMeter output by pressing the **OUTPUT ON/OFF** key (the `ARM` annunciator turns on).
2. Press **CONFIG** and then **TRIG** to access the trigger configuration menu.
3. Select **HALT** to place the SourceMeter in the idle state (`ARM` annunciator turns off).
4. Press **EXIT** to return to the normal display.
5. To arm the sweep, press **SWEEP**.

6. To start the sweep, press **TRIG**. At the end of each two-point sweep, a delta reading is calculated and displayed. If the buffer is enabled, the reading is stored in the buffer.
7. Turn off the source by pressing the **ON/OFF OUTPUT** key on the SourceMeter. Turning the source off prevents heat from building up in the DUT.
8. On the 2182A, to access delta readings stored in the buffer, press **RECALL**.
9. To repeat the sweep, on the 2182A, press **EX TRIG** twice to disable and then re-enable external triggering. The TRIG annunciator indicates that the 2182A is in the external triggering mode. Repeat steps 1 to 8.

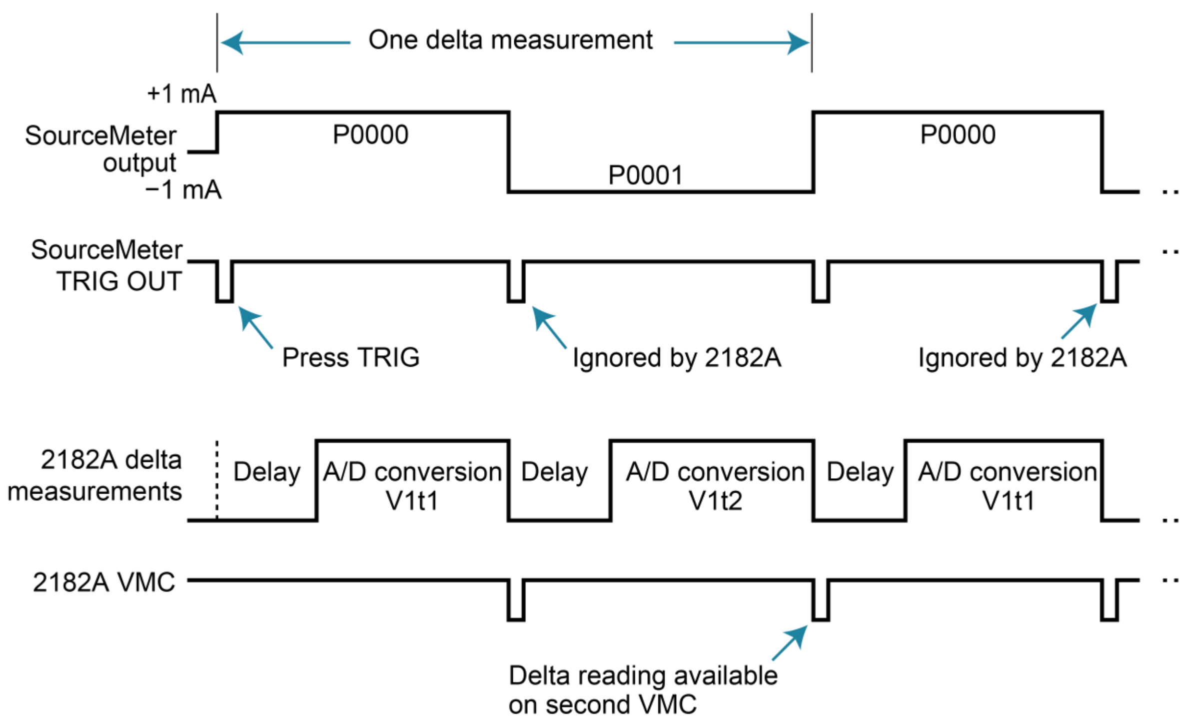
NOTE

To abort a sweep in progress, press **EXIT** on the SourceMeter.

Model 2182A and SourceMeter trigger synchronization

The timing diagram in the following figure shows trigger synchronization between the SourceMeter and the 2182A for a two-point custom sweep. As shown in the timing diagram, the SourceMeter outputs a trigger after every source sweep point, and the 2182A outputs a trigger after every A/D conversion.

Figure 47: Triggering timing diagram



When the TRIG key on the SourceMeter is pressed, it starts the sweep (outputs +1 mA) and triggers the 2182A to start a delta measurement. After the delay period, the 2182A performs an A/D conversion for the V1t1 phase of the delta measurement, and then triggers the SourceMeter to output the second point of the sweep (−1 mA). At this point, the 2182A does not wait for the return trigger from the SourceMeter to perform the V1t2 phase of the delta measurement. The second trigger from the SourceMeter is ignored. After the A/D conversion for the V1t2 phase, the delta reading is calculated and displayed.

If programmed for another sweep, the trigger from the 2182A to the SourceMeter starts the sweep again to make another delta measurement.

Delta measurement procedure using a Model 622x

With the use of a bipolar current source, the 2182A can perform basic delta measurements. See Delta (on page 6-1) for details on basic delta measurements.

This section summarizes the enhanced delta, pulse delta, and differential conductance measurement processes that can be performed with the use of the Keithley Model 622x Current Source. It does not provide the procedures to configure and make these measurements.

Detailed information on all aspects of delta, pulse delta, and differential conductance operation are provided in the *Model 622x Reference Manual*. An abbreviated version of this information is provided in the *Model 622x User's Manual*.

You can use the example software that is available at tek.com/keithley as a learning tool to configure and run delta, pulse delta, and differential conductance. For details, see “Using the example software” in the *Model 622x Reference Manual*.

Keithley instrumentation requirements

Keithley instrumentation requirements for delta, pulse delta, and differential conductance are:

- **Models 6220 and 2182A:** Delta and differential conductance measurements.
- **Models 6221 and 2182A:** Delta, pulse delta, and differential conductance measurements.

NOTE

The firmware version of the 2182A must be C01 or higher.

Operation overview

The Model 6220 or 6221 Current Source can be used with a 2182A to perform delta and differential conductance. The 2182A and 6221 combination can also perform pulse delta. These operations use a delta current-reversal technique to cancel the effects of thermal EMFs.

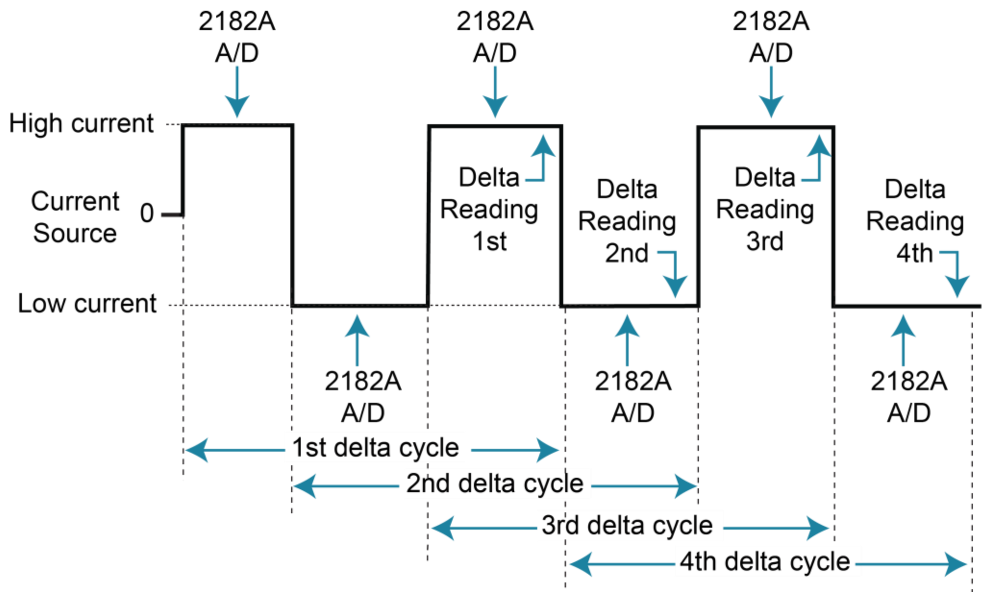
The 6220 or 6221 provides a bipolar output current and the 2182A performs A/D conversions (measurements) at source high and source low points. An averaging algorithm is then used to calculate the delta reading.

Delta

The 6220 or 6221 provides a square-wave current output, and the 2182A performs A/D conversions (measurements) at each high and low output level. A three-point moving-average algorithm is used to calculate delta readings.

As shown in the following figure, the first three 2182A A/D conversions (measurements) yield the first delta reading. Each subsequent 2182A measurement yields a single delta reading. Every delta reading uses the three previous measurements to calculate delta.

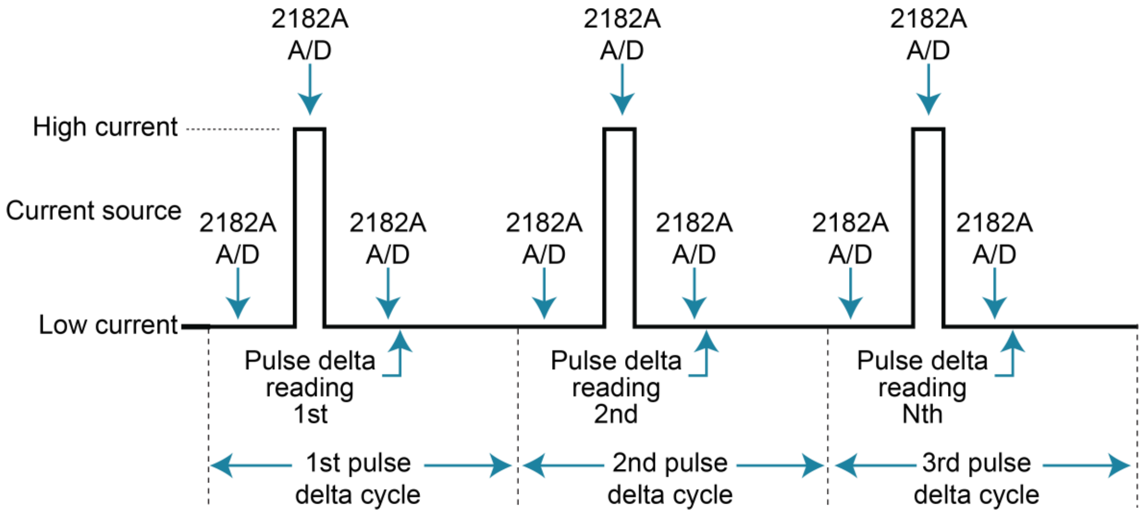
Figure 48: Delta measurements



Pulse delta

The 6221 outputs pulses and uses three-point repeating-average measurements to calculate pulse delta voltage. For each pulse, the 2182A performs an A/D conversion (measurement) at pulse low, pulse high, and pulse low. Each set of three A/D readings yields a single pulse delta reading. The following figure shows pulse delta measurements. If device heating is a concern, you can use two-point measurements instead (the second low pulse is not measured due to corruption from heat).

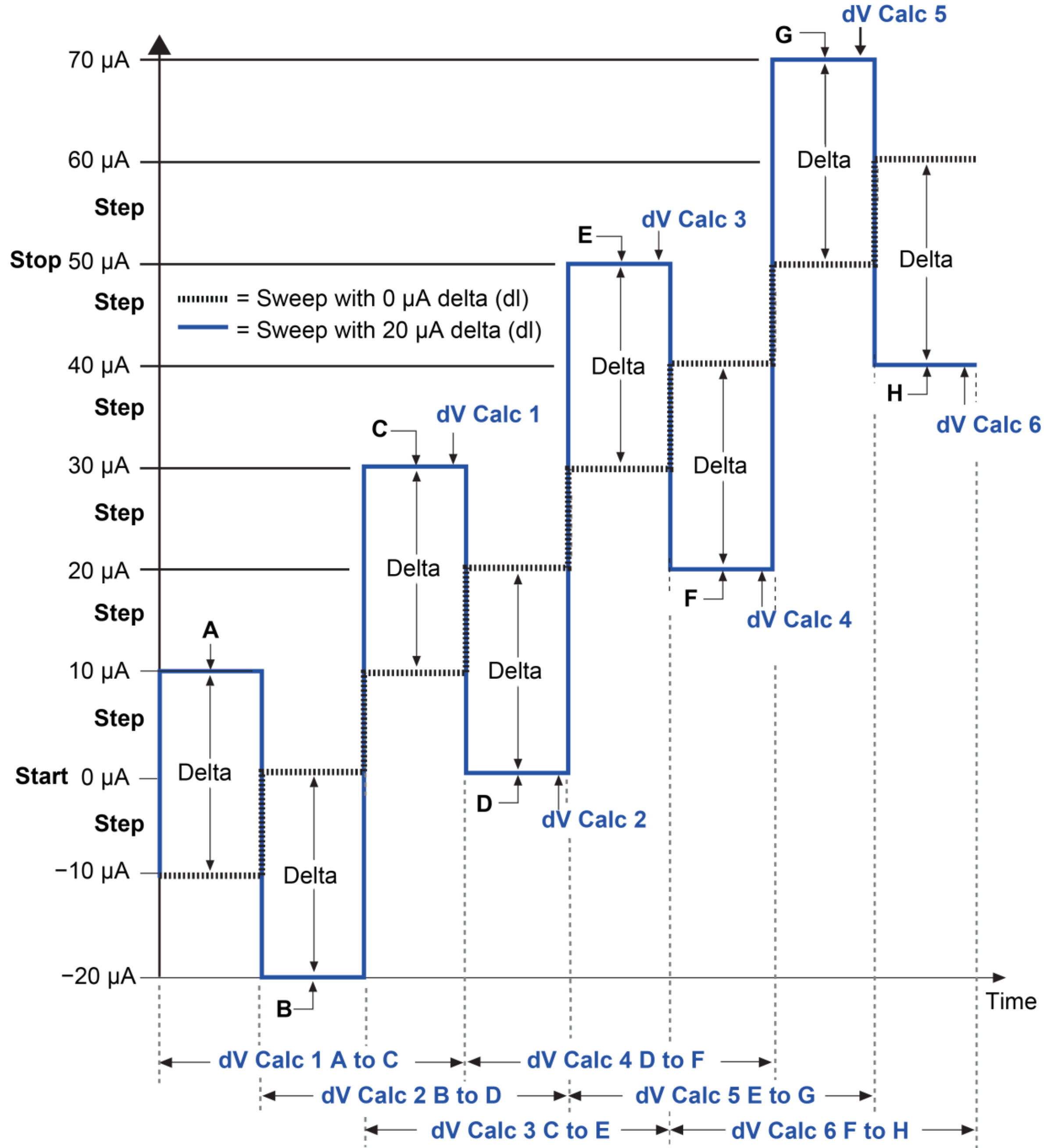
Figure 49: Pulse delta measurements



Differential conductance

The 6220 or 6221 outputs a differential current (dI) sweep and measures differential voltage (dV). This function uses a three-point moving average algorithm to calculate dV. With dI known and dV calculated, the 6220 and 6221 can then calculate differential conductance (dG) or differential resistance (dR). The following figure shows differential conductance measurements.

Figure 50: Differential conductance measurements



Test system configurations

You can set up delta, pulse delta, and differential conductance measurements in a front-panel stand-alone configuration or a remote programming configuration. Both systems use RS-232 serial communications between the 6220 or 6221 and the 2182A. The 6220 or 6221 sends setup commands to the 2182A and the 2182A sends delta, pulse delta, or differential conductance readings to the buffer of the 6220 or 6221. Once the test is started, trigger synchronization between the two instruments is controlled by the Trigger Link.

The configurations are shown in the following figures.

Figure 51: Front-panel test system configuration

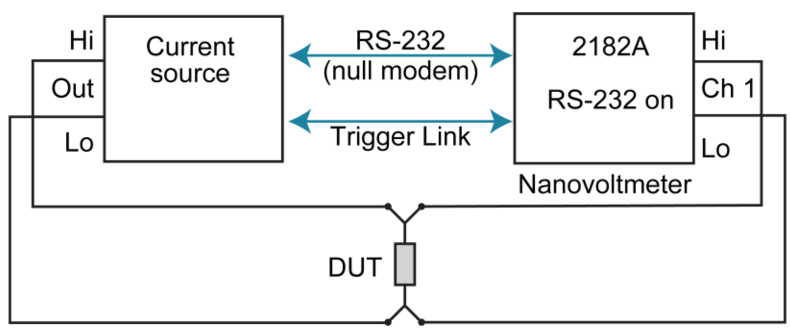
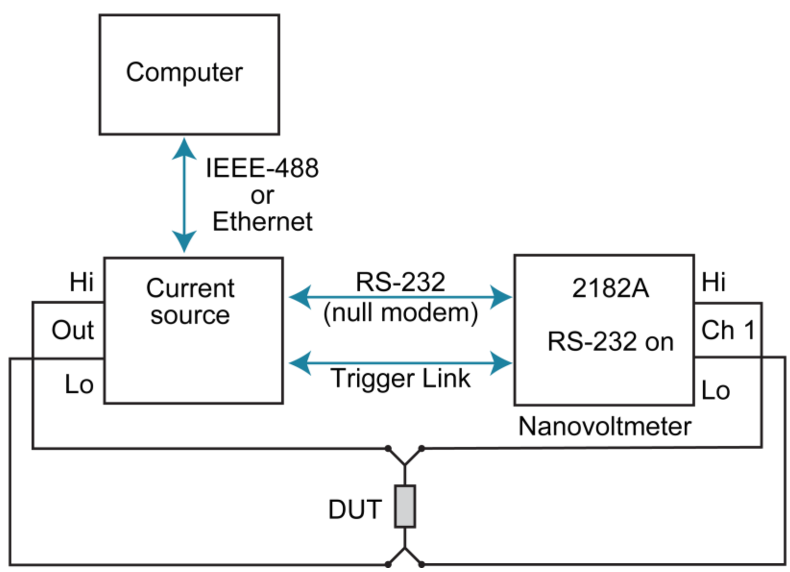


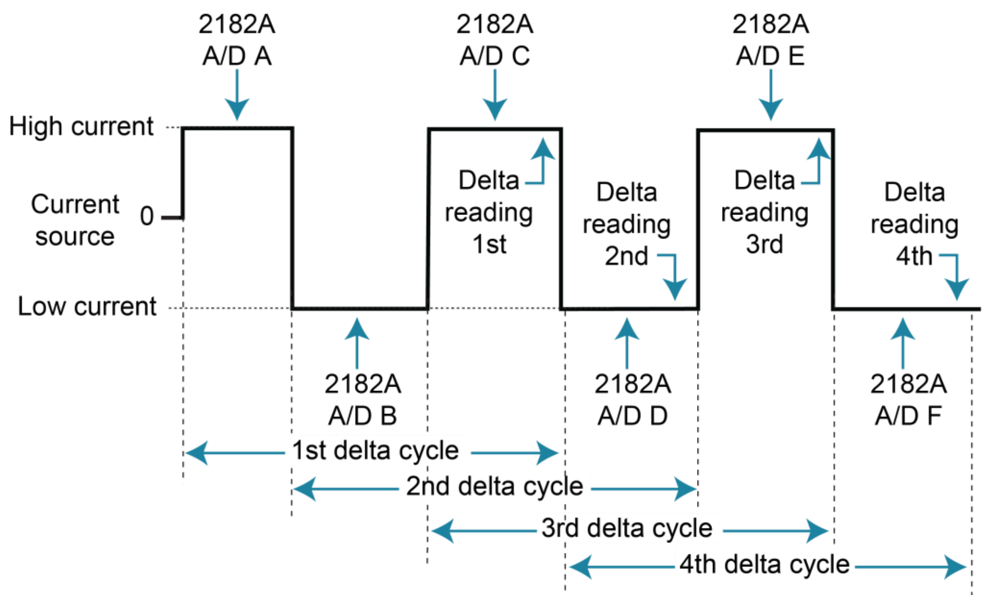
Figure 52: Remote programming test system configuration



Delta measurement process

The delta process is shown in the following figure. As shown, three 2182A A/D conversions are performed to yield a single delta reading. When delta starts, three 2182A A/D readings (A, B, and C) are made and the delta reading is calculated. After the 1st delta cycle, the moving-average technique is used. As shown, a delta reading is yielded for every subsequent 2182A A/D reading. The new A/D reading replaces the oldest A/D reading in the delta calculation.

Figure 53: Delta measurement technique



$$\text{1st delta reading} = \left(\frac{A - 2B + C}{4}\right) \times (-1)^0 \quad \text{3rd delta reading} = \left(\frac{C - 2D + E}{4}\right) \times (-1)^2$$

$$\text{2nd delta reading} = \left(\frac{B - 2C + D}{4}\right) \times (-1)^1 \quad \text{4th delta reading} = \left(\frac{D - 2E + F}{4}\right) \times (-1)^3$$

The following equation can be used to calculate any delta reading:

$$\text{Delta} = \left(\frac{X - 2Y + Z}{4}\right) \times (-1)^n$$

Where:

- X, Y, and Z are the A/D measurements for a delta reading.
- n = Delta cycle number − 1

For example, to calculate the 21st delta reading:

X, Y, and Z are the three A/D measurements for the 21st delta reading.

n = Delta Cycle Number − 1

= 21 − 1

= 20

Therefore:

$$\text{Delta} = \left(\frac{X - 2Y + Z}{4}\right) \times (-1)^{20}$$
$$= \left(\frac{X - 2Y + Z}{4}\right)$$

The $(-1)^n$ term in the delta calculation is used for polarity reversal of every other calculated delta reading. This makes all calculated delta readings in the test the same polarity.

The fundamental measurement for delta is voltage (Volts; V). However, the voltage reading can be converted into a conductance (Siemens; S), resistance (Ohms; Ω), or power (Watts; W) reading by the 6220 or 6221.

Delta calculation example

Assume you want to measure the voltage across a 1 Ω DUT using a constant +10 mA current source and a voltmeter. Ideally, the measured voltage is 10 mV (V = I × R). However, due to a 10 µV thermal EMF in the test leads, the voltmeter actually reads 10.01 mV (0.1% error due to EMF).

The error contributed by EMF can be eliminated by using delta. Assume the square wave output of the 6220 or 6221 is set to 10 mA (high) and −10 mA (low) and the following 2182A measurement conversions (A/Ds) are made for the first delta cycle.

A/D A = 10.01 mV

A/D B = −9.99 mV

A/D C = 10.01 mV

The first delta reading is calculated as follows:

$$\text{Delta} = \left(\frac{A - 2B + C}{4}\right) \times (-1)^{0}$$
$$= \left(\frac{10.01\text{ mV} - 2(-9.99\text{ mV}) + 10.01\text{ mV}}{4}\right) \times (1)$$
$$= \left(\frac{40\text{ mV}}{4}\right) \times (1)$$
$$= 10\text{ mV}$$

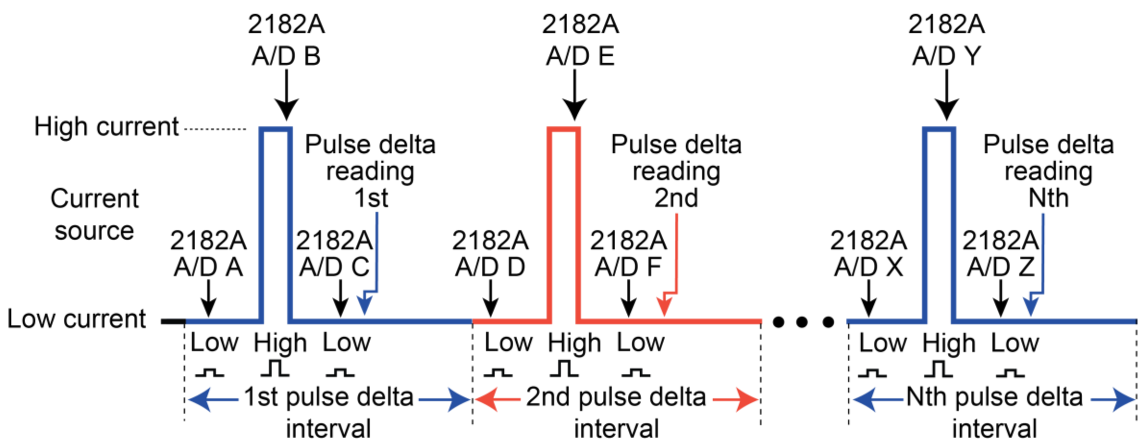
The 10 mV delta reading effectively canceled the 10 µV EMF to provide a more accurate measurement.

Pulse delta measurements

For pulse delta, the 6221 outputs current pulses. Current pulses that have a short pulse width are ideal to test a low-power DUT that is heat sensitive.

By default, pulse delta uses a three-point repeating-average algorithm to calculate readings. Each pulse delta reading is calculated using A/D measurements for a low pulse, a high pulse, and another low pulse. The 6221 outputs the pulses and the 2182A performs the A/D measurements. As shown in the following figure, every three pulses yields a single pulse delta voltage reading.

Figure 54: Pulse delta three-point measurement technique



$$\text{1st pulse delta reading} = \left(\frac{2B - A - C}{2}\right)$$

$$\text{2nd pulse delta reading} = \left(\frac{2E - D - F}{2}\right)$$

$$\text{Nth pulse delta reading} = \left(\frac{2Y - X - Z}{2}\right)$$

Where:
X, Y, and Z are the A/Ds for the first low, high, and second low pulses for the pulse delta cycle.

In cases where the high pulse will cause heating of the DUT, the measurement at the second low pulse could be adversely affected by the heat caused by the high pulse. In that case, the measurement at the second low pulse can be disabled. This does not change the overall timing of the pulse output. Eliminating the second low pulse measurement changes the basic calculation to the following:

Pulse delta = $(2Y - 2X) / 2$

Where:

- Y is the measurement at the high pulse.
- X is the measurement at the first low pulse.

Example: Three-point measurement technique

Three-point measurement technique

Assume you want to measure the voltage across a low power 1 Ω DUT. The pulse delta process reduces DUT heating and eliminates the effects of thermal EMFs.

Assume the 6221 is configured to output +10 mA and 0 mA pulses. Due to a 10 µV thermal EMF in the test leads, the following 2182A measurement conversions (A/Ds) are made for the first pulse delta cycle.

A/D A = 0.01 mV

A/D B = 10.01 mV

A/D C = 0.01 mV

The first pulse delta reading (using the three-point measurement technique) is calculated as follows:

$$PulseDelta = \left(\frac{2B - A - C}{2}\right) \times (-1)^0$$
$$= \frac{2(10.01) - (0.01) - (0.01)}{2}$$
$$= \frac{20\text{ mV}}{2}$$
$$= 10\text{ mV}$$

The above three-point measurement technique effectively eliminated the 10 µV thermal EMF from the pulse delta reading.

Example: Two-point measurement technique

Assume for the above example that DUT heating causes the A/D measurement at point C to be 1.01 mV. Using the three-point measurement technique, pulse delta (by calculation) would instead be 9.5 mV. This results in 5% measurement error due to heating.

To eliminate the effects of heating, do not make the measurement at point C (the low pulse). For this two-point measurement technique, pulse delta is calculated as follows:

$$PulseDelta = \left(\frac{2B - 2A}{2}\right) \times (-1)^0$$
$$= \frac{2(10.01) - 2(0.01)}{2}$$
$$= \frac{20\text{ mV}}{2}$$
$$= 10\text{ mV}$$

Measurement units

The fundamental pulse delta measurement is in volts. The reading can be converted into an Ohms (W), Siemens (S), or power (W) reading by the Model 6221.

With power units selected, a pulse delta reading can be expressed (and displayed) as a peak power reading or an average power reading:

W (peak power) = I × V

W (average power) = I × V × Duty Cycle

Pulse delta outputs

Pulse delta output is made up of one or more pulse delta cycles. Each cycle is made up of three output pulses (low, high, and low). The period for a cycle is adjustable and is the same for all cycles. The output pulses have an adjustable pulse width, which is the same for all pulses.

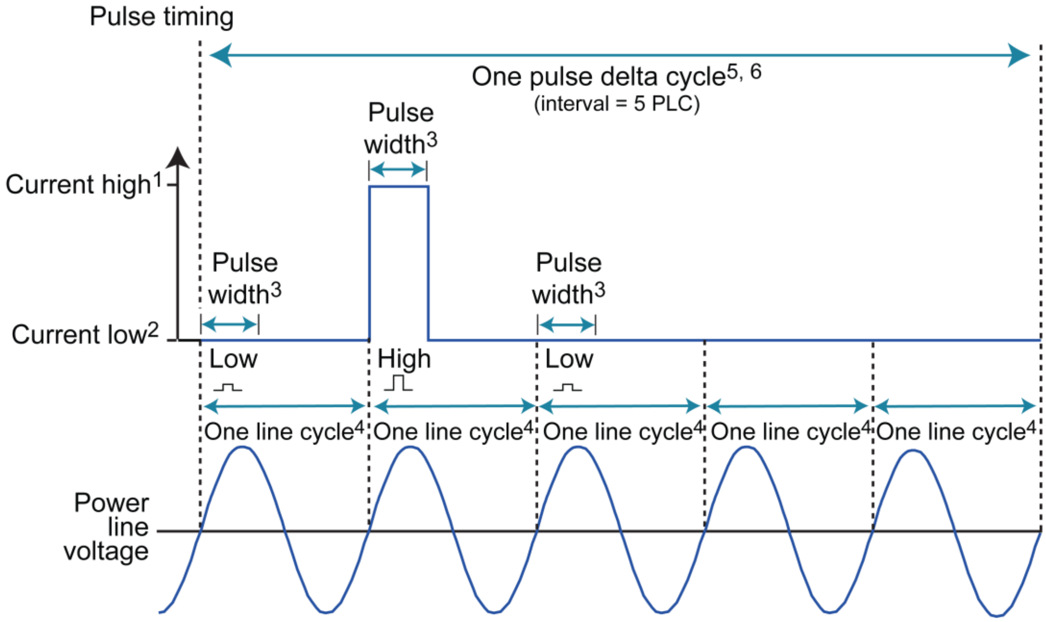
The basic pulse delta output types are fixed output and sweep output. For fixed output, all high and low pulses are fixed for all pulse delta cycles in the test. For sweep output, the sweep (SWP) function of the 6221 outputs a staircased, logarithmic or user-specified (custom) pulse sweep.

Fixed output

The following figure shows one pulse delta cycle for a fixed output. As shown, the Model 6221 outputs a low pulse, a high pulse, and then another low pulse during every pulse delta cycle. The pulse width is adjustable and is the same for all high and low pulses. The cycle interval is also adjustable and is based on the set number of power line cycles. The pulse delta interval shown in the figure is set for 5 power line cycles (PLCs), which is the default setting. After the set interval expires, the next pulse delta cycle starts if pulse count is >1.

Pulses are synchronized to the frequency of the power line voltage. When pulse delta is started, the three pulses (low, high, and low) are generated on the positive-going edges of the first three power line cycles. For the remaining power line cycles in the interval, the output remains at the current low level.

Figure 55: Pulse timing



Note	Description
1	Current high can be set from −105 mA to +105 mA (default is 1 mA)
2	Current low can be set from −105 mA to +105 mA (default is 0 mA)
3	Pulse width can be set from 50 µs to 12 ms (default is 110 µs)
4	One 60 Hz power line cycle = 16.667 ms (1/60) One 50 Hz power line cycle = 20 ms (1/50)
5	With interval set to 5 PLCs: ■ 60 Hz: One pulse delta cycle = 83.33 ms (5/60) ■ 50 Hz: One pulse delta cycle = 100 ms (5/50)
6	Interval can be set from 5 to 999999 PLC (default is 5 PLC)

Sweep output

The sweep feature of the Model 6221 can be used to output a series of pulses that allow the use of different levels for the high pulses. Each high pulse returns to the programmed low pulse level. The low level is the same for all pulses.

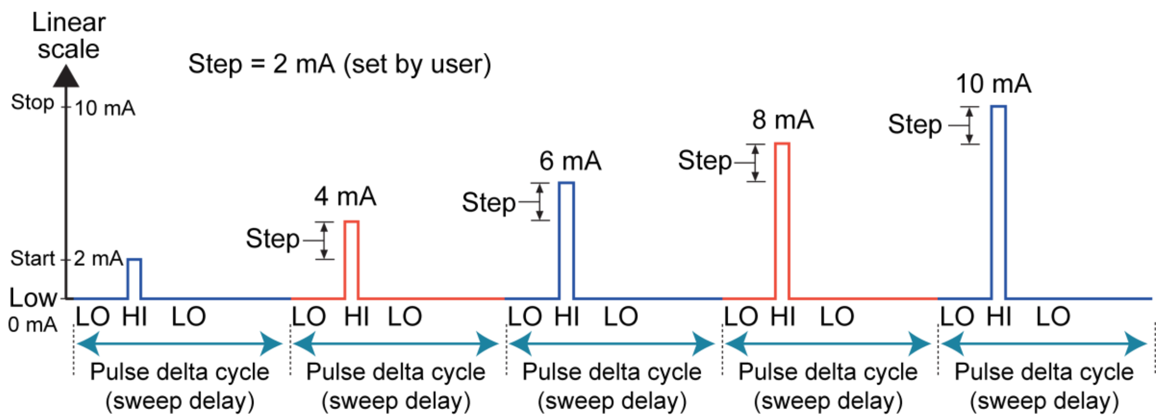
Like the fixed output shown in Fixed output (on page 6-19), a sweep output is synchronized to the frequency of the power line voltage. The pulse width is adjustable and is the same for all pulses.

The available sweeps include staircase sweep, logarithmic sweep, and custom sweep. Examples of these sweep outputs are described in the following topics.

Staircase sweep

The following figure shows an example of a staircase sweep output. The sweep is configured to start high pulses at 2 mA and staircase to 10 mA in 2 mA steps. The low pulse level for this sweep is 0 mA.

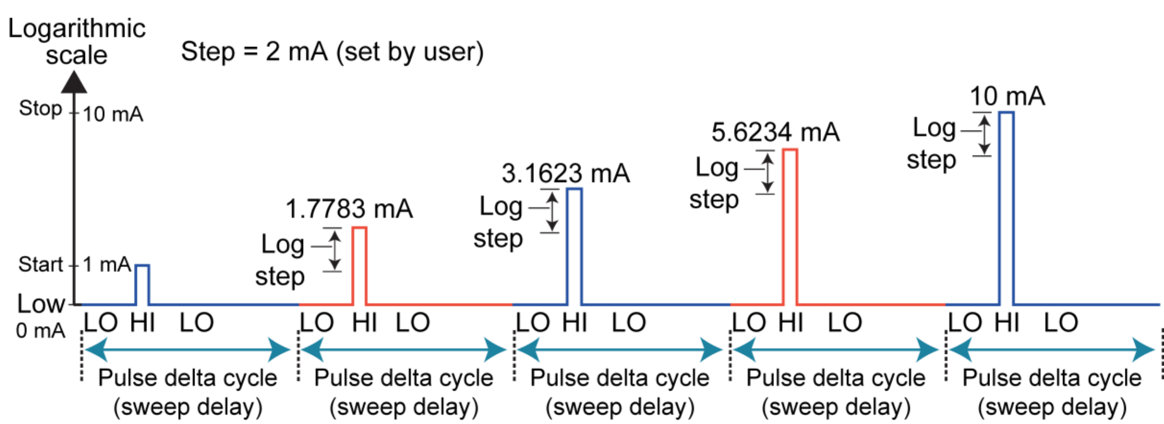
Figure 56: Fixed pulse delta output



Logarithmic sweep

The following figure shows an example of a logarithmic sweep output. The sweep is configured to output five high pulses (points). The first high pulse starts at 1 mA and logarithmically steps to 10 mA. The low pulse level for this sweep is 0 mA.

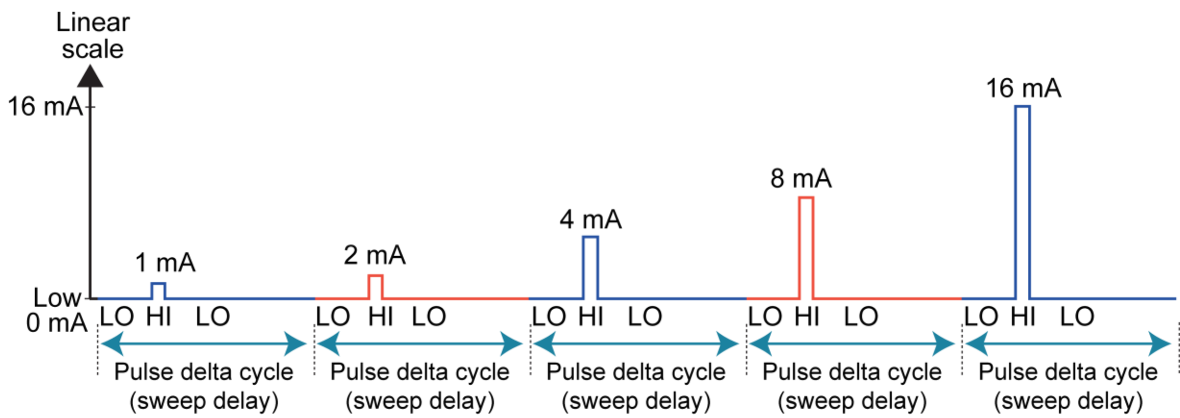
Figure 57: Logarithmic sweep



Custom sweep pulse

The following figure shows an example of a custom sweep output. The sweep is configured to output five high pulses (points). The level for each high pulse is specified by the user. The high pulse levels for this output are 1 mA, 2 mA, 4 mA, 8 mA and 16 mA. The low pulse level for this sweep is 0 mA.

Figure 58: Custom sweep pulse train

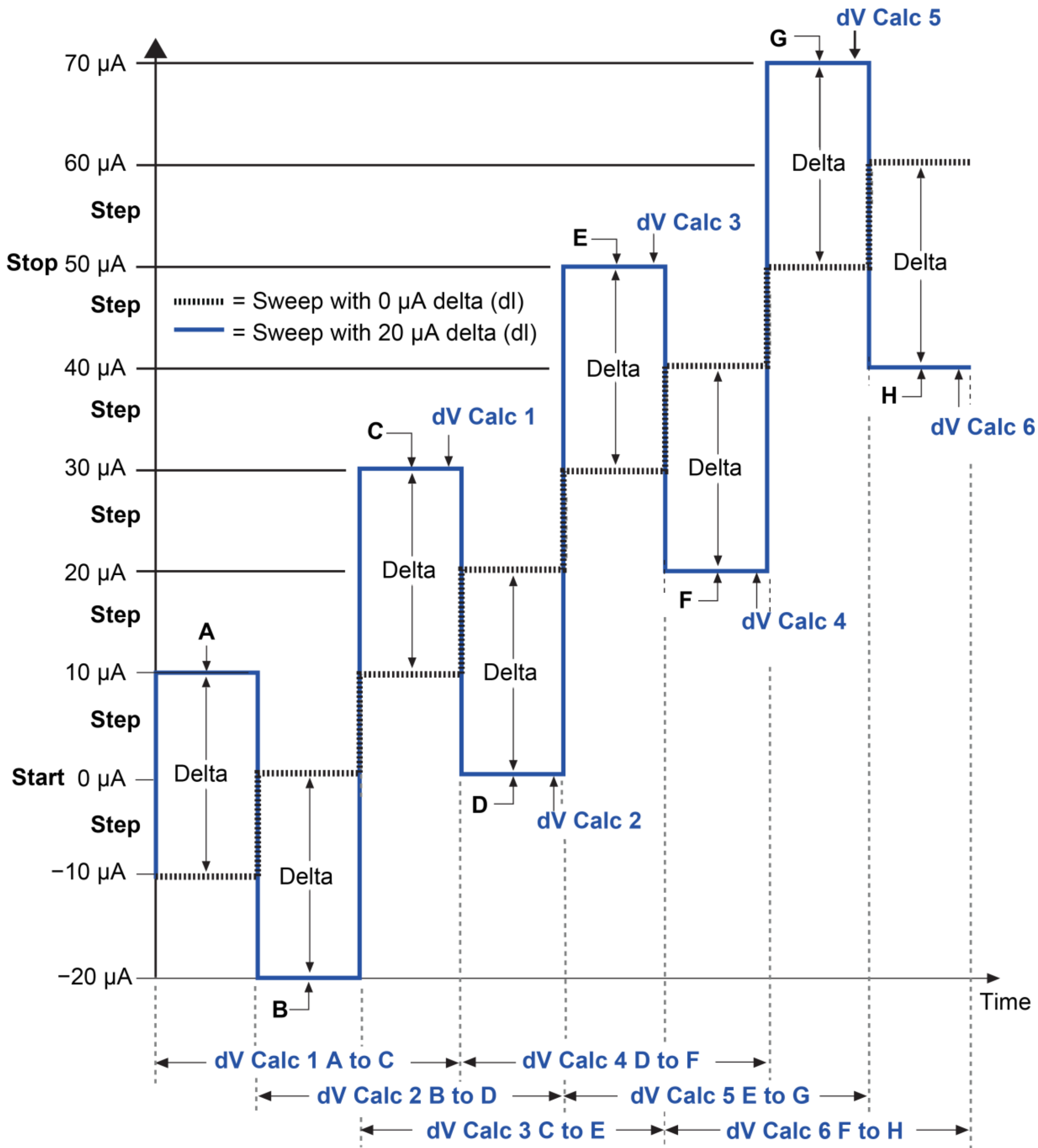


Differential conductance measurement process

Differential measurements can be used to study the individual slopes of an I-V curve. By applying a known differential current (dI) to a device, differential voltage (dV) measurements can be made. With dI and dV known, differential conductance (dG) (and differential resistance dR) can be calculated.

This differential measurement process is shown in the following figure. The Model 622x is configured to output a stepped sweep with a specified delta, which is the differential current (dI). As shown in the figure, delta is added to and subtracted from each subsequent step in the sweep. The solid line is the actual output of the Model 622x.

Figure 59: Differential Conductance measurement process



The settings are:

- dV Calc = The differential voltage (dV) calculation using the last three A/D readings
- Start = 0 µA
- Step = 10 µA
- Stop = 50 µA
- Delta (dI) = 20 µA
- B, C, D, E, F, G, and H are the 2182A voltage measurement conversions (A/D readings)

As shown, each differential voltage calculation (dV Calc) uses the three previous 2182A A/D measurement conversions. Delta (dI) is the same for all calculated points. With dI known and dV calculated, the Model 622x can also calculate, display, and store the differential conductance (dG) or differential resistance (dR) for each calculated point.

The following equations are used by the 622x to calculate differential voltage (dV). To calculate dV, points A through H are 2182A voltage measurements (A/D readings).

$$dV\,1 = \frac{[(A-B)/2] + [(C-B)/2]}{2} \times (-1)^0 \qquad dV\,4 = \frac{[(D-E)/2] + [(F-E)/2]}{2} \times (-1)^3$$

$$dV\,2 = \frac{[(B-C)/2] + [(D-C)/2]}{2} \times (-1)^1 \qquad dV\,5 = \frac{[(E-F)/2] + [(G-F)/2]}{2} \times (-1)^4$$

$$dV\,3 = \frac{[(C-D)/2] + [(E-D)/2]}{2} \times (-1)^2 \qquad dV\,6 = \frac{[(F-G)/2] + [(H-G)/2]}{2} \times (-1)^5$$

dG and dR calculations

With dI known (dI = delta) and dV calculated, the 622x can calculate differential conductance (dG) or differential resistance (dR).

With G units selected, readings are calculated as follows: dG = dI/dV.

With R units selected, readings are calculated as follows: dR = dV/dI.

dV calculations

The dV calculations for the first six dV readings are shown in Differential Conductance measurement process (on page 6-22). To calculate any dV reading in the test, you can use the following formula.

$$dV = \frac{\frac{(X-Y)}{2} + \frac{(Z-Y)}{2}}{2} \times (-1)^n$$

Where:

- X, Y, and Z are the three A/D measurements for a dV reading
- n = Reading Number − 1

The $(-1)^n$ term in the dV calculation is used for polarity reversal of every other calculated dV reading. This makes all calculated dV readings in the test the same polarity.

Measurement units

The fundamental measurement for differential conductance is differential voltage (dV). However, the dV reading can be converted into a differential conductance (dG), differential resistance (dR), or power (Watts) reading by the Model 622x.

With Ohms (dR) or Siemens (dG) measurement units selected, the reading is calculated as follows:

dR = dV/dI

dG = dI/dV

With power measurement units selected, power is calculated using average voltage and average current and is explained in the following paragraphs.

Average voltage calculation

Average voltage is the average bias voltage that was present across the device when the corresponding differential conductance reading was made. For remote operation, the average voltage reading for differential conductance can be included in the returned data string.

Average voltage is calculated as follows:

$$\text{Average voltage} = \frac{(X + 2Y + Z)}{4}$$

Where:

- Average voltage is the average voltage corresponding to a given differential voltage (dV) reading
- X, Y, and Z are the three A/D measurements for the dV reading

Power calculation

With watts (power) measurement units selected, power for differential conductance is calculated using the average voltage and average current. Average current is calculated by the Model 622x as follows:

$$\text{Average current} = \frac{(X + 2Y + Z)}{4}$$

Where:

- Average current corresponds to a given differential conductance reading
- *X*, *Y*, and *Z* are the current levels for the differential conductance reading

With average voltage and average current known (calculated by the Model 622x), power is then calculated as follows:

Power = Average voltage × Average current

Section 7

Stepping and scanning

In this section:

Step and scan overview

You can use the 2182A to step or scan its two input channels. You can use Channel 1 to scan external cards installed in a switching mainframe such as the Model 707B.

When you internally step or scan the 2182A input channels, Channel 2 is measured first and then Channel 1 is measured. For every step and scan cycle, Channel 2 is measured once and Channel 1 can be measured from 1 to 1023 times.

When you use external switching cards, the switching mainframe controls the opening and closing of individual channels. Channel 1 of the 2182A can step, scan, and measure up to 800 external channels.

For both internal and external stepping and scanning, measured readings are automatically stored in the buffer.

SCPI programming - stepping and scanning

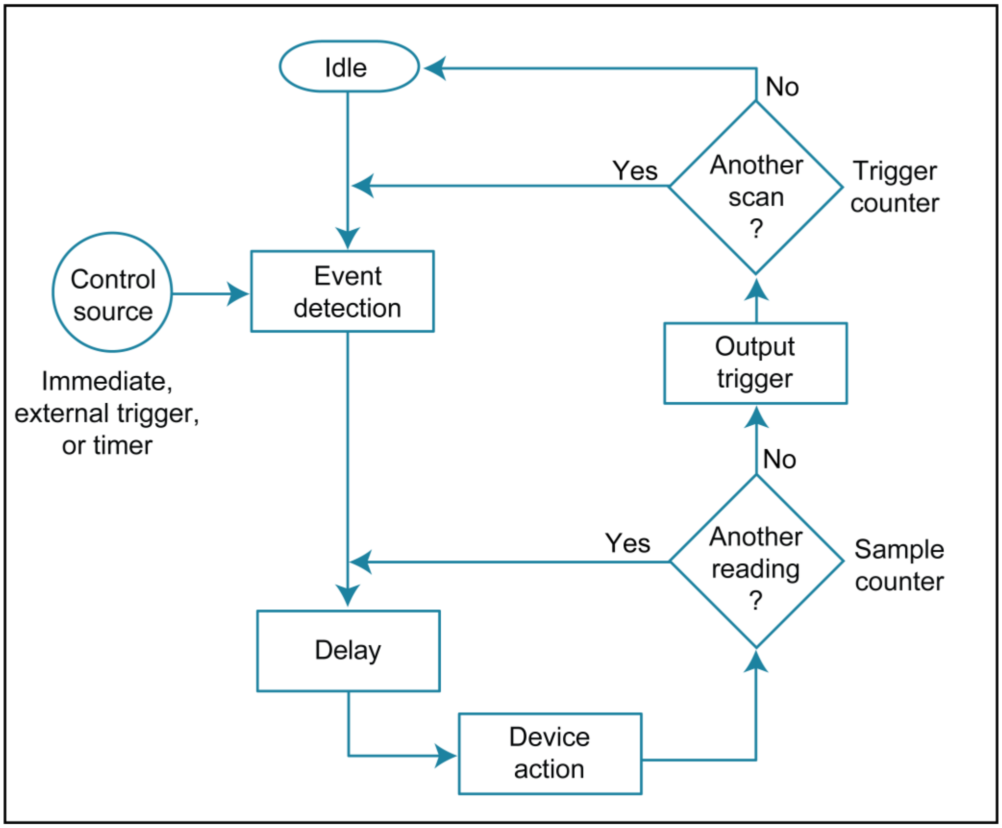
The stepping and scanning command descriptions and example are provided in the ROUTe subsystem (on page 13-39).

Trigging commands are also used for scanning. Refer to the TRIGger subsystem (on page 13-94) for information on triggering commands. Also see Triggering (on page 9-1) for general information on triggering.

Internal stepping and scanning

The following figure shows the front-panel trigger model for internal scanning.

Figure 60: Front-panel trigger model when using internal scanning



The control source can be immediate, external trigger, or a timer. With immediate triggering, event detection occurs immediately and operation proceeds to the delay block.

If an external trigger is used, after the internal scan is configured, press the EX TRIG key to place the instrument in the external trigger mode. When the SCAN key is pressed, the internal scan is enabled. However, it does not start until an external trigger is received or the TRIG key is pressed. After the trigger occurs, operation proceeds to the delay block.

If a timer is used, the timer controls the time interval between internal scans. It has no affect on the time interval between each measurement in a scan cycle. When SCAN is pressed, the timer starts, event detection occurs immediately, and operation proceeds to the delay block. If the instrument is configured for an additional scan, operation loops back to this control source, where it waits until the timer interval expires. If the timer interval expired, event detection occurs immediately.

The delay block waits for the automatic or manual delay period to expire. If no delay is set, action moves to the device action block.

Measurements are made at the device action block. The first measurement is made on Channel 2 of the 2182A. Subsequent measurements are made on Channel 1.

For each scan cycle, Channel 2 is measured once and Channel 1 is measured a specified number of times. The sample count is the sum of those two values (CH1 Count + 1). Operation continues to loop back to the delay and device action blocks until all the sample readings are made.

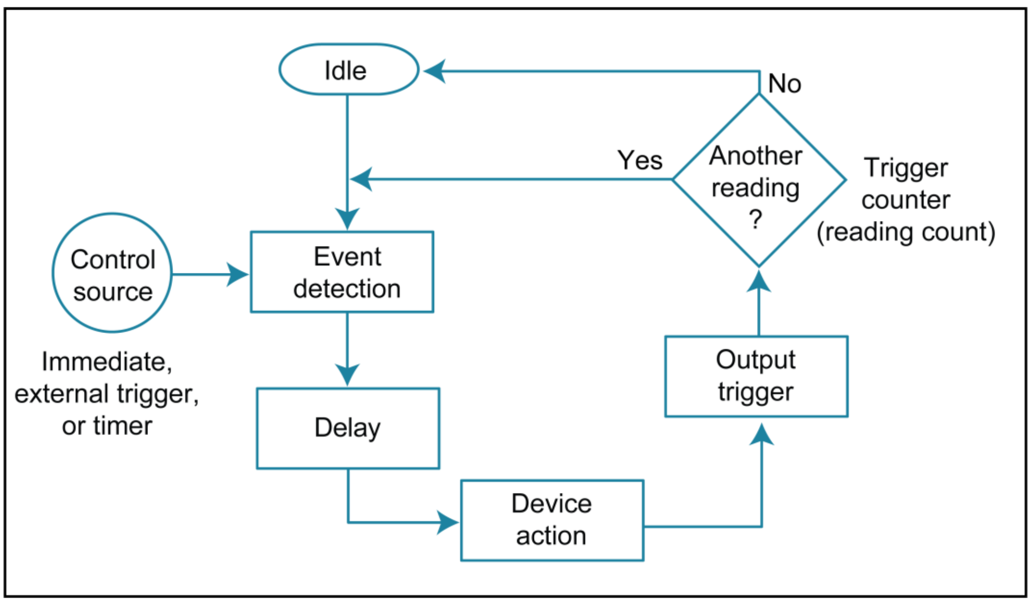
After the last Channel 1 measurement is made, an output trigger is applied to the rear-panel Trigger Link connector.

The trigger counter determines how many scan cycles are performed. If the instrument is configured for another scan, operation loops back to the control source for another pass through the trigger model.

Other stepping and scanning operations

The following figure shows the front-panel trigger model for other stepping and scanning operations.

Figure 61: Front-panel trigger model when using other stepping and scanning



The control source can be immediate, external trigger, or a timer. With immediate triggering, event detection occurs immediately and operation proceeds to the delay block.

If an external trigger is used, after the stepping or scanning operation is configured, press the EX TRIG key to place the instrument in the external trigger mode. When the STEP or SCAN key is pressed, the step or scan is enabled. However, it does not start until an external trigger is received or the TRIG key is pressed. After the trigger occurs, operation proceeds to the delay block.

When a timer is set, it sets a time interval between channels in a step and scan cycle. When STEP or SCAN is pressed, the timer starts, event detection occurs immediately, and operation proceeds to the delay block. When operation returns the control source, it waits until the timer interval expires. If the timer interval is already expired, event detection occurs immediately.

The delay block waits for the automatic or manual delay period to expire. If no delay is set, action moves to the device action block.

Measurements are made at the device action block. For internal stepping, the first measurement is performed on Channel 2 of the 2182A. Subsequent measurements are made on Channel 1. For internal scanning and external scanning, each measurement corresponds to a channel in the step and scan list.

After each measurement is made, an output trigger is applied to the rear-panel Trigger Link connector.

The trigger count specifies the number of measurements to make. If another measurement is to be made, operation returns to the event detection.

Stepping and scanning controls

The front-panel control keys for stepping and scanning are:

SHIFT then **CONFIG:** Selects and configures internal or external scanning.

EX TRIG: Selects the External Trigger control source.

TRIG: Satisfies event detection for the External Trigger control source.

STEP: Enables a stepping operation of consecutive channels.

SCAN: Enables a scanning operation of consecutive channels.

SHIFT then **HALT:** Stops stepping or scanning and restores the trigger model to idle.

Internal stepping and scanning settings

The settings for internal stepping and scanning are:

- **Timer:** The maximum timer interval is 99H:99M:99.999S (hour:minute:second format).
- **Channel 1 Count:** The number of measurements to be made on Channel 1 (1 to 1023). For each step or scan cycle, only one measurement is made on Channel 2.
- **Reading Count:** The total number of measurements to be made for each step or scan cycle (2 to 1024). For example, if the count for Channel 1 is set to 3, the reading count is initially set to 4 (the extra reading is for Channel 2). For each additional step or scan cycle, add 4 to the reading count. Therefore, to perform three step or scan cycles, set the reading count to 12.

NOTE

If you change the reading count, it must be a multiple of the initial count value. For example, if the initial reading count is 3, you can change it to a value such as 6, 9, or 12. If you enter a non-multiple value, the instrument selects the next lower value that is a multiple.

NOTE

You can set the reading count to a value that is more than 1024 or `INFinite`, but only the first 1024 readings are stored in the buffer.

To configure internal stepping or scanning:

1. Press **SHIFT** and then **CONFIG**. Use the ► key to display the present SCANNING type (INTernal or EXTernal).
2. Press the ▲ and ▼ key to display **INT**. Press **ENTER**.
3. The present state of the timer is displayed (OFF or ON). To change it, press ▲ and ▼ to display the correct timer state and press **ENTER**.
4. If you turned the timer on, the timer interval is displayed. Use the edit keys (◄, ►, ▲, and ▼) to set the timer interval. Press **ENTER**.
5. The present Channel 1 Count (CH1 CNT) is displayed. Use the edit keys to set the number of measurements for Channel 1 and press **ENTER**.
6. The present reading count (RNG CNT) is displayed. To increase the reading count, use the edit keys to display the value and press **ENTER** to select the value and return to the normal display state.

Internal scanning example

In this example, the channel 1 count is set to 4, so each scan cycle measures channel 2 once and channel 1 four times, for a total of five measurements. This sets the sample counter in the trigger model to 5. The reading count is set to 10, so the five measurement scans are done twice. This sets the trigger counter to 2. A total of 10 measurements are made and stored in the buffer.

Settings:

- **Control Source:** Immediate (timer off)
- **Delay:** Auto
- **Channel 1 Count:** 4
- **Reading Count:** 10

Press the **SCAN** key. Operation proceeds to the device action block, where a measurement on channel 2 is made. The sample counter is decremented to 4, causing operation to loop back to the device action block for a measurement on Channel 1. Operation loops back to the device action block three more times to complete the scan cycle.

After the scan cycle, the trigger counter is decremented to 1 and an output trigger is sent. Operation loops back to the control source, where it immediately repeats the five measurement scan. An output trigger is again sent and the instrument goes into the idle state.

Internal stepping example

In this example, a reading count of ten sets the trigger counter in the trigger model shown in Other stepping and scanning operations (on page 7-3) to 10. A total of ten measurements are made and stored in the buffer. The 1st and 6th measurements are made on Channel 2. The remaining measurements are made on Channel 1.

Settings:

- **Control Source:** Immediate (timer off)
- **Delay:** Auto
- **Channel 1 Count:** 4
- **Reading Count:** 10

Press the **STEP** key. Operation proceeds to the device action block, where a measurement on channel 2 is made. An output trigger is sent and the trigger counter is decremented to 9. Operation then returns to the device action block four more times to make four measurements on Channel 1. An output trigger is sent after every measurement.

At this point, the trigger counter is at 5. Operation continues to loop until channel 2 is again measured one time, and Channel 1 is again measured four times. After the last measurement is made, the instrument goes into idle.

External stepping and scanning (channel 1)

When using external switching cards, the switching mainframe controls the opening and closing of individual channels. You can step, scan, and measure up to 800 external channels using Channel 1 of the 2182A. Channel 2 cannot be used for external stepping or scanning. Readings are automatically stored in the buffer.

To synchronize 2182A measurements with external channel closures, connect the Trigger Link lines of the nanovoltmeter and switching mainframe. Refer to Triggering (on page 9-1) for details and an example that uses external triggering.

NOTE

Channel annunciators do not turn on during an external step or scan.

External stepping and scanning settings

The settings for external stepping and scanning are:

- **Min and Max Values:** The minimum and maximum values specify the beginning and ending channels for the step or scan list. Valid values for Min are 1 to 799. Valid values for Max are 2 to 800. The Max value must be larger than the Min value.
- **Timer:** The maximum timer interval is 99H:99M:99.999S (hour:minute:second format).
- **Reading Count:** The total number of measurements that are made for each step/scan cycle. For example, if Min is set to 1 and Max is set to 10, the reading count is automatically set to 10. For each additional step or scan cycle, add 10 to the reading count. Therefore, to perform three step or scan cycles, set the reading count to 30. Reading count can be set from 2 to 1024.

To configure external stepping or scanning:

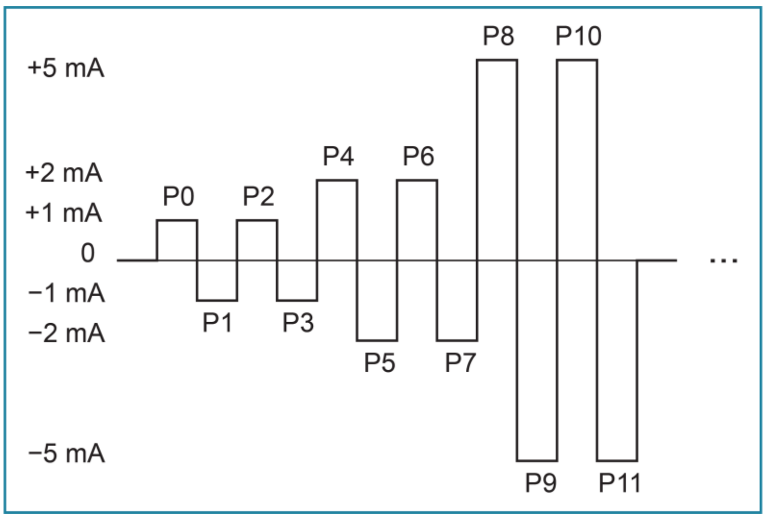
1. Press **SHIFT** and then **CONFIG**. Use the ▼ key to display the present SCANNING type (INTernal or EXTernal).
2. Press the ▲ and ▼ keys to display **EXT** and press **ENTER**.
3. The minimum channel (MIN CHAN) to step or scan is displayed. Use the edit keys (▲, ▼, ◄, and ▼) to specify the Min channel and press **ENTER**.
4. The maximum channel (MAX CHAN) is displayed. Use the edit keys to specify the Max channel and press **ENTER**.
5. The present state of the timer is displayed (OFF or ON). Press ▲ and ▼ to display the timer state, and press **ENTER**.
6. If you turned the timer on, the timer interval is displayed. Use the edit keys (▲, ▼, ◄, and ▼) to set the timer interval. Press **ENTER**.
7. The present reading count (RNG CNT) is displayed. It is ((Max − Min) + 1). To increase the reading count, use the edit keys to display the value. Press **ENTER** to return to the normal display state.

Constant magnetic field or temperature example

You can use SCAN to measure voltage while sweeping the current through a sample with a constant magnetic field or a constant temperature.

This example uses a Keithley Model 2400 SourceMeter instrument and a Model 8501 Trigger Link cable. The 2400 is programmed to sweep the current using the bipolar dc current-reversal technique, increasing the amplitude, as shown in the following figure. Channel 1 measures the voltage across the sample, while Channel 2 measures the voltage across a known reference resistor (R_{REF}) to determine the current in the sample.

Figure 62: Waveform to be programmed into Model 2400



This sweep is stored in the 2182A memory and can be recalled at the end of the sweep. Because the 2400 and 2182A are in tight hardware control, the dc current-reversal technique can be run at a rate of 8 readings per second at a 1 PLC integration time. This greatly reduces any thermal EMFs in the system by reversing the dc current before temperature effects can occur. It allows collector currents (I_C) to be located at a faster reading rate.

NOTE

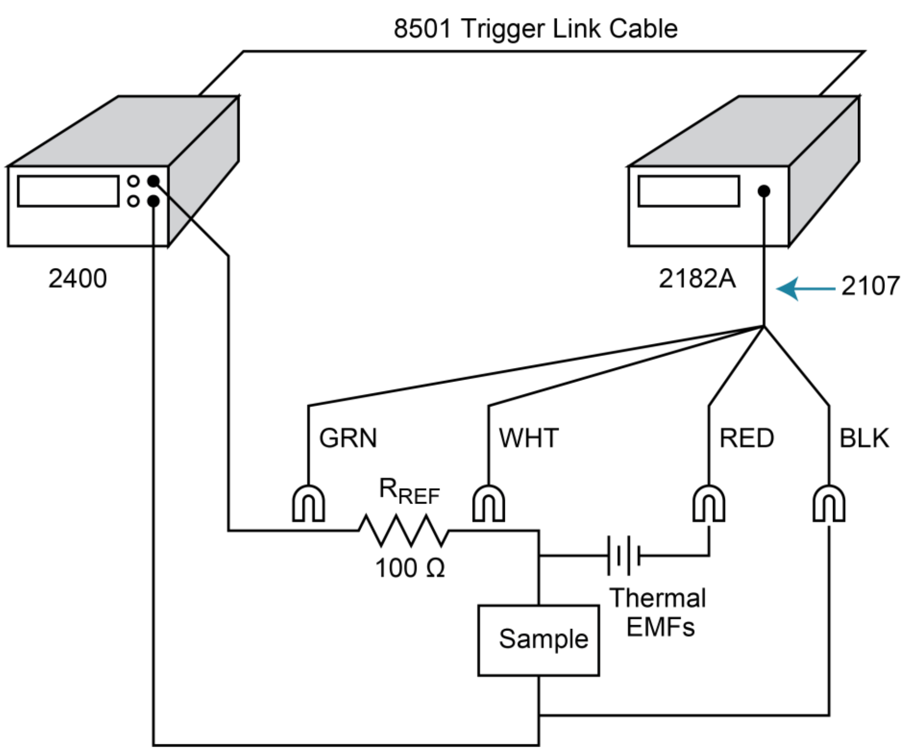
For information on triggering, refer to Triggering (on page 9-1).

Set up the 2182A

For this example, set up the 2182A as follows:

- Restore factory defaults
- Filters: Off
- Rate: 1 PLC
- Ch1: 10 mV
- Ch2: 1 V
- External Trigger: on
- Delay: Set to time needed for cable settling
- Config SCAN: INT with the timer off
- Config SCAN channel 1: Count 3 (channel 1 stores three readings for each 2400 programmed current level)
- Config SCAN channel 2: Channel 2 stores one reading for each programmed current level
- Reading count: 48

Figure 63: Setup of the 2182A and 2400



Set up the 2400

Menu: Savesetup: Global: Reset: Bench

Meas: V

Source: I

Configure the trigger:

- ARM-LAYER:
  - ARM-IN: IMMEDIATE
  - ARM-OUT: LINE: #3
  - ARM-OUT: EVENTS: TRIG-LAYER-DONE= OFF
- TRIG-LAYER: TRIGGER-IN: TRIGGER-LINK: #1: EVENT DETECT BYPASS NEVER: TRIGGER IN EVENTS: SOURCE= ON all others off TRIG-LAYER: TRIGGER-OUT: LINE: #2
  - EVENTS :TRIGGER OUT EVENTS: SOURCE = ON, all others off.
  - COUNT 12

Config Sweep:

- TYPE: CUSTOM
  - #-POINTS: 12
  - ADJUST-POINTS: see waveform
  - COUNT: INFINITE

Speed: 0.01 plc

Turn output: On

SWEEP :exit

TRIG: HALT

Run the scan from the front panel

To run the scan:

1. On the 2182A, enable **SCAN**. The memory buffer annunciator is activated.
2. On the 2400, enable **SWEEP**. The Arm annunciator is activated.
3. On the 2400, press **Trig**.
4. After completion of the sweep, recall the data from the 2182A using the `TRACe` command.

5. To remove thermal EMFs from the readings, do the following math on the recalled data:
   - CH2 Rdg #1 = (Buffer Rdg #1 - Buffer Rdg #5) / 2
   - CH2 Rdg #2 = (Buffer Rdg #9 - Buffer Rdg #13) / 2
6. Repeat.
   - CH1 Rdg_Pos #1 = (Buffer Rdg #2 + Buffer Rdg #3 + Buffer Rdg #4) / 3
   - CH1 Rdg_Neg #1 = (Buffer Rdg #6 + Buffer Rdg #7 + Buffer Rdg #8) / 3
   - CH1 Rdg#1 = (CH1 Rdg_Pos #1 - CH1 Rdg_Neg #1) / 2
   - CH1 Rdg_Pos #2 = (Buffer Rdg #10 + Buffer Rdg #11 + Buffer Rdg #12) / 3
   - CH1 Rdg_Neg #2 = (Buffer Rdg #14 + Buffer Rdg #15 + Buffer Rdg #16) / 3
   - CH1 Rdg#2 = (CH1 Rdg_Pos #2 - CH1 Rdg_Neg #2) / 2
7. Repeat.

Run the scan using code

The following code extracts the data programmatically.

```
' This is for Channel 2 Data.
Let NumRdgsPerStep = 4   ; 1 Ch2 and 3 Ch1 readings stored in the buffer
     / 2400 current level.
Let CalcRdgs = 6   ; Total number of positive or negative current
       levels out of the 2400.
Let k = 1
For j =1 to CalcRdgs
   CH2 Rdg#( j ) = Buffer Rdg#( k )- Buffer Rdg#( k + NumRdgsPerStep )
   CH2 Rdg#( j ) = CH2 Rdg#( k ) / 2
   k = k + ( NumRdgsPerStep * 2 )
Next j

' This for For Channel 1 Data.'
Let NumRdgsPerStep = 4
Let CalcRdgs = 6
Let k = 1
For j = 1 to CalcRdgs
   Let CH1 Rdg#_pos = 0
   Let CH1 Rdg#_neg = 0
For m =1 to NumRdgsPerStep - 1
   CH1 Rdg#_pos = CH1 Rdg#_pos + Buffer Rdg#( k +m )
   CH1 Rdg#_neg = CH1 Rdg#_neg + Buffer Rdg#( k +m + NumRdgsPerStep )
Next m
   CH1 Rdg#( j ) = ( CH1 Rdg#_pos - CH1 Rdg#_neg ) /
      ( (NumRdgsPerStep - 1) * 2)
Next j
```

Code to calculate the dc current reversal data

The following QBASIC code sets up an array for all the data from the 2182A buffer, parses the comma-separated data into the array, and calculates the dc current reversal data for Channel 1 and Channel 2.

```
CONST Addr = 7      'The address of the 2182A.
CONST NumRdgsPerStep = 4   'The total number of CH1 & CH2
                           'readings at each positive or negative
                           'current step.
CONST CalcReadings = 6     'Represents number of paired positive
                           'and negative steps in 2400.
CONST NumRdgs = (NumRdgsPerStep * 2 * CalcReadings)
       'NumRdgs represents the number of readings to
       'store in 2182A buffer.
   'The 2 accounts for positive and negative steps.

DIM DataCH2$(CalcReadings)  'Array represents total number of
                            'channel 2 readings.
DIM DataCH1$(CalcReadings)  'Array represents total number of
                            'channel 1 readings.
FOR i = 1 TO (CalcReadings)   'Allocates space for each channel reading.
   'DataCH2$(i) = SPACE$(18)
   'DataCH1$(i) = SPACE$(18)
NEXT i

'Code for parsing single string of buffer response into 48 individual readings.
OneReading$ = SPACE$(20)   'Represents one reading from
                           'buffer string response.
OneCharacter$ = SPACE$(2)  'Represents one character from
                           'buffer string response.
AsciiRdgsBuf$ = SPACE$(18 * NumRdgs)
'Represents the string of buffer response.
DIM Readings!(1 TO NumRdgs)   'array of the 48 individual readings in
'numerical representation form - converted from ASCII.

CALL send(Addr, "TRACE:DATA?", status%)
'Ask 2182A for the buffer response.
CALL enter(AsciiRdgsBuf$, length%, Addr, status%)
'Read in buffer response.
'Start parsing the data readings.
ParseLength = 1   'Represents how many characters to extract from
                  'response string.
CurrentPosition% = 1   'Represents which character on in response
                       'string.
OneReading$ = ""       'Clear out string contents.
ReadingOn% = 1         'Represents the individual reading on.
DO
   OneCharacter$ = MID$(AsciiRdgsBuf$, CurrentPosition%, ParseLength)
      'Above line reads in the next character for the
      'buffer response.
   IF (OneCharacter$ = ",") THEN
      'Found an individual reading so store it as such.
```

```
      Readings!(ReadingOn%) = VAL(OneReading$)
      OneReading$ = ""   'Clear out so able to read next
                                'individual reading.
         ReadingOn% = ReadingOn% + 1
    'Increment counter for next individual reading.
   ELSE
'Still building an individual reading so add on
'the next character.
      OneReading$ = OneReading$ + OneCharacter$
   END IF
   CurrentPosition% = CurrentPosition% + 1
'Increment character on in the buffer response.
LOOP UNTIL (CurrentPosition% > length%)
'Loop until pass the number of characters read
'in with the buffer response.
Readings!(ReadingOn%) = VAL(OneReading$)
'Store last individual reading since it will not
'be separated by a comma.

'Calculate DataIC and DataV values
' Where Chan2! is the CH2 numerical representation for string CH2 data
' and Chan1! is the CH1 numerical representation for string CH1 data.
' CH1pos! is the positive portion for channel 1.
' CH1neg! is the negative portion for channel 1.
k% = 1   'represents the reading in Reading! to use in
         'calculation.
FOR j = 1 TO (CalcReadings)
   Chan2! = Readings!(k%) - Reading!(k% + NumRdgsPerStep)
   Chan2! = Chan2! / 2
   DataCH2$(j) = STR$(Chan2!)
   CH1pos! = 0!
      CH1neg! = 0!
   FOR i = 1 TO (NumRdgsPerStep - 1)
      CH1pos! = CH1pos! + Reading!(k% + i)
      CH1neg! = CH1neg! + Reading!(k% + i + NumRdgsPerStep)
   NEXT i
   Chan1! = ((CH1pos! - CH1neg!) / ((NumRdgsPerStep - 1) * 2))
   DataCH1$(j) = STR$(Chan1!)
   k% = k% + (NumRdgsPerStep * 2)
NEXT j

'Printing results to a file.
OPEN "chan1.xls" FOR OUTPUT AS #1  'This places chan1.xls in your
   'current working directory.
OPEN "chan2.xls" FOR OUTPUT AS #2  'This places chan2.xls in your
   'current working directory.
FOR i = 1 TO CalcReadings
   PRINT #2, DataCH2$(i)
   PRINT #1, DataCH1$(i)
NEXT i
CLOSE #1   'Close the chan1.xls file.
CLOSE #2   'Close the chan2.xls file.
```

Section 8

Reading buffers

In this section:

Buffer operations

The 2182A has a buffer that stores from two to 1024 readings and units. It also stores the channel number for step/scan readings and overflow readings. The buffer fills with the specified number of readings and stops.

You can include statistical information in the recalled data. The available statistical information is minimum, maximum, peak-to-peak, average, and standard deviation. If an overflow reading is stored in the buffer, statistics are not calculated.

Readings are placed in the buffer after any math operations are performed. Math operations include relative, ratio, delta, mx+b, and percent.

Buffer data is overwritten each time the storage operation is selected. The data is volatile; it is not saved through a power cycle.

Measurements made during stepping or scanning operations are automatically stored in the buffer. There is no need to configure and enable the buffer. For information on stepping and scanning, refer to Stepping and scanning (on page 7-1).

When the interface is changed (GPIB to RS-232 or vice versa), all data in the buffer is cleared.

Store

To store readings:

1. Set up the instrument for the measurement.
2. Press the **STORE** key.
3. Use the cursor keys (◄ and ►) and the RANGE ▲ and ▼ keys to set the number of readings to store (2 to 1024).

4. Press **ENTER** to enable the buffer. If the instrument is in the immediate trigger mode, the storage process starts immediately. If it is in the external trigger mode, each input trigger (or press of TRIG key) stores a reading.

NOTE

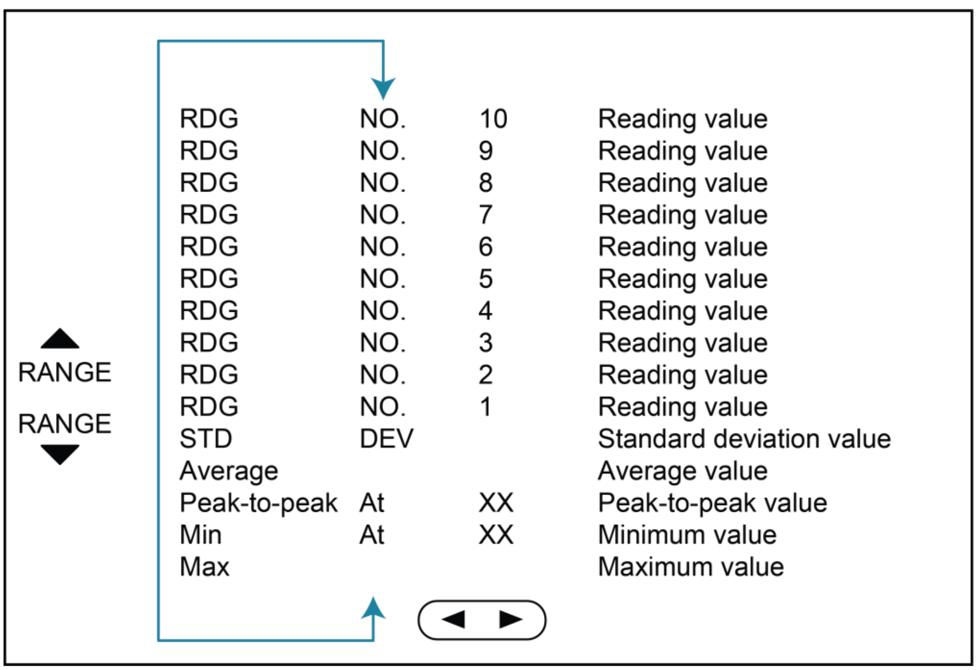
The asterisk (*) annunciator turns on to indicate that buffer operation is enabled. It turns off when the buffer is full.

Recall

To view stored readings and buffer statistics:

1. Press **RECALL**. The BUFFER annunciator turns on to indicate that stored readings are displayed. The arrow annunciator also turns on to indicate that additional data is available for viewing.
2. As shown in the following figure, use the RANGE ▲ and ▼ keys and the cursor ◄ and ► keys to navigate through the reading numbers, reading values, and buffer statistics. When the buffer statistics are displayed (standard deviation, average, peak-to-peak, minimum, or maximum), the STAT annunciator is on.
3. To return to the normal display, press **EXIT**.

Figure 64: Buffer locations



Buffer statistics

The minimum (MIN) and maximum (MAX) locations return the minimum and maximum readings stored in the buffer. It also indicates the buffer location of these readings.

Peak-to-peak is the absolute value of the difference between the MIN and MAX readings. It is calculated as follows:

Peak-to-Peak = |MAX − MIN|

Average is the mean of the buffer readings. Mean is calculated as follows:

$$y = \frac{\sum_{i=1}^{n} X_i}{n}$$

where:

- X_i is a stored reading.
- n is the number of stored readings.

Standard deviation is the standard deviation of the buffered readings. The equation used to calculate the standard deviation is:

$$y = \sqrt{\frac{\sum_{i=1}^{n} X_i^2 - \left(\frac{1}{n}\left(\sum_{i=1}^{n} X_i\right)^2\right)}{n-1}}$$

where:

- X_i is a stored reading.
- n is the number of stored readings.

The 2182A uses IEEE Std-754 floating point format for math calculations.

SCPI programming - buffer

`TRACe` subsystem commands store and recall readings in the buffer. Refer to the TRACe subsystem (on page 13-89) for information.

`CALCulate2` commands obtain statistics from the buffer data. Refer to the CALCulate2 subsystem (on page 13-13) for information.

NOTE

There is no SCPI command to obtain the peak-to-peak statistic. To get the peak-to-peak statistic, your program must calculate it from the maximum and minimum statistics.

Section 9

Triggering

In this section:

Trigger model

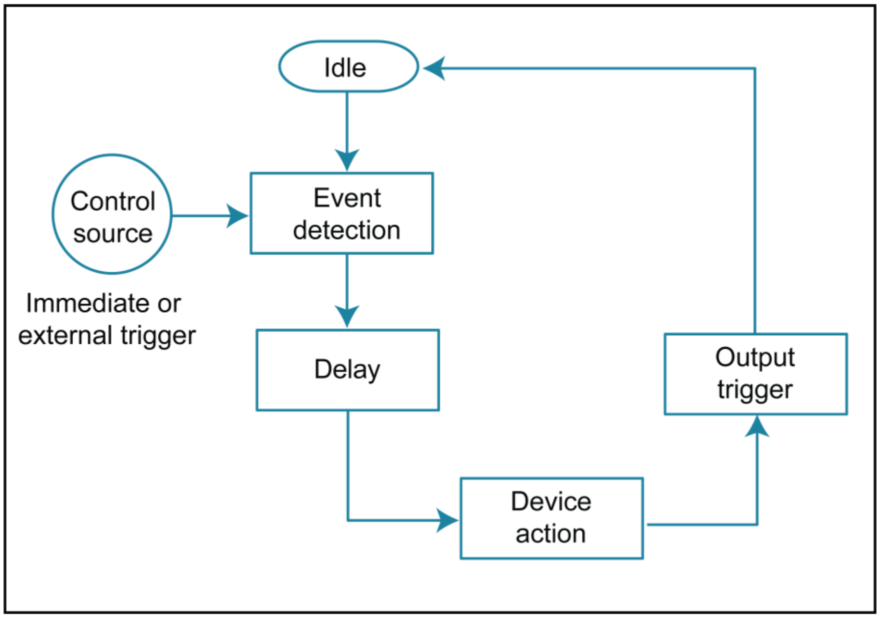
Triggering allows you to start and synchronize measure actions on one or more instruments with a trigger event or a combination of trigger events that you set. Triggering is controlled by a trigger model. It is called a trigger model because it is modeled after the SCPI commands used to control triggering.

You can operate the trigger model from the front panel or by using remote commands.

Front-panel trigger model

The flowchart in the following figure summarizes front-panel triggering. For stepping and scanning, the trigger model has additional control blocks. Refer to Stepping and scanning (on page 7-1) for detail.

Figure 65: Trigger model without stepping or scanning



Idle

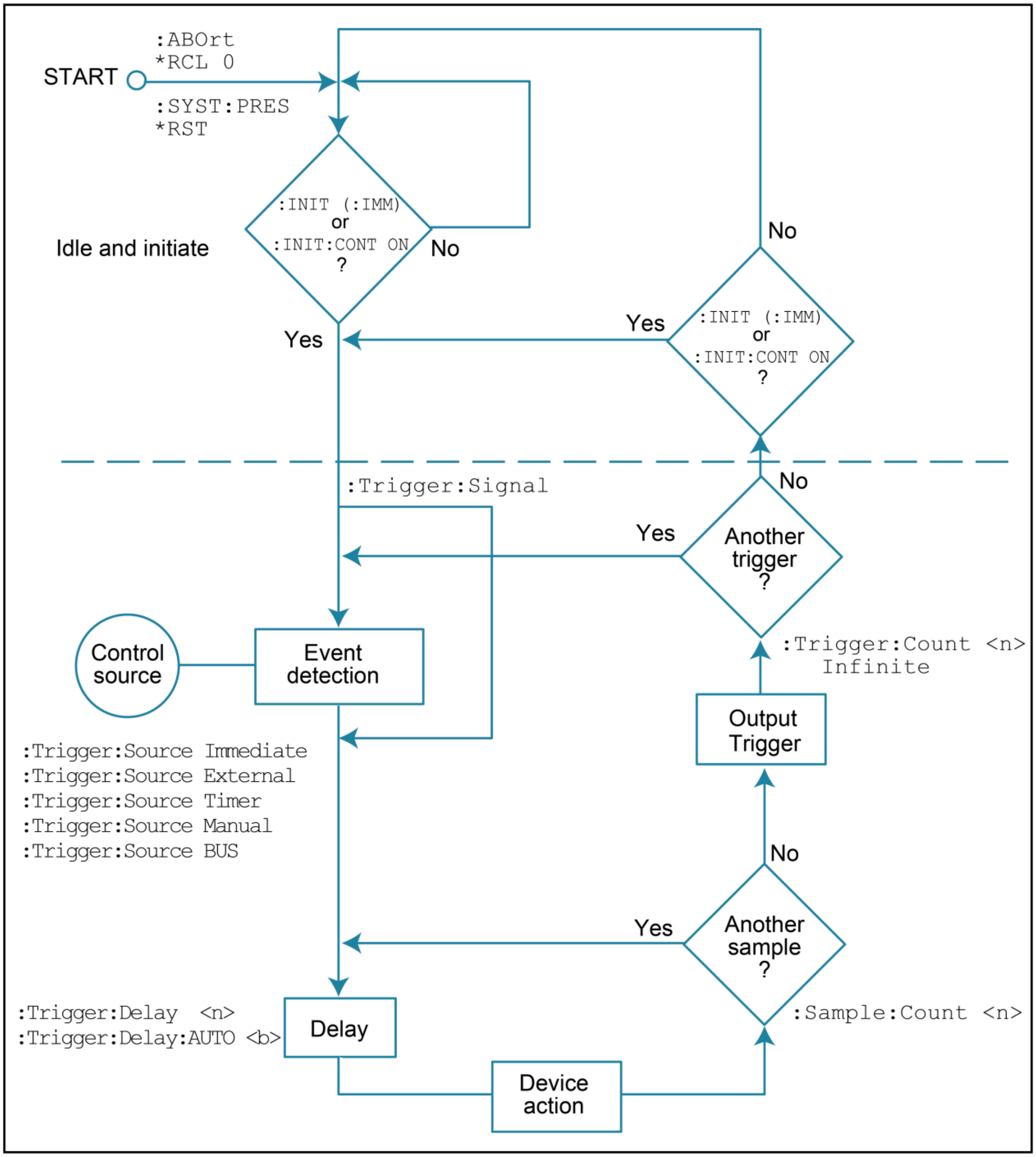
The instrument is considered to be in the idle state whenever it is not performing any measurements or scanning operations. From the front panel, the instrument is considered to be idle at the end of a step or scan operation when the reading for the last channel remains displayed. To restore triggers, press **SHIFT** and then **HALT**.

Once the 2182A is taken out of idle, operation proceeds through the trigger model.

Trigger model (remote operation)

The following figure describes how the 2182A operates when controlled remotely. Key SCPI commands are included in the trigger model flow chart. For descriptions of all the trigger commands, refer to TRIGger subsystem (on page 13-94).

Figure 66: Trigger model when operating remotely



Idle and initiate

The instrument is considered to be in the idle state whenever operation is at the top of the trigger model. As shown in the figure in Trigger model when operating remotely (on page 9-2), initiation needs to be satisfied to take the instrument out of idle. While in the idle state, the instrument cannot perform any measure or step and scan operations.

The following commands return operation to idle:

- `:ABORt`
- `:SYSTem:PRESet`
- `*RCL 0`
- `*RST`

What happens next depends on the state of initiation. If continuous initiation is already enabled, the instrument leaves the idle state.

- `SYSTem:PRESet` enables continuous initiation. Therefore, operation immediately leaves the idle state when it is sent.
- The `*RCL 0` command also enables continuous initiation if `INITiate:CONTinuous ON` is a user-saved default.
- `*RST` disables continuous initiation and the instrument remains in the idle state.

Either of the following two initiate commands takes the instrument out of the idle state:

- `:INITiate`
- `:INITiate:CONTinuous ON`

NOTE

While in remote, pressing the LOCAL key restores continuous front-panel operation.

Control source and event detection

The control source pauses operation until the programmed event occurs and is detected. You can choose an immediate or external control source.

When the control source is immediate, event detection is immediately satisfied and operation continues.

When the control source is external, an event must occur before trigger operation continues. Events include:

- An input trigger through the Trigger Link line EXT TRIG.
- Press of the front-panel TRIG key. The 2182A must be taken out of remote before it responds to the TRIG key. Use the LOCAL key or send `LOCAL 707` over the bus.
- A trigger command (`*TRG` or `GET`) received over the bus.

Delay

You can set a manual or automatic delay after event detection. With automatic delay, the 2182A selects a delay based on the selected voltage range. Auto delay is typically used for external scanning. The nominal delay is long enough to allow each relay to settle before making the measurement.

The auto delays are listed in the following table. There is no auto delay for temperature measurements.

Auto delay times

	Delay time (ms)	
Range	**DCV1**	**DCV2**
10 mV	1 ms	Not applicable
100 mV	1 ms	1 ms
1 V	1 ms	1 ms
10 V	1 ms	1 ms
100 V	5 ms	Not applicable

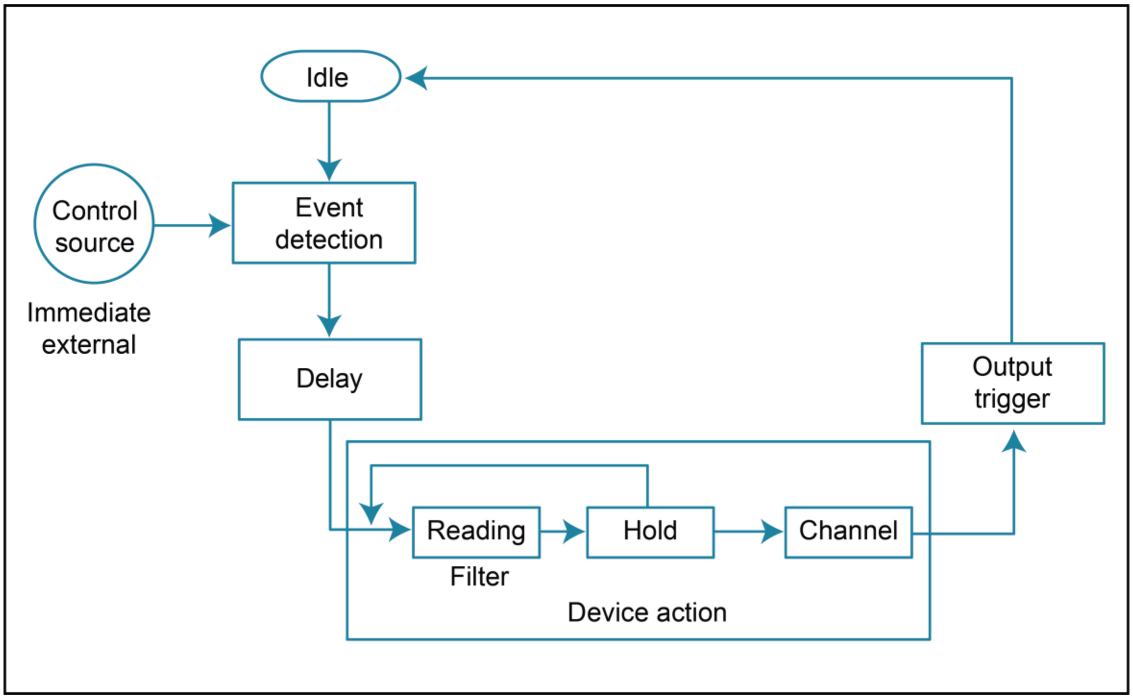
To set a delay from the front panel:

1. Pressing **SHIFT** and then **DELAY**. The present delay setting (AUTO or MANual) is displayed.
2. Press the ▲ and ▼ key to display the delay setting and press **ENTER**.
3. If you select **MANual**, enter the duration of the delay using the ◄, ►, ▲, and ▼ keys. The maximum is 99H:99M:99.999 s.
4. Press **ENTER** to accept the delay or **EXIT** for no change.

Device action

The primary device action is a measurement. However, the device action block can also include a filter, hold, or channel closure, as shown in the following figure.

Figure 67: Device actions



If the repeating filter is enabled, the instrument samples the specified number of reading conversions to yield a single filtered reading. If the moving average filter is enabled, the instrument samples the specified number of reading conversions. After the specified number of reading conversions is reached, only one reading conversion is made. After a reading is made, operation proceeds to hold.

Hold helps screen out reading anomalies and provides an autosettling time for switching relays. When hold is enabled, you select a window and count for the hold. In general, when a reading is outside the window it is rejected and operation returns to the start of the device action block. The hold count specifies how many readings must be in the window before it is accepted. After a hold reading is acquired, operation proceeds to channel control. See Reading hold (autosettle) (on page 9-7) for additional information.

When stepping or scanning is used, the last device action is channel control. Each open or close transition restarts the hold process. A reading for each channel is made after the relay settles.

Output trigger

The output trigger is available at the rear-panel Trigger Link connector. You can use this trigger to trigger another instrument to perform an operation, such as selecting the next channel for an external scan.

Reading hold (autosettle)

The reading hold feature screens out readings that are not in a specified reading window.

When hold is enabled, the HOLD annunciator is on, the first processed reading is made, and operation loops in the device action block. After the next reading is processed, the reading is compared to the hold window of the initial reading. If the reading is in the window, operation again loops in the device action block. This looping continues until the specified number (2 to 100) of consecutive readings are in the window. If one of the readings is not in the window, the instrument acquires a new initial reading and the hold process continues.

When a hold reading is acquired, an audible beep is sounded (if enabled) and the reading is considered to be a true measurement. The reading is shown on the display until an out of window reading occurs.

For remote operation or when scanning, the hold process seeks a new seed when it is satisfied and the readings are released. For front-panel operation, the hold process does not seek a new seed until the held condition is removed.

NOTE

Hold can only be used with Channel 1. When hold is enabled, Channel 2 becomes inoperative.

Reading hold is automatically disabled when ratio or delta is enabled.

To enable reading hold:

1. Press **SHIFT** and then **HOLD** to display the present window (0.01%, 0.1%, 1%, or 10%).
2. To change the window, press the ▲ and ▼ key to display the new window.
3. Press **ENTER**. The present hold count is displayed (2 to 100).
4. To change the hold count, use the ◄, ►, ▲, and ▼ keys to display the count.
5. Press **DCV1** to measure voltage on Channel 1.
6. Apply the test signal to Channel 1 of the 2182A. Once the signal becomes stable enough to satisfy the hold condition, the reading is released and the beeper sounds (if enabled).
7. Remove the hold condition by disconnecting the signal from Channel 1. Hold seeks a new seed.
8. To disable HOLD, press **SHIF**T and then **HOLD**.

Enable or disable the reading hold beeper

To enable the beeper for hold:

1. Press the LIMITS **ON/OFF** key to display the beeper selections (NEVER, OUTSIDE, and INSIDE).
2. To enable the beeper, use the ▲ and ▼ keys to display **OUTSIDE** or **INSIDE**.
3. Press **ENTER**. The instrument returns to the normal display state. Limit testing is enabled.
4. To disable limit testing, press ON/OFF.
5. To disable the beeper, use the ▲ and ▼ keys to display **NEVER**.

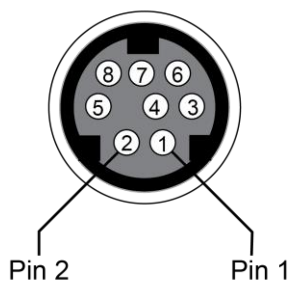
External triggering

The EX TRIG key selects triggering from the Trigger Link and the TRIG key. When EX TRIG is pressed, the TRIG annunciator lights and dashes are displayed to indicate the instrument is waiting for an external trigger. From the front panel, press the TRIG key to trigger a single reading. Press the EX TRIG key again to toggle to continuous triggers.

The 2182A uses two lines of the Trigger Link rear-panel connector as external trigger (EXT TRIG) input and voltmeter complete (VMC) output. The EXT TRIG line allows the 2182A to be triggered by other instruments. The VMC line allows the 2182A to trigger other instruments.

At the factory, line 1 is configured as VMC and line 2 as EXT TRIG. The connector pinout is shown in the following figure.

Figure 68: Rear-panel pinout



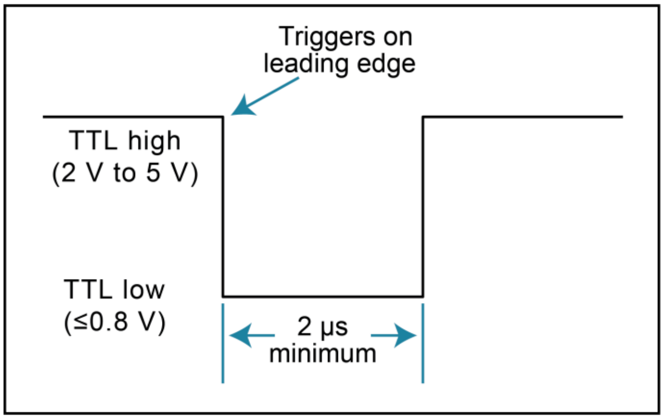
Pin number	Description
1	Voltmeter complete output
2	External trigger input
3	No connection; can be configured as an output instead of pin 1*
4	No connection; can be configured as an output instead of pin 2*
5	No connection; can be configured as an output instead of pin 1*
6	No connection; can be configured as an output instead of pin 2*
7	Signal ground
8	Signal ground

*See the *Model 2182A Service Manual* for details

External trigger

The EXT TRIG input requires a falling-edge TTL-compatible pulse with the specifications shown in the following figure. In general, you can use external triggers to control measure operations. For the 2182A to respond to external triggers, the trigger model must be configured for it.

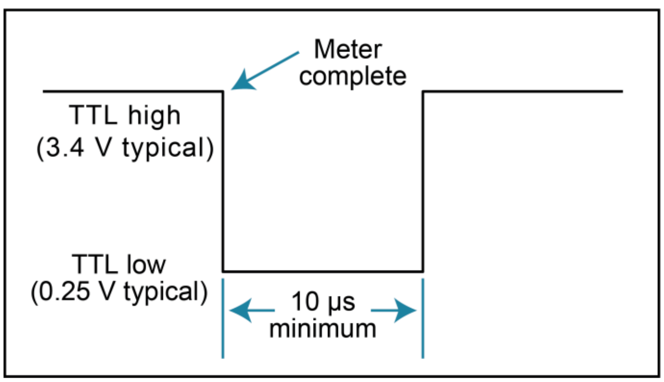
Figure 69: Trigger Link input pulse specifications (EXT TRIG)



Voltmeter complete

The voltmeter complete (VMC) output provides a TTL-compatible output pulse that you can use to trigger other instruments. The specifications for this trigger pulse are shown in following figure. Typically, you have the 2182A to output a trigger after the settling time of each measurement.

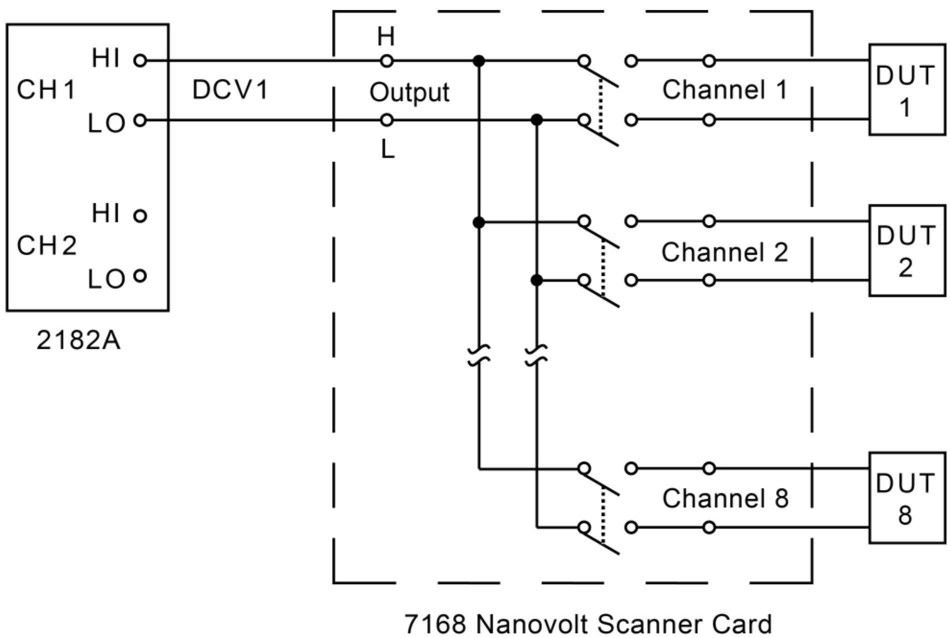
Figure 70: Trigger link output pulse specifications (VMC)



External triggering example

In a typical test system, you may want to close a channel and then measure the DUT connected to the channel with the 2182A. An example of this type of system is shown in the following figure, which uses a 2182A to measure eight DUTs switched by a Model 7168 Nanovolt Scanner Card in a Model 7001 or 7002 Switch System. See Stepping and scanning (on page 7-1) for details on external scanning.

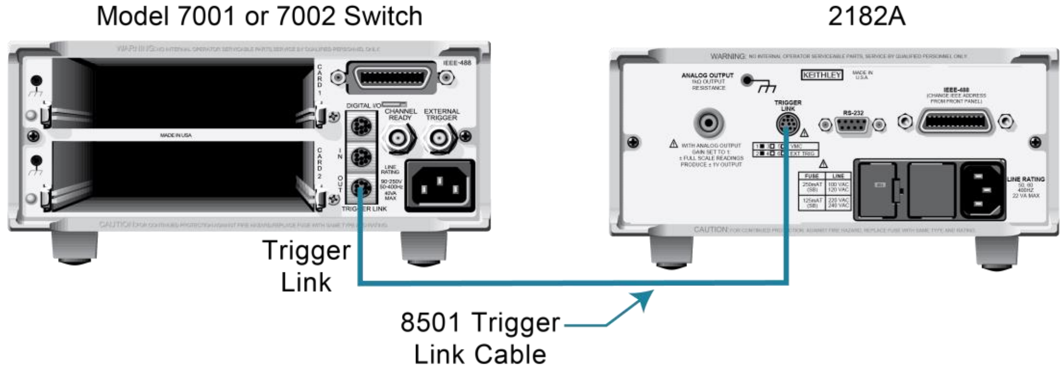
Figure 71: DUT test system



Trigger Link connections

The Trigger Link connections for this test system are shown in the following figure. The Trigger Link of the 2182A is connected to Trigger Link (either IN or OUT) of the Model 7001 or 7002. If you are using the default trigger settings of the 7001 or 7002, line #1 is an input and line #2 is an output. This complements the trigger lines on the 2182A.

Figure 72: Trigger Link connections



Front-panel settings

The following figure summarizes the front-panel operations to configure a scan for the External triggering example (on page 9-10).

Both instrument setups assume factory defaults. Channel 1 of the 2182A must be used for external scanning.

Refer to the figure that follows the steps for additional detail.

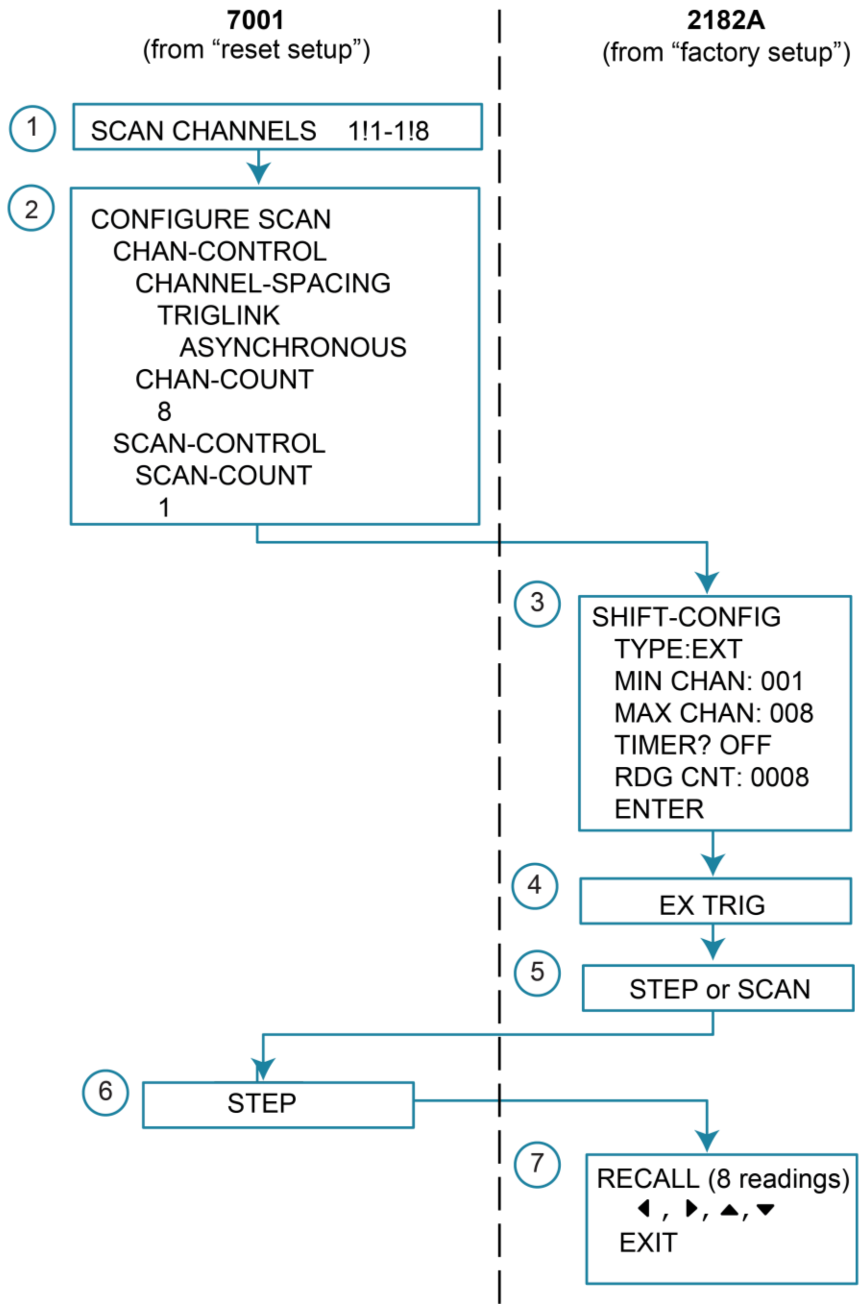
To set up the example on the Model 7001 Switch System:

1. Restore factory defaults.
2. Enter a scan list of channels 1 to 8 on card 1.
3. Configure the instrument for Trigger Link triggers and one scan of eight channels.

To set up the example on the 2182A:

1. Select **SHIFT** then **RESTR**.
2. Press the ▼ and ▲ keys to select **FACT** (factory).
3. Select **SHIFT** then **CONFIG**.
4. Configure an external scan of the first eight channels with no timer, and eight readings.
5. On the 2182A, press **EX TRIG** to set external triggers.
6. On the 2182A, press **STEP** or **SCAN**. The asterisk and STEP or SCAN annunciator lights.
7. On the Model 7001, press **STEP** to start channel closures.
8. After the scan, you can recall eight readings from the 2182A buffer.

Figure 73: External scanning example with Model 7001

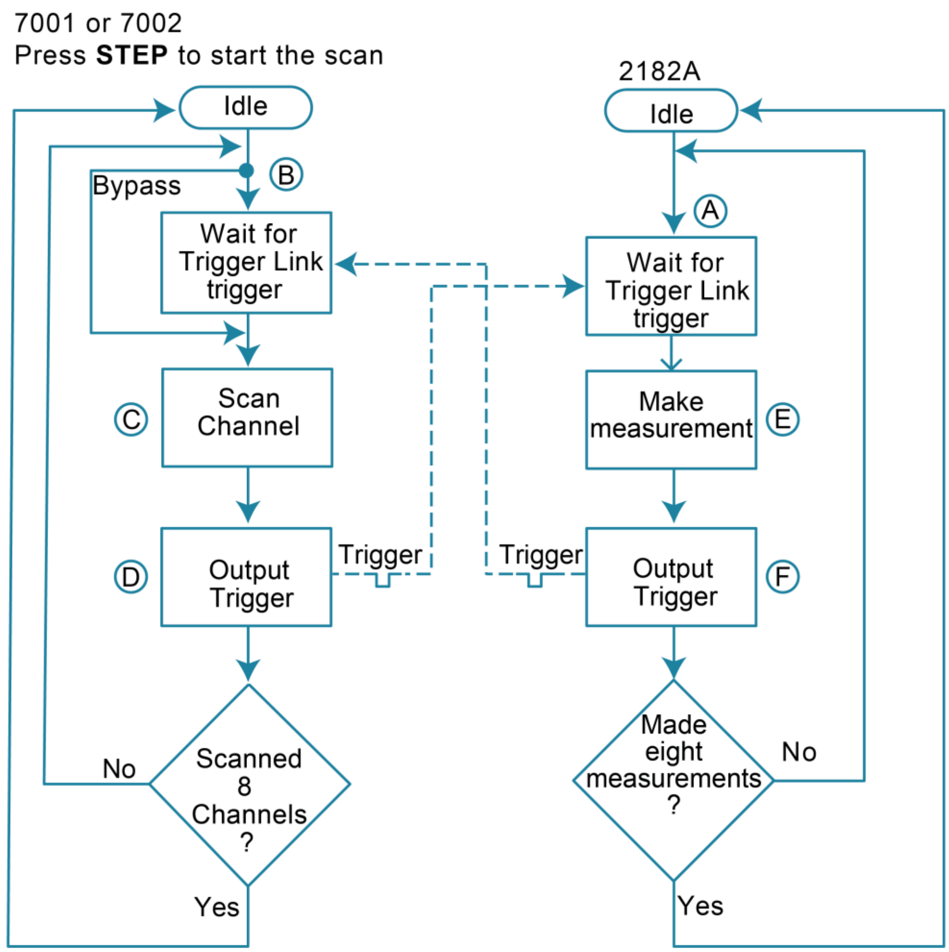


Running the example

To run the test and store readings in the 2182A with the instrument set for external triggers, press **STEP** or **SCAN**. The 2182A waits (with the asterisk annunciator lit) for an external trigger from the Model 7001 or 7002.

Press **STEP** on the 7001 or 7002 to take it out of idle and start the scan. The output pulse of the scanner triggers the 2182A to make a reading, store it, and send a trigger pulse. The following explanation of operation is referenced to the operation model shown in following figure.

Figure 74: Operation model for triggering example



A Pressing **EX TRIG** then **STEP** or **SCAN** on the 2182A places it at point A in the flowchart, where it is waiting for an external trigger.

B Pressing **STEP** on the 7001 or 7002 takes it out of the idle state and places operation at point B in the flowchart.

C For the first pass through the model, the scanner does not wait at point B for a trigger. Instead, it closes the first channel.

D After the relay settles, the 7001 or 7002 outputs a Channel Ready pulse. Since the instrument is programmed to scan eight channels, operation loops back to point B, where it waits for an input trigger.

E The 2182A operation is at point A waiting for a trigger. The output Channel Ready pulse from the 7001 or 7002 triggers the nanovoltmeter to measure DUT #1 (point E).

F After the measurement is complete, the 2182A outputs a completion pulse (point F) and then loops back to point A, where it waits for another input trigger.

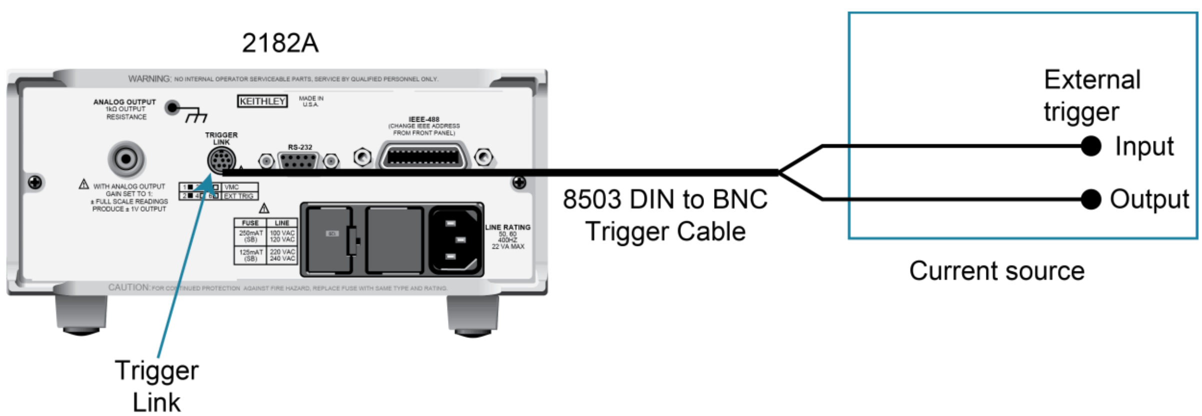
The trigger applied to the 7001 or 7002 from the 2182A closes the next channel in the scan. This triggers the nanovoltmeter to measure the next DUT. The process continues until all eight channels are scanned and measured.

External triggering with BNC connections

An adapter cable is available to connect the micro-DIN Trigger Link of the 2182A to instruments with BNC trigger connections. The Model 8503 DIN to BNC Trigger Cable has a micro-DIN connector at one end and two BNC connectors at the other end. The BNC cables are labeled VMC (trigger line 1) and EXT TRIG (trigger line 2).

The following figure shows how to connect a current source to the Trigger Link of the 2182A using the adapter cable. When used with the STEP mode of the Model 6220, you can perform synchronized source-measure operations without the use of a computer. Whenever the Model 6220 receives a trigger from the 2182A, it steps to the next current source value.

Figure 75: DIN to BNC trigger cable



NOTE

If the 2182A trigger line configuration is changed from the factory setting, you must use the Model 8502 Trigger Link Adapter to interface with instruments that have BNC trigger connections. It has two micro-DIN connectors and six BNC connectors, one for each trigger line.

Trigger model operation

Once the instrument is taken out of idle, operation proceeds through the trigger model to the device action. In general, the device action includes a measurement and, when stepping/scanning, closes the next channel.

Control source: As shown in the figure in Trigger model (remote operation) (on page 9-2), a control source holds up operation until the programmed event occurs. The control source options are as follows:

- **IMMediate:** Event detection is immediately satisfied, allowing operation to continue.
- **MANual:** Event detection is satisfied by pressing the TRIG key. The 2182A must be in LOCAL mode for it to respond to the TRIG key. To return the instrument to local control, press the LOCAL key or send LOCAL 7 over the bus.
- **TIMer:** Event detection is immediately satisfied on the initial pass through the loop. Each subsequent detection is satisfied when the programmed timer interval (0 s to 999999.999 s) elapses. The timer source is only available during step/scan operation. The timer resets to its initial state when the instrument goes into the normal mode of operation or into the idle state.
- **EXTernal:** Event detection is satisfied when an input trigger through the TRIGGER LINK connector is received by the 2182A.
- **BUS:** Event detection is satisfied when a bus trigger (`GET` or `*TRG`) is received by the 2182A.

Delay and device action: These blocks of the trigger model operate the same for front panel and GPIB operation. See Trigger model (on page 9-1) for operating information on these trigger model blocks. See Reading hold (autosettle) (on page 9-7) for details on hold.

Counters: Programmable counters repeat operations in the trigger model. For example, if you perform a 10-channel scan, the sample counter is set to 10. Operation continues until all 10 channels are scanned and measured. If you wanted to repeat the scan three times, set the trigger counter to three.

For a sample count value >1, the sample readings are automatically stored in the buffer. For example, with the sample count set to 5, the five measured readings are stored in the buffer. If the trigger model is configured to repeat the sample readings (for example, the trigger count is set to 2), the five new readings overwrite the original five readings in the buffer.

Output trigger: The 2182A sends one or more output triggers. The output trigger is applied to the Trigger Link connector on the rear panel. It can be used to trigger an external instrument to perform an operation.

You can configure the trigger model to output a trigger after the completion of a series of measurements or after every measurement. For example, with the sample counter set to 10 and the trigger counter set to one, a trigger is sent after the 10 measurements are made. If the trigger counter is set to 10 and the sample counter is set to 1, a trigger is sent after each measurement.

Triggering commands

Commands for triggering are summarized in the following table. For complete command descriptions, refer to the SCPI command reference TRIGger subsystem (on page 13-94).

Commands	Description
:ABORt	Stops trigger model commands.
:INITiate:CONTinuous <b>	Enables or disables continuous initiation: OFF or ON.
:INITiate [:IMMediate]	Initiates one trigger cycle.
:SAMPle:COUNt <n>	Sets the sample count: 1 to 1024.
:SENSe[1]:HOLD:COUNt	Sets the reading hold count: 2 to 100.
:SENSe[1]:HOLD:STATe	Selects the reading hold state: OFF or ON.
:SENSe[1]:HOLD:WINDow	Sets the reading hold window: 0.01 to 20.
:TRIGger[:SEQuence]:COUNt <n>	Sets the trigger count: 1 to 9999 or INF.
:TRIGger[:SEQuence]:DELay <n>	Sets the trigger delay in seconds: 0 to 999999.999
:TRIGger[:SEQuence]:DELay:AUTO	Enables or disables auto delay: OFF or ON.
:TRIGger[:SEQuence]:SIGNal	Bypasses the control source.
:TRIGger[:SEQuence]:SOURce	Selects the control source: IMMediate, TIMer, MANual, BUS, or EXTernal.
:TRIGger[:SEQuence]:TIMer <n>	Sets the trigger timer interval in seconds: 0 to 999999.99.

Programming example

The following program fragment triggers 10 readings, stores them in the buffer, and displays them on the computer monitor.

```
CALL SEND(7,"*RST",status%)   'Restore *RST defaults.
CALL SEND(7,":TRIG:DEL 0.5",status%)   'Set delay for 0.5 s.
CALL SEND(7,":SAMP:COUN 10",status%)   'Set sample count to 10.
CALL SEND(7,":READ?",status%)   'Trigger and request readings.
reading$ = SPACE$(80)
CALL ENTER(reading$, length%, 7,status%)   'Address 2182A to talk.
PRINT reading$   'Display the 10 readings on
      'the computer.
```

Section 10

Limits

In this section:

Limit operations

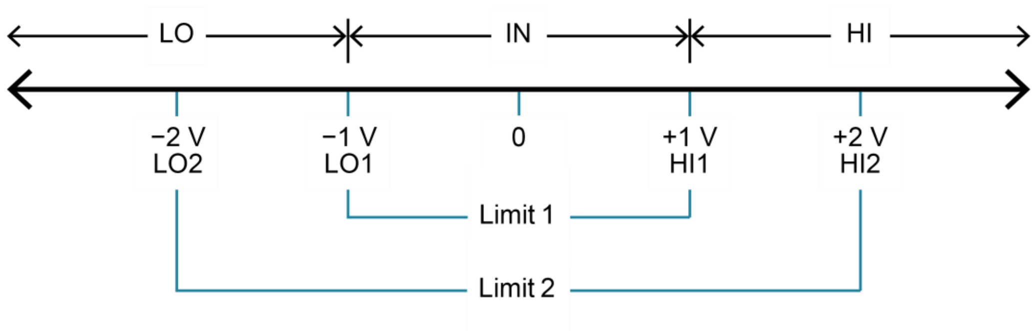
Limit 1 and Limit 2 operations set and control the values that determine the HI, IN, or LO status of subsequent measurements. Both limit tests use uses high and low limits. However, the HI, IN, or LO status message only applies to Limit 1. The limit test is performed on the result of an enabled relative offset, mx+b, or percent operation.

The following figure illustrates the factory default values of the limits.

Limit 1: HI1 = +1 V and LO1 = −1 V

Limit 2: HI2 = +2 V and LO2 = −2 V

Figure 76: Default limits



For the limits shown in the figure above, a reading of 1.5 V is outside Limit 1 (which is the primary limit). Therefore, the message `HI` is displayed.

A beeper is also available for limit testing. The beeper has the following options:

- **NEVER:** Disables the beeper.
- **OUTSIDE:** The beeper sounds when the reading is outside (HI or LO) Limit 1. In the figure above, a 1.5 V reading is outside (HI) Limit 1, so the beeper sounds.
- **INSIDE:** The beeper sounds when the reading is inside Limit 1 or Limit 2. If the reading is inside Limit 1, the beeper sounds at its normal pitch. If the reading is outside Limit 1 but inside Limit 2, the beeper sounds at a lower pitch. The beeper will not sound for readings outside Limit 2. For the limits shown in the figure above, a 0.5 V reading sounds the beeper at its normal pitch, a 1.5 V reading sounds the beeper at a lower pitch, and a 2.5 V reading does not sound the beeper.

NOTE

To use the Limit 2 test, the INSIDE beeper mode must be selected. With NEVER or OUTSIDE selected, Limit 2 is (in effect) disabled.

Setting limit values

To enter high and low limit values using the front panel:

1. Press the Limits **VALUE** key to view the present HI1 limit value.
   `HI1:+1.000000 ^    (default)`
2. To change the HI1 limit, use the cursor keys (◄ and ►) and the manual range keys (▲ and ▼) to display the value. Move the cursor to the rightmost position (^) and use the (▲ and ▼) keys to move the decimal point. To change the polarity, place the cursor on the polarity sign, then press ▲ or ▼ to toggle the polarity.
3. Press **ENTER** to view the present LO1 limit value.
   `LO1:-1.000000 ^    (default)`
4. Enter the value for this low limit.
5. Press **ENTER** to view the present HI2 limit value.
   `HI2:+2.000000 ^    (default)`
6. Enter the value for this high limit.
7. Press **ENTER** to view the present LO2 limit value.
   `LO2:-2.000000 ^    (default)`
8. Enter the value for this low limit and press **ENTER** to return to the normal display.

Enabling limits

To turn on limits operation:

1. Press the **Limits ON/OFF** key to view the present beeper status.
2. To change the beeper setting, use the ▲ and ▼ keys to display **NEVER**, **OUTSIDE**, or **INSIDE**.
3. Press **ENTER** to return to the normal display. The reading and the HI, IN, or LO status is displayed.
4. To disable limits, press the **ON/OFF** key.

SCPI programming - limits

For remote operation, the testing capabilities of Limit 1 and Limit 2 are the same. Information on the commands to configure and control limit testing are provided in CALCulate3 subsystem (on page 13-16).

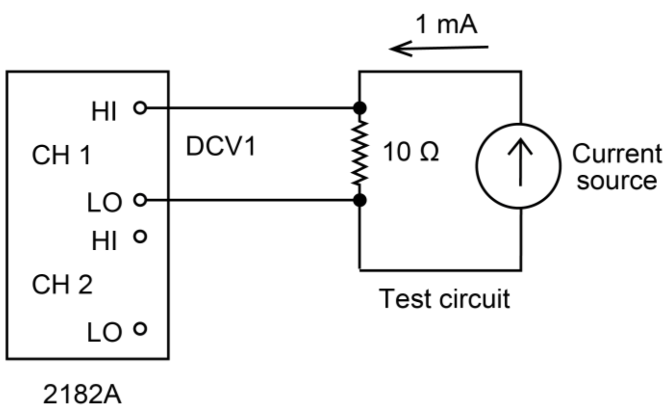
NOTE

When testing limits remotely, the front-panel HI, IN, and LO status messages only apply to Limit 1. Also, if the front-panel beeper is set for OUTSIDE or INSIDE, it operates according to its front-panel definition.

Application for sorting resistors

You can use limits to sort resistors. The following figure shows a basic setup to test 10 Ω resistors. The Model 6220 sources a constant 1 mA through the resistor and the 2182A measures the voltage drop.

Figure 77: Setup to test 10 Ω resistors



This application sorts a batch of 10 Ω resistors into three bins. Bin 1 is for resistors that are within 1% of the nominal value. Bin 2 is for resistors that exceed 1% tolerance, but are within 5%. Bin 3 is for resistors that exceed 5% tolerance.

Limit 1 tests for the 1% tolerance and Limit 2 tests for the 5% tolerance. The 2182A does not directly measure resistance, so the tolerances have to be converted to voltage values.

The voltage drop across a nominal 10 Ω resistor is calculated as follows:

V_{NOM} = 10 Ω × 1 mA
= 10 mV

The voltage values for the 1% and 5% tolerances are calculated as follows:

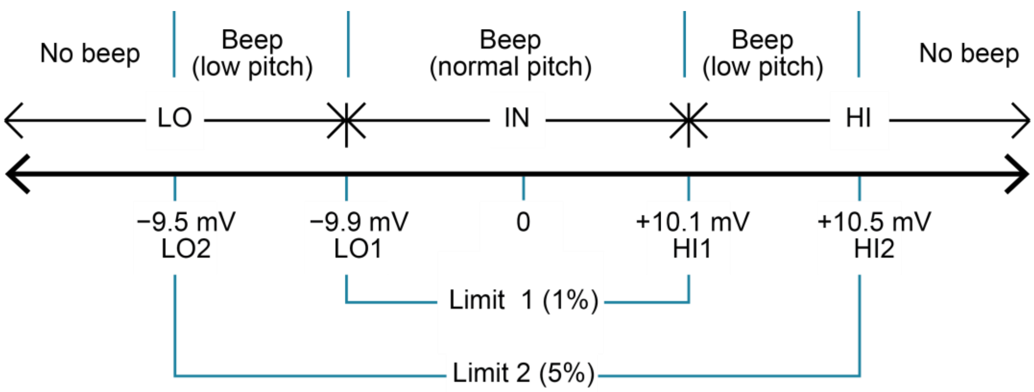
$V_{1\%}$	$= 10\text{ mV} \times 1\%$	$V_{5\%}$	$= 10\text{ mV} \times 5\%$
	$= 0\text{ mV} \times 0.01$		$= 10\text{ mV} \times 0.05$
	$= 0.1\text{ mV}$		$= 0.5\text{ mV}$

Finally, the high and low limits are calculated as follows:

HI Limit 1	$= 10\text{ mV} + V_{1\%}$	HI Limit 2	$= 10\text{ mV} + V_{5\%}$
	$= 10\text{ mV} + 0.1\text{ mV}$		$= 10\text{ mV} + 0.5\text{ mV}$
	$= 10.1\text{ mV}$		$= 10.5\text{ mV}$
LO Limit 1	$= 10\text{ mV} - V_{1\%}$	LO Limit 2	$= 10\text{ mV} - V_{5\%}$
	$= 10\text{ mV} - 0.1\text{ mV}$		$= 10\text{ mV} - 0.5\text{ mV}$
	$= 9.9\text{ mV}$		$= 9.5\text{ mV}$

The limits are illustrated in the following figure.

Figure 78: Limits to sort 10 Ω resistors (1%, 5%, and >5%)



Beeper mode: INSIDE

For front-panel operation, you must use the inside beeper mode. A normal pitch beep and the message `IN` indicates that the resistor is within the 1% tolerance limit. This 1% resistor belongs in Bin 1. A low pitch beep and the `HI` or `LO` message indicates that the resistor is >1% tolerance but <5% tolerance. This 5% resistor belongs in Bin 2. For resistors >5%, no beeper sounds. Place these resistors in Bin 3.

For remote operation, make sure both Limit 1 and Limit 2 are enabled. The following table evaluates the three possible pass/fail combinations for this application.

Limit 1 result	Limit 2 result	Resistor tolerance	Bin assignment
Pass	Pass	1%	1
Fail	Pass	5%	2
Fail	Fail	>5%	3

A fail condition must be reset before testing the next resistor. Fail can be reset manually or automatically. See:CALCulate3:LIMit2:CLEar:AUTO (on page 13-17) and :CALCulate3:LIMit[1]:CLEar[:IMMediate] (on page 13-21).

To maximize handling speed, quick-disconnect test clips are typically used for the resistor connections. Unfortunately, these connections may contribute enough thermal EMFs to corrupt the measurement. You can use the relative offset feature of the 2182A to null out this offset.

To use the relative offset feature:

1. Connect the test circuit. Leave the current source disconnected.
2. Using the lowest possible range (or autorange) measure the offset voltage.
3. Press **REL** to zero the display of the 2182A.
4. Connect the current source and test the resistors.

NOTE

If all resistor leads in the batch are made of the same metal, the relative offset value obtained for the first resistor should be satisfactory for each subsequent resistor.

Section 11

Maintenance

In this section:

Introduction

This section describes routine maintenance of the instrument that an operator can perform.

Cleaning input connectors

The CHANNEL connector on the front panel connects the 2182A to external test circuits. This connector mates to the connector on the 2107 input cable.

The contacts of the CHANNEL connector are made of copper. These copper-to-copper connections minimize thermal EMFs. However, exposed copper is susceptible to oxidation, which could cause measurement errors.

NOTE

Clean the contacts whenever the cable is disconnected from the 2182A for an extended period. Use DeoxIT® DL100L, included with the 2182A, to clean the contacts.

NOTE

To minimize the accumulation of oxides on CHANNEL connector contacts, keep the contacts mated whenever possible.

To clean the CHANNEL connector contacts:

1. Turn off the 2182A.
2. Disconnect the power cord and any other cables or wires connected to the instrument.
3. Stand the 2182A so that the front panel is facing up.
4. Apply one drop of DeoxIT to each of the four contacts of the input connector on the 2182A. You can use a clean wire (such as a resistor lead) to carry a drop of the solution from the bottle of DeoxIT to the connector.
5. Wipe off any excess DeoxIT using a clean cloth.
6. Connect and disconnect the cable from the 2182A several times to distribute the DeoxIT solution and to clean the input cable contacts.

Line fuse replacement

A fuse on the 2182A rear panel protects the power line input of the instrument. See the following instructions to replace the fuse. You do not need to return your instrument for service if the fuse is damaged.

WARNING

Disconnect the line cord at the rear panel and remove all test leads connected to the instrument before replacing a line fuse. Failure to do so could expose the operator to hazardous voltages that could result in personal injury or death.

Use only the correct fuse type. Failure to do so could result in injury, death, or instrument damage. If the instrument repeatedly blows fuses, locate and correct the cause of the problem before replacing the fuse. See the *Model 2182A Service Manual* for troubleshooting information. If the fuse continues to become damaged, return the instrument to Keithley Instruments for repair.

The powerline fuses for the 2182A are listed in the following table.

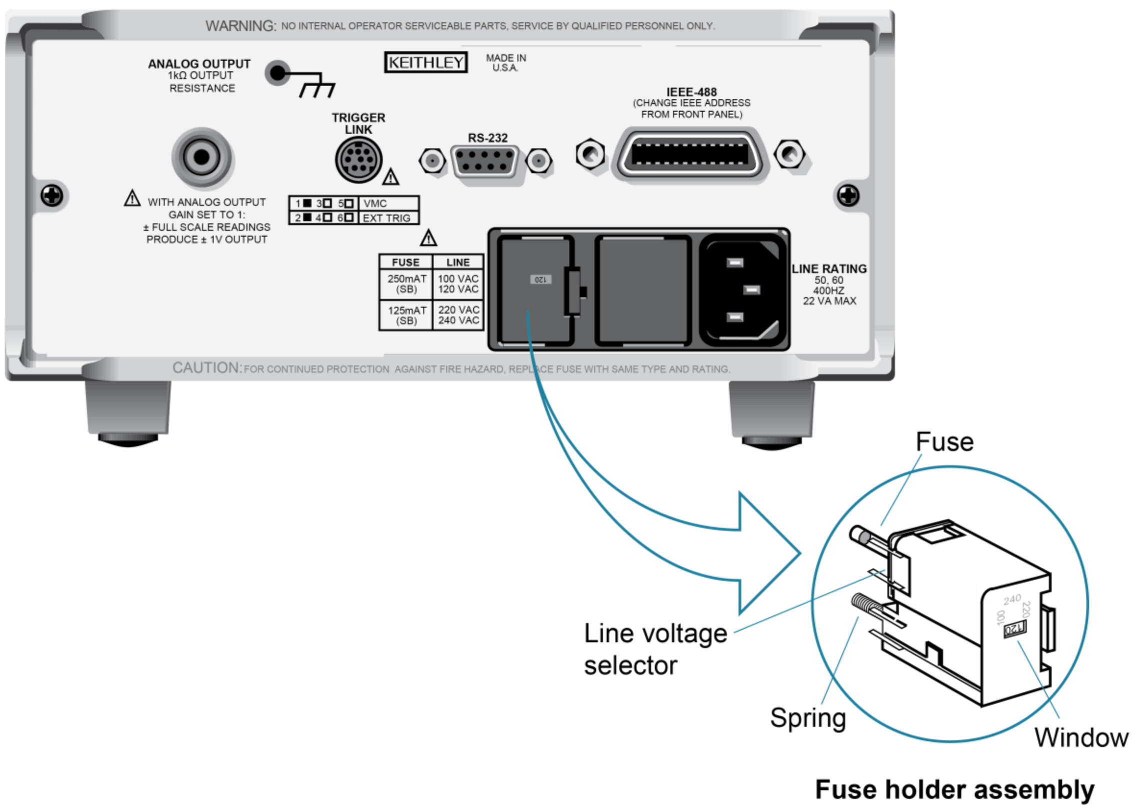
Fuse ratings

Line voltage	Fuse rating	Part number
100 V or 120 V	0.25 A, slow blow, 5 mm × 20 mm	Littelfuse 239.250
220 V or 240 V	0.125 A, slow blow, 5 mm × 20 mm	Schurter 0034.3108

To replace the line fuse:

1. Power off the instrument.
2. Remove all test leads connected to the instrument.
3. Remove the line cord.
4. Make note of the voltage displayed in the fuse holder assembly window.
5. Place the tip of a flat-blade screwdriver into the power module by the fuse holder assembly.
6. Gently push in and move to the left. Release pressure on the assembly. Its internal spring pushes it out of the power module, as shown in the following figure.

Figure 79: Fuse and line voltage selector



7. Remove the blown fuse.
8. Replace the fuse.
9. Install the fuse holder assembly into the power module by pushing it in until it locks in place.

Upgrading the firmware

For instruction on upgrading the firmware, refer to the release notes for the firmware. Firmware and release notes are available on tek.com/keithley. Filter by **Software** to locate the firmware.

Status and error messages

In the following tables, the Event codes are:

- EE = error event
- SE = status event
- SYS = system error event

Errors and status messages with a positive number are instrument-dependent. Negative errors are reserved by SCPI. For more information on SCPI-confirmed messages, see the IVI Foundation website (ivifoundation.org).

Error messages

Number	Description	Event
-440	Query unterminated after indefinite response	EE
-430	Query deadlocked	EE
-420	Query unterminated	EE
-410	Query interrupted	EE
-363	Input buffer overrun	SYS
-350	Queue overflow	SYS
-330	Self-test failed	EE
-315	Configuration memory lost	EE
-314	Save/recall memory lost	EE
-260	Expression error	EE
-241	Hardware missing	EE
-230	Data corrupt or stale	EE
-225	Out of memory	EE
-224	Illegal parameter value	EE
-223	Too much data	EE
-222	Parameter data out of range	EE
-221	Settings conflict	EE
-220	Parameter error	EE
-215	Arm deadlock	EE
-214	Trigger deadlock	EE
-213	Init ignored	EE
-212	Arm ignored	EE
-211	Trigger ignored	EE

Number	Description	Event
-210	Trigger error	EE
-202	Settings lost due to rtl	EE
-201	Invalid while in local	EE
-200	Execution error	EE
-178	Expression data not allowed	EE
-171	Invalid expression	EE
-170	Expression error	EE
-168	Block data not allowed	EE
-161	Invalid block data	EE
-160	Block data error	EE
-158	String data not allowed	EE
-154	String too long	EE
-151	Invalid string data	EE
-150	String data error	EE
-148	Character data not allowed	EE
-144	Character data too long	EE
-141	Invalid character data	EE
-140	Character data error	EE
-128	Numeric data not allowed	EE
-124	Too many digits	EE
-123	Exponent too large	EE
-121	Invalid character in number	EE
-120	Numeric data error	EE
-114	Header suffix out of range	EE
-113	Undefined header	EE
-112	Program mnemonic too long	EE
-111	Header separator error	EE
-110	Command header error	EE
-109	Missing parameter	EE
-108	Parameter not allowed	EE
-105	GET not allowed	EE
-104	Data type error	EE
-103	Invalid separator	EE
-102	Syntax error	EE
-101	Invalid character	EE
-100	Command error	EE

Status messages

Number	Description	Event
+000	No error	SE
+101	Operation complete	SE
+121	Device calibrating	SE
+125	Device measuring	SE
+171	Waiting in trigger layer	SE
+174	Re-entering the idle layer	SE
+180	Filter settled	SE
+301	Reading overflow	SE
+302	Low limit 1 event	SE
+303	High limit 1 event	SE
+304	Low limit 2 event	SE
+305	High limit 2 event	SE
+306	Reading available	SE
+308	Buffer available	SE
+309	Buffer half full	SE
+310	Buffer full	SE

Calibration messages

Number	Description	Event
+400	10m vdc zero error	EE
+401	1 vdc zero error	EE
+402	10 vdc zero error	EE
+403	100 vdc zero error	EE
+404	10 vdc full scale error	EE
+405	-10 vdc full scale error	EE
+406	100 vdc full scale error	EE
+408	10 vdc ch2 high zero error	EE
+409	10 vdc ch2 low zero error	EE
+410	B_7_div100 ACAL error	EE
+411	B_0_div100 ACAL error	EE
+412	B_7_1 ACAL error	EE
+413	B_0_1 ACAL error	EE
+414	B_1_1 ACAL error	EE
+415	B_1_10 ACAL error	EE
+416	B_0_10 ACAL error	EE
+417	B_P1_10 ACAL error	EE
+418	B_P1_100 ACAL error	EE
+419	B_0_10 ACAL error	EE
+420	Analog output zero error	EE
+421	Analog positive gain error	EE
+422	Analog negative gain error	EE

Number	Description	Event
+423	B_0_100 ACAL error	EE
+430	Precal selection error	EE
+432	ACAL Temperature Error	EE
+438	Date of calibration not set	EE
+439	Next date of calibration not set	EE
+440	Gain aperture correction error	EE
+449	10 vdc ch2 Low-Q zero error	EE
+500	Calibration data invalid	EE
+510	Reading buffer data lost	EE
+511	GPIB address lost	EE
+512	Power-on state lost	EE
+514	DC calibration data lost	EE
+515	Calibration dates lost	EE
+516	Linearity precal lost	EE
+522	GPIB communication language lost	EE
+610	Questionable Calibration	EE
+611	Questionable Temperature Measurement	EE
+612	Questionable ACAL	EE
+800	RS-232 Framing Error detected	EE
+802	RS-232 Overrun detected	EE
+803	RS-232 Break detected	EE
+805	Invalid system communication	EE
+806	RS-232 Settings Lost	EE
+807	RS-232 OFLO: Characters Lost	EE
+808	ASCII only with RS-232	EE
+900	Internal System Error	EE
+953	DDC Uncalibrated Error	EE

DDC Status Model messages

Number	Description	Event
+960	DDC Mode IDDC Error	EE
+961	DDC Mode IDDCO Error	EE

Keithley 182 Serial Poll Byte Events

Number	Description	Event
+962	DDC Ready	SE
+963	DDC Reading Done	SE
+964	DDC Buffer Half Full	SE
+965	DDC Buffer Full	SE
+966	DDC Reading overflow	SE

Introduction to SCPI commands

In this section:

Programming syntax

The Standard Commands for Programmable Instruments (SCPI) standard is a syntax and set of commands that are used to control test and measurement devices. For more information, see the IEEE-488.2 and SCPI standards.

The following information describes some basic SCPI command information and how SCPI is used with the 2182A and presented in the 2182A documentation.

Commands that are listed as SCPI compliant have commands and parameters that are SCPI confirmed. Other commands are SCPI commands, but do not conform to the SCPI standard set of commands. It is not a recognized command by the SCPI consortium. SCPI confirmed commands that use one or more non-SCPI parameters are explained in the “Details” section of the command description.

Command words

Program messages are made up of one or more command words.

Some command words are enclosed in brackets ([ ]). These brackets denote an optional command word that does not need to be included in the program message. For example:
`:INITiate[:IMMediate]`

These brackets indicate that `:IMMediate` is an optional command word and does not have to used. When using optional command words in your program, do not include the brackets. For example, the above command can be sent in either of these ways:

```
:INITiate
```

```
:INITiate:IMMediate
```

Commands and command parameters

Common commands and SCPI commands may or may not use a parameter. Parameters are shown in angle brackets (< >). The following are some examples:

*SAV <NRf>	Parameter (NRf) required.
*RST	No parameter used.
:INITiate:CONTinuous <b>	Parameter <b> required.
:SYSTem:PRESet	No parameter used.

Put at least one space between the command word and the parameter. Do not include the angle brackets when sending the command.

Parameter types

The following table lists the common parameter types.

<b>	Boolean. Enables or disables an instrument operation. 0 or OFF disables the operation, and 1 or ON enables the operation. For example, the following command enables continuous initiation: :INITiate:CONTinuous ON
<name>	Name parameter. Select a parameter name from a listed group. For example, to select the percentage calculation: :CALCulate:FORMat PERCent
<NRf>	Numeric representation format. A number that can be expressed as an integer (such as 8), a real number (such as 23.6), or an exponent (2.3E6). For example, to set the buffer size to 20: :TRACe:POINts 20
<n>	Numeric value. Can consist of an NRf number or one of the following name parameters: DEFault, MINimum, or MAXimum. When the DEFault parameter is used, the instrument is programmed to the *RST default value. When the MINimum parameter is used, the instrument is programmed to the lowest allowable value. When the MAXimum parameter is used, the instrument is programmed to the largest allowable value. For example, to set the timer to 100 ms: :TRIGger:TIMer 0.1 To set the timer to the default of 0.1 s: :TRIGger:TIMer DEFault To set the timer to the minimum of 1 ms: :TRIGger:TIMer MINimum To set the timer to the maximum of 999999.999 s: :TRIGger:TIMer MAXimum
<list>	Specify one or more switching channels. For example, to specify a scan list of channels 1 to 5, send: :ROUT:SCAN (@1:5)

Long-form and short-form versions

This documentation shows SCPI commands with both uppercase and lowercase letters. The uppercase letters are the required elements of a command. The lowercase letters are optional. If you choose to include the letters that are shown in lowercase letters, you must include all of them.

When you send a command to the instrument, letter case is not important. You can mix uppercase and lowercase letters in program messages.

For example, you can send the command `:SYSTem:PRESet` in any of the following formats:

```
:SYSTem:PRESet
:SYST:PRES
:SYSTem:PRES
:syst:pres
```

Query commands

SCPI queries have a question mark (`?`) after the command. You can use the query to determine the present value of the parameters of the command or to get information from the instrument.

For example, to determine what the present setting for the timer is, you can send:

```
:TRIGger:TIMer?
```

This query returns the present setting.

If the command has `MINimum`, `MAXimum`, and `DEFault` options, you can use the query command to determine what the minimum, maximum, and default values are. In these queries, the `?` is placed before the `MINimum`, `MAXimum`, or `DEFault` parameter. For example, to determine the default value for the timer, you can send:

```
:TRIGger:TIMer? DEFault
```

If you send two query commands without reading the response from the first, and then attempt to read the second response, you may receive some data from the first response followed by the complete second response. To avoid this, do not send a query command without reading the response. When you cannot avoid this situation, send a device clear before sending the second query command.

Program messages

A program message consists of one or more command words sent by the computer to the instrument.

Each common command is a three-letter acronym preceded by an asterisk (`*`). Common commands are described in Common commands (on page 14-1).

SCPI commands consist of several levels. The following discussion uses the `:STATus` subsystem to explain how command words are structured to create program messages.

Command structure

`:STATus`	Path (root)
`  :OPERation`	Path
`    :ENABle  <NRf>`	Command and parameter
`    :ENABle?`	Query command
`  :PRESet`	Command

Single command messages

The `:STATus` command structure has three levels. The first level is made up of the root command (`:STATus`) and serves as a path. The second level is made up of another path (`:OPERation`) and a command (`:PRESet`). The third level is made up of one command for the `:OPERation` path. You can execute these commands by sending the following program messages:

```
:STAT:OPER:ENAB  1
:STAT:OPER:ENAB?
:STAT:PRES
```

In each of these program messages, the path pointer starts at the root command (`:STAT`) and moves down the command levels until the command is executed.

Multiple command messages

You can send multiple command messages in the same program message if they are separated by semicolons (;). The following is an example showing two commands in one program message:

```
:STAT:OPER; :STAT:OPER:ENAB 1
```

When this command is sent, the first command word is recognized as the root command (`:STAT`). When the next colon is detected, the path pointer moves to the next command level and executes the command. When the path pointer sees the colon after the semicolon (`;`), it resets to the root level.

Commands that are on the same command level can be executed without having to retype the entire command path. For example:

```
:STAT:OPER:ENAB 1; ENAB?
```

After the first command (`:ENAB`) is executed, the path pointer is at the third command level in the structure.

Command path rules

- Each new program message must begin with the root command unless it is optional, such as `[:SENSe]`. If the root is optional, treat the command word on the next level as the root.
- The colon (`:`) at the beginning of a program message is optional.
- The path pointer can only move down. It cannot be moved up a level. Executing a command at a higher level requires that you start over at the root command.

Using common commands and SCPI commands in the same message

You can use common commands and SCPI commands in the same message if they are separated by semicolons (`;`). A common command can be executed at any command level and does not affect the path pointer.

```
:STAT:OPER:ENAB 1; *ESE 1
```

Program message terminator (PMT)

Each program message must be terminated with a line feed (LF), end or identify (EOI), or LF+EOI. The bus hangs if your computer does not provide this termination. The following example shows how a program message must be terminated:

```
:ROUT:SCAN  (@1:5)  <PMT>
```

Command execution rules

- Commands execute in the order that they are presented in the program message.
- An invalid command generates an error and is not executed.
- Valid commands that precede an invalid command in a multiple command program message are executed.
- Valid commands that follow an invalid command in a multiple command program message are ignored.

Sending strings

If you are sending a string, it must begin and end with matching quotes (either single quotes or double quotes). To include a quote character as part of the string, type it twice with no characters in between.

Response messages

A response message is the message sent by the instrument to the computer in response to a query command program message.

After sending a query command, the response message is placed in the Output Queue. When the 2182A is addressed to talk, the response message is sent from the Output Queue to the computer.

Each response is terminated with a line feed (LF) and end or identify (EOI). The following example shows how a multiple response message is terminated:

```
0; 1; 1; 0;  <RMT>
```

Multiple response messages

If you send more than one query command in the same program message, the response messages for all the queries are sent to the computer when the 2182A is addressed to talk. The responses are sent in the order that the query commands were sent and are separated by semicolons (;). Items in the same query are separated by commas (,). The following example shows the response message for a program message that contains four single item query commands:

```
0; 1; 1; 0
```

Message exchange protocol

These rules summarize the message exchange protocol:

1. Always tell the 2182A what to send to the computer. To send information from the instrument to the computer:
   a. Send the appropriate query commands in a program message.
   b. Address the 2182A to talk.
2. The complete response message must be received by the computer before another program message can be sent to the 2182A.

Using the SCPI command reference

The SCPI command reference contains detailed descriptions of each of the SCPI commands that you can use to control your instrument. Each command description is broken into several standard subsections. The following figure shows an example of a command description.

Figure 80: SCPI command description example

:EXAMple:COMMand:STATe

This command is an example of a typical SCPI command that turns an instrument feature on or off.

Type	Affected by	Where saved	Default value
Command and query	Recall settings Instrument reset Power cycle	Save settings	1 (ON)

Usage

```
:EXAMple:COMMand:STATe <state>
:EXAMple:COMMand:STATE?
```

`<state>`	Disable the example feature: 0 or `OFF` Enable the example feature: 1 or `ON`

Details

This command is an example of a typical SCPI command that enables or disables a feature.

Example

`:EXAMple:COMMand:STATe ON`	Turn the example feature on.

Also see

:EXAMple:COMMand:UNIT (on page 6-100)

Each command listing is divided into five subsections that contain information about the command:

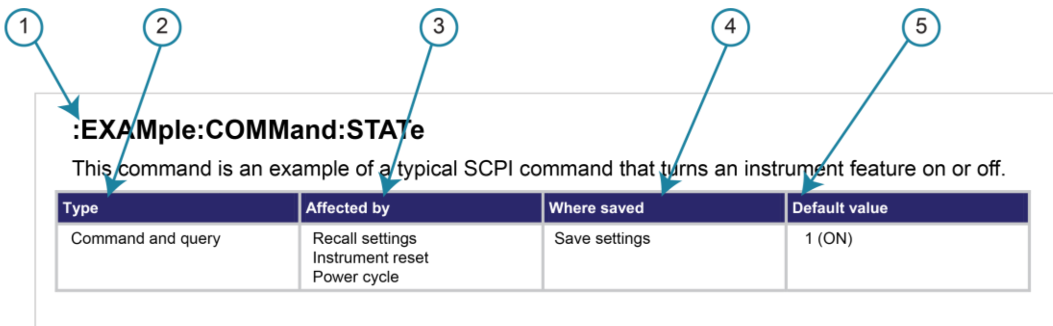
- Command name and summary table
- Usage
- Details
- Example
- Also see

The content of each of these subsections is described in the following topics.

Command name and summary table

Each instrument command description starts with the command name, followed by a table with relevant information for each command. Definitions for the numbered items are listed following the figure.

Figure 81: SCPI command name and summary table



1 **Instrument command name**. Signals the beginning of the command description and is followed by a brief description of what the command does.

2 **Type of command**. Options are:

- **Command only**. There is a command but no query option for this command.
- **Command and query**. The command has both a command and query form.
- **Query only**. This command is a query.

3 **Affected by**. Commands or actions that have a direct effect on the instrument command.

- **Recall settings.** If you send `*RCL` to recall the system settings, this setting is changed to the saved value.
- **Instrument reset.** When you reset the instrument, this command is reset to its default value. Reset can be done from the front panel or when you send `*RST`.
- **Power cycle.** When you power cycle the instrument, this command is reset to its default value.

4 **Where saved**. Indicates where the command settings reside once they are used on an instrument. Options include:

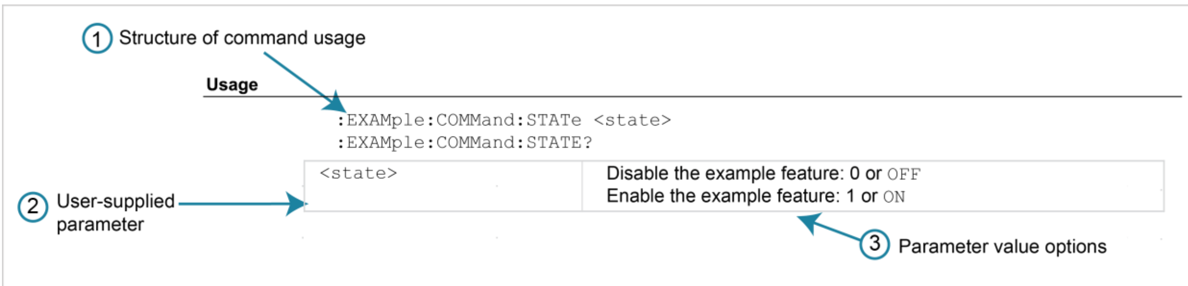
- **Not saved.** Command is not saved and must be sent each time you use it.
- **Nonvolatile memory.** The command is stored in a storage area in the instrument where information is saved even when the instrument is turned off.
- **Save settings.** This command is saved when you send the `*SAV` command.

5 **Default value:** Lists the default value for the command. The parameter values are defined in the Usage or Details sections of the command description.

Command usage

The Usage section of the remote command listing shows how to properly structure the command. Each line in the Usage section is a separate variation of the command usage; all possible command usage options are shown here.

Figure 82: SCPI command description usage identification



1. **Structure of command usage:** Shows the organization of the parts of the command.
2. **User-supplied parameters:** Indicated by angle brackets (< >).
3. **Parameter value options:** Descriptions of the options that are available for the parameter.

Command details

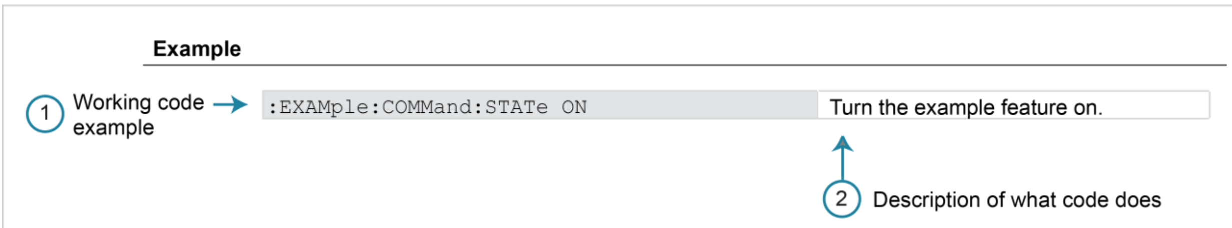
This section lists additional information you need to know to successfully use the command.

Figure 83: Details section of command listing

Example section

The Example section of the command description shows some simple examples of how the command can be used.

Figure 84: SCPI command description code examples

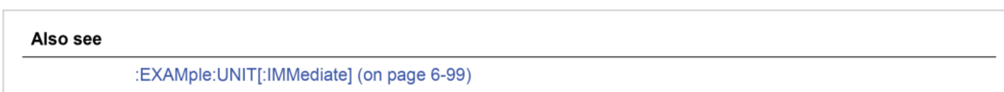


1. Example code that you can copy from this table and paste into your own application. Examples are generally shown using the short forms of the commands.
2. Description of the code and what it does. This may also contain the output of the code.

Related commands list

The **Also see** section of the remote command description provides links to commands that are related to the command.

Figure 85: SCPI related commands list example



QuickBASIC programming

Programming examples used throughout this manual presume Microsoft QuickBASIC version 4.5 (or higher) and a Keithley KPC-488.2. The 2182A must be set to address 07 for the IEEE-488 bus.

About program fragments

This documentation uses program fragments to demonstrate proper programming syntax. Only a fragment of the whole program is used to avoid redundancy.

At the beginning of each program, you must edit the following line to include the QuickBASIC libraries on your computer:

```
$INCLUDE: 'c:'b45\ieeeqb.bi'
```

Then initialize the interface card as address 21:

```
CALL INITIALIZE (21, 0)
```

Initialize also sends out an interface clear (IFC) to the entire GPIB system to initialize the other devices (see IFC (interface clear) (on page 2-14)).

A typical program fragment includes a `CALL SEND` command and a `CALL ENTER` command. The `CALL SEND` command sends a program message (command string) to the 2182A. If the program message includes a query command, then the `CALL ENTER` command is required to get the response message from the 2182A. The `CALL ENTER` command addresses the 2182A to talk. The following example program fragment demonstrates how to use `CALL SEND` and `CALL ENTER` commands. The commands assume address 07, which is the factory-set address of the 2182A.

```
CALL SEND (7, "*RST", status%)
CALL SEND (7, ":READ?", status%)
```

To display the response message, the computer reads the message and then prints it to the display as follows:

```
reading$ = SPACE$(80)
CALL ENTER (reading?, length%, 7, status%)
PRINT reading$
```

The following programming example shows how to use the above statements together.

```
$INCLUDE: 'c:'b45\ieeeqb.bi   ' Include QuickBASIC libraries.
CALL INITIALIZE (21, 0)    ' Initialize card as address 21.
CALL SEND (7, "*RST", status%)    ' Restore 2182A to *RST defaults.
CALL SEND (7, ":READ?", status%)    ' Trigger and request a reading.
reading$ = SPACE$(80)    ' Allocate room for data.
CALL ENTER (reading?, length%, 7, status%)
      ' Address 2182A to talk.
PRINT reading$   ' Display reading on computer.
```

Section 13

SCPI command reference

In this section:

:CONFigure:<function>

This command selects the active measure function, voltage or temperature.

Type	Affected by	Where saved	Default value
Command and query	Recall settings SYSTem:PRESet Instrument reset Power cycle	Save settings	VOLT

Usage

```
:CONFigure:<function>
:CONFigure?
```

`<function>`	The type of measurement: ■ Voltage: `VOLTage[:DC]` ■ Temperature: `TEMPerature`

Details

This command configures the instrument for subsequent measurements on the specified function. The input channel does not change. For example, if `TEMP 2` (Channel 2) is presently selected, sending `CONFigure:VOLT` selects `DCV2` (Channel 2).

This command places the instrument in a one-shot measurement mode. You can then use the `:READ?` command to trigger a measurement and acquire a reading.

When this command is sent, the 2182A is configured as follows:

- The function specified by this command is selected. The input channel remains the same.
- All controls related to the selected function are defaulted to their `*RST` values.
- Continuous initiation is disabled (`:INITiate:CONTinuous OFF`).
- The control source of the trigger model is set to immediate.
- The count values of the trigger model are set to one.
- The delay of the trigger model is set to zero.
- The 2182A is placed in the idle state.
- All math calculations are disabled.
- Buffer operation is disabled. If a storage operation is in process, it is aborted.
- Autozero is set to the `*RST` default value.
- All operations associated with stepping or scanning are disabled.

This command is automatically asserted when the `:MEASure?` command is sent.

Example

This example makes 10 voltage readings with a 5 s delay.

```
CALL SEND (7, ":CONF:VOLT", status%)   ' Perform CONFigure operations.
CALL SEND (7, ":TRIG:DEL 0.5", status%)   ' Set delay for 0.5 s.
CALL SEND (7, ":SAMP:COUN 10", status%)   ' Set sample count to 10.
CALL SEND (7, ":READ?", status%)   ' Trigger and request readings.
reading$ = SPACE$(80)
CALL ENTER(reading$, length%, 7, status%)   ' Address 2182A to talk.
PRINT reading$   ' Display the 10 readings.
```

Also see

:MEASure? (on page 13-4)
:READ? (on page 13-5)

:FETCh?

This command requests the latest reading.

Type	Affected by	Where saved	Default value
Query only	Not applicable	Not applicable	Not applicable

Usage

```
:FETCh?
```

Details

This command requests the latest post-processed reading. After sending this command and addressing the 2182A to talk, the reading is sent to the computer. This command does not affect the instrument setup.

This command does not trigger a measurement. It requests the last available reading and can repeatedly return the same reading until there is a new reading. If your application requires a new reading, use the `:SENSe:DATA:FRESh?` command.

This command is automatically asserted when the `:READ?` or `:MEASure?` command is sent.

If the instrument does not have a reading available (indicated by dashes in the display), sending this command causes a −230, "Data corrupt or stale," error.

If this query is used, the following conditions should be met:

- A reading has been triggered, either by free running (`:INIT:CONT ON` and `:TRIG:SOUR IMM`), by some event such as a bus trigger (`*TRG`), or by an external trigger (`:TRIG:SOUR EXT`).
- It is confirmed that the reading is completed, either by tracking the setting of the `RAV` bit in the status model, or by allowing sufficient time to pass for the reading to complete.

Also see

:INITiate:CONTinuous (on page 13-94)
:MEASure? (on page 13-4)
:READ? (on page 13-5)
:SENSe:DATA:FRESh? (on page 13-54)
:TRIGger[:SEQuence[1]]:SOURce (on page 13-99)

:MEASure?

This command makes measurements, places them in a reading buffer, and returns the last reading.

Type	Affected by	Where saved	Default value
Query only	Not applicable	Not applicable	Not applicable

Usage

```
:MEASure?
:MEASure:<function>?
```

`<function>`	The type of measurement: ■ Voltage: `VOLTage[:DC]` ■ Temperature: `TEMPerature`

Details

This command performs a one-shot measurement and acquires a reading.

When this command is sent, the following commands execute in this order:

- `:ABORt`
- `*RST`
- `:CONFigure:<function>`
- `:TRIGger:SOURce IMMediate`
- `:TRIGger:COUNt 1`
- `:READ?`

When `:ABORt` is executed, the instrument goes into the idle state if continuous initiation is disabled. If continuous initiation is enabled, the operation re-starts at the beginning of the trigger model.

When `:CONFigure` is executed, the instrument goes into one-shot measurement mode.

When `:READ?` is executed, the instrument acquires the reading.

This query is much slower than a `:READ?` or `:FETCh?` query because it reconfigures the instrument each time it is sent.

Also see

:ABORt (on page 13-94)
:CONFigure:<function> (on page 13-1)
:READ? (on page 13-5)
:TRIGger[:SEQuence[1]]:COUNt (on page 13-97)
:TRIGger[:SEQuence[1]]:SOURce (on page 13-99)

:READ?

This command returns a new reading by performing an ABORt, INITiate, and FETCh?.

Type	Affected by	Where saved	Default value
Query only	Not applicable	Not applicable	Not applicable

Usage

```
:READ?
```

Details

This command is typically used with the instrument in the one-shot measurement mode to trigger and acquire a specified number of readings. The readings are stored in the buffer. The :SAMPle:COUNt command specifies the number of readings.

The setting of :FORMat:ELEMents determines which elements of the reading are returned. The default is to return only the reading.

When this command is sent, the following commands execute in this order:

- :ABORt
- :INITiate
- :FETCh?

When :ABORt is executed, the instrument goes into the idle state if continuous initiation is disabled. If continuous initiation is enabled, the operation re-starts at the beginning of the trigger model.

If the instrument is in the idle state, :INITiate takes the instrument out of the idle state. If continuous initiation is enabled, sending this query may cause a −213, Init ignored, error, but it still produces a new reading.

This command is not available when the trigger source is set to BUS or EXTernal. If either of these trigger sources are selected, a −214, "Trigger deadlock," error occurs. Use :FETCh? or :SENSe:DATA:FRESh? instead.

If the 2182A receives a *RST command, it defaults to :INIT:CONT OFF, :TRIG:SOUR IMM, and :TRIG:COUNT 1. Sending a :READ? query under these conditions triggers a new reading.

NOTE

If the sample count is more than one and there are readings stored in the buffer, you cannot use the :READ? command. To use READ?, either set sample count to one or clear the buffer.

Example

This programming example demonstrates a simple method for making and displaying a specified number of readings. The number of readings is specified by the `:SAMPle:COUNt` command. When `:READ?` is asserted, the specified number of readings is made. After all the readings are made, they are sent to the computer and stored in the buffer.

This example makes 10 readings on the `DCV1` function and displays them on the computer.

```
' For QuickBASIC 4.5 and CEC PC488 interface card
' edit the following line to where the QuickBASIC libraries are
' on your computer.
' $INCLUDE: 'c:\qb45\ieeeqb.bi
' Initialize the CEC interface as address 21.
CALL initialize(21, 0)
' Reset controls, clear buffer, and place 2182A in idle.
CALL SEND(7, "*RST", status%)
CALL SEND(7, "TRAC:CLE", status%)
CALL SEND(7, "SAMPLE:COUN 10", status%)
CALL SEND(7, "FORM:ELEM read,unit", status%)
CALL SEND(7, "READ?", status%)
reading$ = SPACE$ (300)
CALL ENTER(reading$, length%, 16, status%)
PRINT reading$
```

Also see

:ABORt (on page 13-94)
:FETCh? (on page 13-3)
:FORMat:ELEMents (on page 13-35)
:INITiate:CONTinuous (on page 13-94)
:INITiate[:IMMediate] (on page 13-95)
:SAMPle:COUNt (on page 13-96)
:SENSe[1]:DATA:FRESh? (on page 13-54)
:TRIGger[:SEQuence[1]]:COUNt (on page 13-97)
:TRIGger[:SEQuence[1]]:SOURce (on page 13-99)

CALCulate[1] subsystem

The commands in this subsystem configure and control the mx+b and percent calculations. Refer to Calculations that you can apply to measurements (on page 4-28) for more information.

:CALCulate[1]:DATA?

This command returns the last calculation result or refreshes the result.

Type	Affected by	Where saved	Default value
Query only	Not applicable	Not applicable	Not applicable

Usage

```
:CALCulate[1]:DATA[:LATest]?
:CALCulate[1]:DATA:FRESh?
```

Details

`FRESh` triggers a reading and returns the resulting calculated result.

When calculations are enabled, the return from `[:SENSe[1]]:DATA:FRESh?` is the reading before calculations are applied and `:CALCulate[1]:DATA:FRESh?` is the reading after calculations are applied. If calculations are disabled, the `[:SENSe[1]]:DATA:FRESh?` and `:CALCulate[1]:DATA:FRESh?` readings are the same.

SCPI compliant.

Also see

[:SENSe[1]]:DATA:FRESh? (on page 13-54)

:CALCulate[1]:FORMat

This command specifies which math operation is performed on measurements when math operations are enabled.

Type	Affected by	Where saved	Default value
Command and query	Recall settings SYSTem:PRESet Instrument reset Power cycle	Save settings	NONE

Usage

```
:CALCulate[1]:FORMat <name>
:CALCulate[1]:FORMat?
```

`<name>`	The math operation: ■ No math operation: `NONE` ■ y = mx+b: `MXB` ■ Percent: `PERCent`

Details

This specifies which math operation is performed on measurements for the selected measurement function.

You can choose one of the following math operations:

- y = mx+b: Manipulate normal display readings by adjusting the m and b factors.
- Percent: Displays measurements as the percentage of deviation from a specified reference constant.

Math calculations are applied to the input signal after relative offset and before limit tests.

The query returns the selected format.

SCPI compliant.

Example

This program fragment shows how to configure and enable the mx+b calculation.

```
CALL SEND(7,":CALC:FORM MXB",status%)   'Selects mx+b calculation.
CALL SEND(7,":CALC:KMAT:MMF 2",status%)   'Sets scale factor (M) to 2.
CALL SEND(7,":CALC:KMAT:MBF 0.5",status%)   'Sets offset (B) to 0.5.
CALL SEND(7,":CALC:KMAT:MUN CD",status%)   'Sets units to CD.
CALL SEND(7,":CALC:STAT ON",status%)   'Enables calculation.
```

Also see

:CALCulate[1]:KMATh:MBFactor (on page 13-8)
:CALCulate[1]:KMATh:MMFactor (on page 13-9)
:CALCulate[1]:KMATh:MUNits (on page 13-10)
:CALCulate[1]:KMATh:PERCent (on page 13-11)

:CALCulate[1]:KMATh:MBFactor

This command specifies the offset, b, for the y = mx + b operation.

Type	Affected by	Where saved	Default value
Command and query	Recall settings SYSTem:PRESet Instrument reset Power cycle	Save settings	0

Usage

```
:CALCulate[1]:KMATh:MBFactor <NRf>
:CALCulate[1]:KMATh:MBFactor?
```

<NRf>	The offset for the y = mx + b operation; the valid range is −100e6 to 100e6

Details

This attribute specifies the offset (b) for an mx + b operation.

The mx + b math operation lets you manipulate normal display readings (x) mathematically based on the calculation:

$$y = mx + b$$

Where:

- *y* is the displayed result
- *m* is a user-defined constant for the scale factor
- *x* is the measurement reading (if you are using a relative offset, this is the measurement with relative offset applied)
- *b* is the user-defined constant for the offset factor

Example

This program fragment shows how to configure and enable the mx+b calculation.

```
CALL SEND(7,":CALC:FORM MXB",status%)   'Selects mx+b calculation.
CALL SEND(7,":CALC:KMAT:MMF 2",status%)   'Sets scale factor (M) to 2.
CALL SEND(7,":CALC:KMAT:MBF 0.5",status%)   'Sets offset (B) to 0.5.
CALL SEND(7,":CALC:KMAT:MUN CD",status%)   'Sets units to CD.
CALL SEND(7,":CALC:STAT ON",status%)   'Enables calculation.
```

Also see

:CALCulate:FORMat (on page 13-7)

:CALCulate[1]:KMATh:MMFactor

This command specifies the scale factor, m, for the y = mx + b math operation.

Type	Affected by	Where saved	Default value
Command and query	Recall settings SYSTem:PRESet Instrument reset Power cycle	Save settings	1

Usage

```
:CALCulate:KMATh:MMFactor <NRf>
:CALCulate:KMATh:MMFactor?
```

`<NRf>`	The scale factor; the valid range is −100e6 to 100e6

Details

This command sets the scale factor (m) for an mx + b operation for the selected measurement function.

The mx + b math operation lets you manipulate normal display readings (x) mathematically according to the following calculation:

$$y = mx + b$$

Where:

- *y* is the displayed result
- *m* is a user-defined constant for the scale factor
- *x* is the measurement reading (if you are using a relative offset, this is the measurement with relative offset applied)
- *b* is the user-defined constant for the offset factor

Example

This program fragment shows how to configure and enable the mx+b calculation.

```
CALL SEND(7,":CALC:FORM MXB",status%)   'Selects mx+b calculation.
CALL SEND(7,":CALC:KMAT:MMF 2",status%)   'Sets scale factor (M) to 2.
CALL SEND(7,":CALC:KMAT:MBF 0.5",status%)   'Sets offset (B) to 0.5.
CALL SEND(7,":CALC:KMAT:MUN CD",status%)   'Sets units to CD.
CALL SEND(7,":CALC:STAT ON",status%)   'Enables calculation.
```

Also see

:CALCulate:FORMat (on page 13-7)

:CALCulate[1]:KMATh:MUNits

This command specifies the scale factor, m, for the y = mx + b math operation.

Type	Affected by	Where saved	Default value
Command and query	Recall settings SYSTem:PRESet Instrument reset Power cycle	Save settings	MX

Usage

```
:CALCulate[1]:KMATh:MUNits "<name>"
:CALCulate[1]:KMATh:MUNits?
```

<name>	Specify units for mx+b; see **Details**

Details

Specify the units for the mx+b result, up to 2 characters. You can specify:

- Alphabetic characters: `A` to `Z`
- Degree symbol (°): `\`
- Ohms symbol (Ω): `[`

Example

`CALCulate:KMATh:MUNits '['` `CALCulate:KMATh:MUNits '\'`	Use ohms symbol (Ω) as units designator. Use degree symbol (°) as units designator.

Also see

:CALCulate:FORMat (on page 13-7)

:CALCulate[1]:KMATh:PERCent

This command specifies the reference constant that is used when math operations are set to percent.

Type	Affected by	Where saved	Default value
Command and query	Recall settings SYSTem:PRESet Instrument reset Power cycle	Save settings	1

Usage

```
:CALCulate[1]:KMATh:PERCent <NRf>
:CALCulate[1]:KMATh:PERCent?
```

<NRf>	The reference value for percent; the valid range is −100e6 to 100e6

Example

This program fragment shows how to configure and enable the percent calculation.

```
CALL SEND(7,":CALC:FORM PERC",status%)    'Selects percent calculation.
CALL SEND(7,":CALC:KMAT:PERC:ACQ",status%)    'Uses input signal as
      'reference.
CALL SEND(7,":CALC:STAT ON",status%)    'Enables calculation.
```

Also see

:CALCulate:FORMat (on page 13-7)

:CALCulate[1]:KMATh:PERCent:ACQuire

This command sets the input signal to be the reference value when math operation is set to percent.

Type	Affected by	Where saved	Default value
Command only	Not applicable	Not applicable	Not applicable

Usage

```
:CALCulate:KMATh:PERCent:ACQuire
```

Example

This program fragment shows how to configure and enable the percent calculation.

```
CALL SEND(7,":CALC:FORM PERC",status%)   'Selects percent calculation.
CALL SEND(7,":CALC:KMAT:PERC:ACQ",status%)   'Uses input signal as
      'reference.
CALL SEND(7,":CALC:STAT ON",status%)   'Enables calculation.
```

Also see

:CALCulate:FORMat (on page 13-7)

:CALCulate[1]:STATe

This command enables or disables the selected math operation.

Type	Affected by	Where saved	Default value
Command and query	Recall settings SYSTem:PRESet Instrument reset Power cycle	Save settings	OFF

Usage

```
:CALCulate:STATe <b>
:CALCulate:STATe?
```

<b>	Disable math operations: OFF or 0 Enable math operations: ON or 1

Example

This program fragment shows how to configure and enable the mx+b calculation.

```
CALL SEND(7,":CALC:FORM MXB",status%)   'Selects mx+b calculation.
CALL SEND(7,":CALC:KMAT:MMF 2",status%)   'Sets scale factor (M) to 2.
CALL SEND(7,":CALC:KMAT:MBF 0.5",status%)   'Sets offset (B) to 0.5.
CALL SEND(7,":CALC:KMAT:MUN CD",status%)   'Sets units to CD.
CALL SEND(7,":CALC:STAT ON",status%)   'Enables calculation.
```

Also see

:CALCulate:FORMat (on page 13-7)

CALCulate2 subsystem

The commands in this path configure and control math calculations on buffer data. For additional information, see Buffer operations (on page 8-1).

:CALCulate2:DATA?

This command reads the result of the last statistic calculation.

Type	Affected by	Where saved	Default value
Query only	Not applicable	Not applicable	Not applicable

Usage

```
:CALCulate2:DATA?
```

Details

The `:CALCulate2:DATA?` command does not initiate a calculate operation. If new data is stored in the buffer, you must send the `:CALCulate2:IMMediate` or `:CALCulate2:IMMediate?` command to recalculate the statistic from that new data.

SCPI compliant.

Also see

:CALCulate2:IMMediate (on page 13-14)

:CALCulate2:FORMat

This command selects the statistics that are applied to the data in the buffer.

Type	Affected by	Where saved	Default value
Command and query	Recall settings SYSTem:PRESet Instrument reset Power cycle	Save settings	NONE

Usage

```
:CALCulate2:FORMat <name>
:CALCulate2:FORMat?
```

`<name>`	The buffer statistic: ■ Lowest reading in buffer: `MINimum` ■ Largest reading in buffer: `MAXimum` ■ Mean value of readings in buffer: `MEAN` ■ Standard deviation of readings in buffer: `SDEViation` ■ No calculation: `NONE`

Example

This program fragment stores 20 readings into the buffer and then calculates the mean average of the buffer readings.

```
' Store Readings:
CALL SEND(7,":TRAC:POIN 20",status%)   'Set buffer size to 20.
CALL SEND(7,":TRAC:FEED SENS",status%)   'Store raw input readings.
CALL SEND(7,":TRAC:FEED:CONT NEXT", status%)   'Start storing readings.
CALL SEND(7,":TRAC:DATA?",status%)   'Request all stored
      'readings.
reading$ = SPACE$(80)
CALL ENTER(reading$, length%, 7, status%)   'Address 2182A to talk.
PRINT reading$   'Display all buffer
      'readings on monitor.
' Calculate mean of buffer readings:
CALL SEND(7,":CALC2:FORM MEAN",status%)   'Select mean calculation.
CALL SEND(7, ":CALC2:STAT ON",status%)   'Enable mean calculation.
CALL SEND(7, ":CALC2:IMM?",status%)   'Perform calculation and
      'request result.
reading$ = SPACE$(80)
CALL ENTER(reading$, length%, 7,status%)   'Address 2182A to talk.
PRINT reading$   'Display all buffer
      'readings on monitor
```

Also see

Buffer statistics (on page 8-3)

:CALCulate2:IMMediate

This command recalculates raw input data in the buffer.

Type	Affected by	Where saved	Default value
Command and query	Not applicable	Not applicable	Not applicable

Usage

```
:CALCulate2:IMMediate
:CALCulate2:IMMediate?
```

Details

The query performs a calculation and reads the result.

After the selected statistic is enabled, `IMMediate` or `IMMediate?` must be sent to calculate the statistic from the data in the buffer.

`CALC2:IMM` and `CALC2:IMM?` are slow to respond when performing the standard deviation calculation on large buffers. In this case, use `*OPC` or `*OPC?`.

SCPI compliant.

Example

This program fragment stores 20 readings into the buffer and then calculates the mean average of the buffer readings.

```
' Store Readings:
CALL SEND(7,":TRAC:POIN 20",status%)   'Set buffer size to 20.
CALL SEND(7,":TRAC:FEED SENS",status%)   'Store raw input readings.
CALL SEND(7,":TRAC:FEED:CONT NEXT", status%)   'Start storing readings.
CALL SEND(7,":TRAC:DATA?",status%)   'Request all stored
      'readings.
reading$ = SPACE$(80)
CALL ENTER(reading$, length%, 7, status%)   'Address 2182A to talk.
PRINT reading$   'Display all buffer
      'readings on monitor.
' Calculate mean of buffer readings:
CALL SEND(7,":CALC2:FORM MEAN",status%)   'Select mean calculation.
CALL SEND(7, ":CALC2:STAT ON",status%)   'Enable mean calculation.
CALL SEND(7, ":CALC2:IMM?",status%)   'Perform calculation and
      'request result.
reading$ = SPACE$(80)
CALL ENTER(reading$, length%, 7,status%)   'Address 2182A to talk.
PRINT reading$   'Display all buffer
      'readings on monitor
```

Also see

:CALCulate:FORMat (on page 13-7)
*OPC (on page 14-7)

:CALCulate2:STATe

This command enables or disables the statistic calculation.

Type	Affected by	Where saved	Default value
Command and query	Recall settings SYSTem:PRESet Instrument reset Power cycle	Save settings	OFF

Usage

```
:CALCulate2:STATe <b>
:CALCulate2:STATe?
```

<b>	Disable math operations: `OFF` or `0` Enable math operations: `ON` or `1`

Details

When enabled, the selected `CALC 2` format is calculated when the `:CALCulate2:IMMediate` or `:CALCulate2:IMMediate?` command is executed.

SCPI compliant.

Example

This program fragment stores 20 readings into the buffer and then calculates the mean average of the buffer readings.

```
' Store Readings:
CALL SEND(7,":TRAC:POIN 20",status%)   'Set buffer size to 20.
CALL SEND(7,":TRAC:FEED SENS",status%)   'Store raw input readings.
CALL SEND(7,":TRAC:FEED:CONT NEXT", status%)   'Start storing readings.
CALL SEND(7,":TRAC:DATA?",status%)   'Request all stored
      'readings.
reading$ = SPACE$(80)
CALL ENTER(reading$, length%, 7, status%)   'Address 2182A to talk.
PRINT reading$   'Display all buffer
      'readings on monitor.
' Calculate mean of buffer readings:
CALL SEND(7,":CALC2:FORM MEAN",status%)   'Select mean calculation.
CALL SEND(7, ":CALC2:STAT ON",status%)   'Enable mean calculation.
CALL SEND(7, ":CALC2:IMM?",status%)   'Perform calculation and
      'request result.
reading$ = SPACE$(80)
CALL ENTER(reading$, length%, 7,status%)   'Address 2182A to talk.
PRINT reading$   'Display all buffer
      'readings on monitor
```

Also see

:CALCulate2:FORMat (on page 13-13)
:CALCulate2:IMMediate (on page 13-14)

CALCulate3 subsystem

The commands in the CALCulate3 subsystem to configure and control limit testing. Refer to Limits (on page 10-1) for more information on limit testing.

:CALCulate3:IMMediate

This command recalculates the limit tests.

Type	Affected by	Where saved	Default value
Command only	Not applicable	Not applicable	Not applicable

Usage

```
:CALCulate3:IMMediate
```

Details

When the instrument is not in a continuous measurement mode (waiting for a trigger), you can change the limits and re-test the last reading. After changing the limits, send `CALCulate3:IMMediate` to perform the limit tests on the last reading.

Sending `IMMediate` does not trigger a reading. It repeats the limit tests on the last reading.

SCPI compliant.

Also see

None

:CALCulate3:LIMit2:CLEar:AUTO

This command enables or disables autoclear for limit 2 when a new trigger model cycle starts.

Type	Affected by	Where saved	Default value
Command and query	Recall settings SYSTem:PRESet Instrument reset Power cycle	Save settings	ON

Usage

```
:CALCulate3:LIMit2:CLEar:AUTO <state>
:CALCulate3:LIMit2:CLEar:AUTO?
```

`<state>`	Disable autoclear: `OFF` Enable autoclear: `ON`

Details

With autoclear enabled, the fail message (`0`) is cleared when the instrument returns to the idle state. If the test is programmed not to return to idle, you can manually clear the fail condition by sending the `CLEar[:IMMediate]` command. If autoclear is disabled, the fail condition must be cleared manually.

SCPI compliant.

Also see

:CALCulate3:LIMit2:CLEar[:IMMediate] (on page 13-17)

:CALCulate3:LIMit2:CLEar[:IMMediate]

This command clears the fail indication for limit 2.

Type	Affected by	Where saved	Default value
Query only	Not applicable	Not applicable	Not applicable

Usage

```
:CALCulate3:LIMit2:CLEar[:IMMediate]
```

Details

Use this command to clear the test results when the limit automatic clear option is turned off. Both the high and low test results are cleared.

SCPI compliant.

Also see

:CALCulate3:LIMit2:CLEar:AUTO (on page 13-17)

:CALCulate3:LIMit2:FAIL?

This command queries the results of a limit test for limit 2.

Type	Affected by	Where saved	Default value
Query only	Not applicable	Not applicable	Not applicable

Usage

```
:CALCulate3:LIMit2:FAIL?
```

Details

Query test result:

- Pass: `1`
- Fail: `0`

The fail message (`0`) for a limit test indicates that the reading is outside the specified limits.

SCPI compliant.

Example

The following program fragment checks the results of a limit 2 test.

```
'Check result of Limit 2 test:
CALL SEND (7, ":calc3:lim2:fail?", status%)    'Request result of
      'Limit 2 test.
reading$ = SPACE$(80)
CALL ENTER(reading$, length%, 7,status%)    'Address 2182A to talk.
PRINT reading$    'Display result of
      'Limit 2 test.
```

Also see

None

:CALCulate3:LIMit2:LOWer[:DATA]

This command specifies the lower limit for a limit test.

Type	Affected by	Where saved	Default value
Command and query	Recall settings SYSTem:PRESet Instrument reset Power cycle	Save settings	−2

Usage

```
:CALCulate3:LIMit2:LOWer[:DATA] <value>
:CALCulate3:LIMit2:LOWer[:DATA]?
```

`<value>`	The value of the low limit for limit 2: −100e6 to 100e6

Details

This command sets the low limit for the selected measurement function for limit 2. When limit 2 testing is enabled, the instrument generates a fail indication when the measurement value is less than this value.

SCPI compliant.

Example

The following program fragment configures the limit tests.

```
' Configure limit tests:
CALL SEND(7,":CALC3:LIM:UPP 0.1",status%)    'Set HI1 limit to 0.1.
CALL SEND(7,":CALC3:LIM:LOW -0.1",status%)    'Set LO1 limit to -0.1.
CALL SEND(7,":CALC3:LIM2:UPP 1",status%)   'Set HI2 limit to 1.
CALL SEND(7,":CALC3:LIM2:LOW -1",status%)   'Set LO2 limit to -1.
CALL SEND(7,":CALC3:LIM:STAT ON",status%)   'Enable Limit 1 test.
CALL SEND(7,":CALC3:LIM2:STAT ON",status%)    'Enable Limit 2 test.
```

Also see

:CALCulate3:LIMit2:STATe (on page 13-19)
:CALCulate3:LIMit2:UPPer (on page 13-20)

:CALCulate3:LIMit2:STATe

This command enables or disables a limit test for limit 2.

Type	Affected by	Where saved	Default value
Command and query	Recall settings SYSTem:PRESet Instrument reset Power cycle	Save settings	OFF

Usage

```
:CALCulate3:LIMit2:STATe <state>
:CALCulate3:LIMit2:STATe?
```

`<state>`	Disable the limit 2 test: `OFF` Enable the limit 2 test: `ON`

Details

SCPI compliant.

Example

The following program fragment configures the limit tests.

```
' Configure limit tests:
CALL SEND(7,":CALC3:LIM:UPP 0.1",status%)    'Set HI1 limit to 0.1.
CALL SEND(7,":CALC3:LIM:LOW -0.1",status%)    'Set LO1 limit to -0.1.
CALL SEND(7,":CALC3:LIM2:UPP 1",status%)   'Set HI2 limit to 1.
CALL SEND(7,":CALC3:LIM2:LOW -1",status%)   'Set LO2 limit to -1.
CALL SEND(7,":CALC3:LIM:STAT ON",status%)   'Enable Limit 1 test.
CALL SEND(7,":CALC3:LIM2:STAT ON",status%)    'Enable Limit 2 test.
```

Also see

:CALCulate3:LIMit2:LOWer (on page 13-18)
:CALCulate3:LIMit2:UPPer (on page 13-20)

:CALCulate3:LIMit2:UPPer[:DATA]

This command specifies the upper limit for a limit test.

Type	Affected by	Where saved	Default value
Command and query	Recall settings SYSTem:PRESet Instrument reset Power cycle	Save settings	2

Usage

```
:CALCulate3:LIMit2:UPPer[:DATA] <value>
:CALCulate3:LIMit2:UPPer[:DATA]?
```

`<value>`	The value of the upper limit for limit 2: −100e6 to 100e6

Details

This command sets the high limit for limit 2 for the selected measurement function. When limit testing is enabled for limit 2, the instrument generates a fail indication when the measurement value is more than this value.

SCPI compliant.

Example

The following program fragment configures the limit tests.

```
' Configure limit tests:
CALL SEND(7,":CALC3:LIM:UPP 0.1",status%)    'Set HI1 limit to 0.1.
CALL SEND(7,":CALC3:LIM:LOW -0.1",status%)    'Set LO1 limit to -0.1.
CALL SEND(7,":CALC3:LIM2:UPP 1",status%)    'Set HI2 limit to 1.
CALL SEND(7,":CALC3:LIM2:LOW -1",status%)    'Set LO2 limit to -1.
CALL SEND(7,":CALC3:LIM:STAT ON",status%)    'Enable Limit 1 test.
CALL SEND(7,":CALC3:LIM2:STAT ON",status%)    'Enable Limit 2 test.
```

Also see

:CALCulate3:LIMit2:LOWer (on page 13-18)
:CALCulate3:LIMit2:STATe (on page 13-19)

:CALCulate3:LIMit[1]:CLEar:AUTO

This command enables or disables autoclear, which clears the limit test results when a new trigger model cycle starts.

Type	Affected by	Where saved	Default value
Command and query	Recall settings Instrument reset Power cycle	Save settings	ON

Usage

```
:CALCulate3:LIMit[1]:CLEar:AUTO <state>
:CALCulate3:LIMit[1]:CLEar:AUTO?
```

`<state>`	Disable autoclear: `OFF` Enable autoclear: `ON`

Details

With autoclear enabled, the fail message (`0`) is cleared when the instrument returns to the idle state. If the test is programmed not to return to idle, you can manually clear the fail condition by sending the `CLEar[:IMMediate]` command. If autoclear is disabled, the fail condition must be cleared manually.

SCPI compliant.

Also see

:CALCulate3:LIMit[1]:CLEar[:IMMediate] (on page 13-21)

:CALCulate3:LIMit[1]:CLEar[:IMMediate]

This command clears the fail indication for LIMIT 1.

Type	Affected by	Where saved	Default value
Query only	Not applicable	Not applicable	Not applicable

Usage

```
:CALCulate3:LIMit[1]:CLEar[:IMMediate]
```

Details

Use this command to clear the test results when the limit autoclear option is turned off. Both the high and low test results are cleared.

SCPI compliant.

Also see

None

:CALCulate3:LIMit[1]:FAIL?

This command queries the results of a limit test.

Type	Affected by	Where saved	Default value
Query only	Not applicable	Not applicable	Not applicable

Usage

```
:CALCulate3:LIMit[1]:FAIL?
```

Details

Query test result:

- Pass: `1`
- Fail: `0`

The fail message (`0`) for a limit test indicates that the reading is outside the specified limits.

SCPI compliant.

Example

The following program fragment checks the results of a limit 1 test.

```
'Check result of Limit 1 test:
CALL SEND(7,":CALC3:LIM:FAIL?",status%)   'Request result of
      'Limit 1 test.
reading$ = SPACE$(80)
CALL ENTER(reading$, length%, 7, status%)   'Address 2182A to talk.
PRINT reading$   'Display result of
      'Limit 1 test.
```

Also see

None

:CALCulate3:LIMit[1]:LOWer[:DATA]

This command specifies the lower limit for a limit test.

Type	Affected by	Where saved	Default value
Command and query	Recall settings Instrument reset Power cycle	Save settings	−1

Usage

```
:CALCulate3:LIMit[1]:LOWer[:DATA] <value>
:CALCulate3:LIMit[1]:LOWer[:DATA]?
```

`<value>`	The value of the lower limit: −100e6 to 100e6

Details

This command sets the low limit for limit 1 for the selected measurement function. When limit testing is enabled, the instrument generates a fail indication when the measurement value is less than this value.

SCPI compliant.

Example

The following program fragment performs limit tests on a voltage input to Channel 1.

```
'Configure 2182A for one-shot DCV1 measurements:
CALL SEND(7, "*RST",status%)   'Put 2182A in one-shot mode.

' Configure limit tests:
CALL SEND(7,":CALC3:LIM:UPP 0.1",status%)   'Set HI1 limit to 0.1.
CALL SEND(7,":CALC3:LIM:LOW -0.1",status%)   'Set LO1 limit to -0.1.
CALL SEND(7,":CALC3:LIM2:UPP 1",status%)   'Set HI2 limit to 1.
CALL SEND(7,":CALC3:LIM2:LOW -1",status%)   'Set LO2 limit to -1.
CALL SEND(7,":CALC3:LIM:STAT ON",status%)   'Enable Limit 1 test.
CALL SEND(7,":CALC3:LIM2:STAT ON",status%)   'Enable Limit 2 test.

'Trigger one reading and display it:
CALL SEND(7,":READ?",status%)   'Trigger and request a
      'reading.
reading$ = SPACE$(80)
CALL ENTER(reading$, length%, 7,status%)   'Address 2182A to talk.
PRINT reading$   'Display reading on computer.

'Check result of Limit 1 test:
CALL SEND(7,":CALC3:LIM:FAIL?",status%)   'Request result of
      'Limit 1 test.
reading$ = SPACE$(80)
CALL ENTER(reading$, length%, 7, status%)   'Address 2182A to talk.
PRINT reading$   'Display result of
      'Limit 1 test.
'Check result of Limit 2 test:
CALL SEND (7, ":CALC3:LIM2:FAIL?", status%)   'Request result of
      'Limit 2 test.
reading$ = SPACE$(80)
CALL ENTER(reading$, length%, 7,status%)   'Address 2182A to talk.
PRINT reading$   'Display result of
      'Limit 2 test.
```

Also see

:CALCulate3:LIMit[1]:STATe (on page 13-24)
:CALCulate3:LIMit[1]:UPPer[:DATA] (on page 13-25)

:CALCulate3:LIMit[1]:STATe

This command enables or disables the limit 1 test.

Type	Affected by	Where saved	Default value
Command and query	Recall settings Instrument reset Power cycle	Save settings	OFF

Usage

```
:CALCulate3:LIMit[1]:STATe <state>
:CALCulate3:LIMit[1]:STATe?
```

`<state>`	Disable the limit 1 test: `OFF` Enable the limit 1 test: `ON`

Details

SCPI compliant.

Example

The following program fragment performs limit tests on a voltage input to Channel 1.

```
'Configure 2182A for one-shot DCV1 measurements:
CALL SEND(7, "*RST",status%)   'Put 2182A in one-shot mode.
' Configure limit tests:
CALL SEND(7,":CALC3:LIM:UPP 0.1",status%)   'Set HI1 limit to 0.1.
CALL SEND(7,":CALC3:LIM:LOW -0.1",status%)   'Set LO1 limit to -0.1.
CALL SEND(7,":CALC3:LIM2:UPP 1",status%)   'Set HI2 limit to 1.
CALL SEND(7,":CALC3:LIM2:LOW -1",status%)   'Set LO2 limit to -1.
CALL SEND(7,":CALC3:LIM:STAT ON",status%)   'Enable Limit 1 test.
CALL SEND(7,":CALC3:LIM2:STAT ON",status%)   'Enable Limit 2 test.

'Trigger one reading and display it:
CALL SEND(7,":READ?",status%)   'Trigger and request a
      'reading.
reading$ = SPACE$(80)
CALL ENTER(reading$, length%, 7,status%)   'Address 2182A to talk.
PRINT reading$   'Display reading on computer.

'Check result of Limit 1 test:
CALL SEND(7,":CALC3:LIM:FAIL?",status%)   'Request result of
      'Limit 1 test.
reading$ = SPACE$(80)
CALL ENTER(reading$, length%, 7, status%)   'Address 2182A to talk.
PRINT reading$   'Display result of
      'Limit 1 test.
'Check result of Limit 2 test:
CALL SEND (7, ":CALC3:LIM2:FAIL?", status%)   'Request result of
      'Limit 2 test.
reading$ = SPACE$(80)
CALL ENTER(reading$, length%, 7,status%)   'Address 2182A to talk.
PRINT reading$   'Display result of
      'Limit 2 test.
```

Also see

:CALCulate3:LIMit:LOWer (on page 13-22)
:CALCulate3:LIMit:UPPer (on page 13-25)

:CALCulate3:LIMit[1]:UPPer[:DATA]

This command specifies the upper limit for a limit test.

Type	Affected by	Where saved	Default value
Command and query	Recall settings Instrument reset Power cycle	Save settings	1

Usage

```
:CALCulate3:LIMit[1]:UPPer[:DATA] <value>
:CALCulate3:LIMit[1]:UPPer[:DATA]?
```

<value>	The value of the upper limit: −100e6 to 100e6

Details

This command sets the high limit for the selected measurement function. When limit testing is enabled, the instrument generates a fail indication when the measurement value is more than this value.

SCPI compliant.

Example

The following program fragment performs limit tests on a voltage input to Channel 1.

```
'Configure 2182A for one-shot DCV1 measurements:
CALL SEND(7, "*RST",status%)   'Put 2182A in one-shot mode.

' Configure limit tests:
CALL SEND(7,":CALC3:LIM:UPP 0.1",status%)   'Set HI1 limit to 0.1.
CALL SEND(7,":CALC3:LIM:LOW -0.1",status%)   'Set LO1 limit to -0.1.
CALL SEND(7,":CALC3:LIM2:UPP 1",status%)   'Set HI2 limit to 1.
CALL SEND(7,":CALC3:LIM2:LOW -1",status%)   'Set LO2 limit to -1.
CALL SEND(7,":CALC3:LIM:STAT ON",status%)   'Enable Limit 1 test.
CALL SEND(7,":CALC3:LIM2:STAT ON",status%)   'Enable Limit 2 test.

'Trigger one reading and display it:
CALL SEND(7,":READ?",status%)   'Trigger and request a
      'reading.
reading$ = SPACE$(80)
CALL ENTER(reading$, length%, 7,status%)   'Address 2182A to talk.
PRINT reading$   'Display reading on computer.

'Check result of Limit 1 test:
CALL SEND(7,":CALC3:LIM:FAIL?",status%)   'Request result of
      'Limit 1 test.
```

```
reading$ = SPACE$(80)
CALL ENTER(reading$, length%, 7, status%)    'Address 2182A to talk.
PRINT reading$    'Display result of
       'Limit 1 test.
'Check result of Limit 2 test:
CALL SEND (7, ":CALC3:LIM2:FAIL?", status%)    'Request result of
       'Limit 2 test.
reading$ = SPACE$(80)
CALL ENTER(reading$, length%, 7,status%)    'Address 2182A to talk.
PRINT reading$    'Display result of
       'Limit 2 test.
```

Also see

:CALCulate3:LIMit[1]:LOWer (on page 13-22)
:CALCulate3:LIMit[1]:STATe (on page 13-24)

CALibration subsystem

The commands in the CALibration subsystem configure and control calibration operations.

:CALibration:UNPRotected:ACALibration:INITiate

This command prepares the 2182A for autocalibration.

Type	Affected by	Where saved	Default value
Command only	Not applicable	Not applicable	Not applicable

Usage

```
:CALibration:UNPRotected:ACALibration:INITiate
```

Example

This program fragment performs low-level autocalibration. After sending the following commands, the `:DONE` and `:INIT:CONT` commands do not execute until calibration is complete.

```
CALL SEND(7,":CAL:UNPR:ACAL:INIT",status%)    ' Prepares 2182A for ACAL.
CALL SEND(7,":CAL:UNPR:ACAL:STEP2",status%)    ' Performs low-level ACAL.
CALL SEND(7,":CAL:UNPR:ACAL:DONE",status%)    ' Exits ACAL mode.
CALL SEND(7, ":INIT:CONT ON", status%)    ' Starts continuous
    ' triggering.
```

Also see

ACAL (calibration) (on page 4-3)

:CALibration:UNPRotected:ACALibration:STEP1

This command performs a full autocalibration.

Type	Affected by	Where saved	Default value
Command only	Not applicable	Not applicable	Not applicable

Usage

```
:CALibration:UNPRotected:ACALibration:STEP1
```

Details

Performs a full autocalibration (100 V and 10 mV).

Also see

ACAL (calibration) (on page 4-3)

:CALibration:UNPRotected:ACALibration:STEP2

This command performs a limited autocalibration.

Type	Affected by	Where saved	Default value
Command only	Not applicable	Not applicable	Not applicable

Usage

```
:CALibration:UNPRotected:ACALibration:STEP2
```

Details

Performs a limited autocalibration (10 mV only).

Example

This program fragment performs low-level autocalibration. After sending the following commands, the `:DONE` and `:INIT` commands do not execute until calibration is complete.

```
CALL SEND(7,":CAL:UNPR:ACAL:INIT",status%)    ' Prepares 2182A for ACAL.
CALL SEND(7,":CAL:UNPR:ACAL:STEP2",status%)    ' Performs low-level ACAL.
CALL SEND(7,":CAL:UNPR:ACAL:DONE",status%)    ' Exits ACAL mode.
CALL SEND(7, ":INIT:CONT ON", status%)    ' Starts continuous
   ' triggering.
```

Also see

ACAL (calibration) (on page 4-3)

:CALibration:UNPRotected:ACALibration:DONE

This command exits autocalibration.

Type	Affected by	Where saved	Default value
Command only	Not applicable	Not applicable	Not applicable

Usage

```
:CALibration:UNPRotected:ACALibration:DONE
```

Details

After sending this command, the 2182A goes into the idle state. You need to initiate trigger commands to begin readings, as shown in the example.

Example

This program fragment performs low-level autocalibration. After sending the following commands, the `:DONE` and `:INIT` commands do not execute until calibration is complete.

```
CALL SEND(7,":CAL:UNPR:ACAL:INIT",status%)    ' Prepares 2182A for ACAL.
CALL SEND(7,":CAL:UNPR:ACAL:STEP2",status%)    ' Performs low-level ACAL.
CALL SEND(7,":CAL:UNPR:ACAL:DONE",status%)    ' Exits ACAL mode.
CALL SEND(7, ":INIT:CONT ON", status%)    ' Starts continuous
   ' triggering.
```

Also see

ACAL (calibration) (on page 4-3)
:INITiate[:IMMediate] (on page 13-95)

:CALibration:UNPRotected:ACALibration:TEMPerature?

This command reads the internal temperature at the time of the last autocalibration.

Type	Affected by	Where saved	Default value
Query only	Not applicable	Not applicable	Not applicable

Usage

```
:CALibration:UNPRotected:ACALibration:TEMPerature?
```

Details

The temperature is returned in °C. If present internal temperature differs from the last autocalibration temperature by more than 1 °C, perform autocalibration.

Also see

ACAL (calibration) (on page 4-3)

DISPlay subsystem

This subsystem contains commands that control the front-panel display.

:DISPlay:ENABle

This command turns the front-panel display on or off.

Type	Affected by	Where saved	Default value
Command and query	Power cycle	Nonvolatile memory	ON

Usage

```
:DISPlay:ENABle <b>
:DISPlay:ENABle?
```

<b>	Disable the front-panel display: `OFF` or `0` Enable the front-panel display: `ON` or `1`

Details

When the front-panel display is disabled, the instrument operates at a higher speed. While disabled, the display is frozen at the last reading and all front-panel controls except the LOCAL key are disabled.

This setting persists through `*RST` and `:SYSTem:PRESet`.

Pressing the LOCAL key or cycling power enables the front-panel display.

SCPI compliant.

Also see

*RST (on page 14-10)
:SYSTem:PRESet (on page 13-73)

:DISPlay[:WINDow[1]]:TEXT:DATA

This command controls user test messages.

Type	Affected by	Where saved	Default value
Command and query	Not applicable	Not applicable	Not applicable

Usage

```
:DISPlay[:WINDow[1]]:TEXT:DATA "<a>"
:DISPlay[:WINDow[1]]:TEXT:DATA?
```

<b>	The ASCII message, up to 12 characters

Details

The characters must be enclosed in either double quotes (`"  "`) or single quotes (`'  '`).

`*RST` and `:SYSTem:PRESet` have no effect on a user-defined message. Pressing the LOCAL key or cycling power cancels all user-defined messages.

SCPI compliant.

Also see

*RST (on page 14-10)
:SYSTem:PRESet (on page 13-88)

:DISPlay[:WINDow[1]]:TEXT:STATe

This command enables or disables the text message mode.

Type	Affected by	Where saved	Default value
Command and query	Not applicable	Not applicable	OFF

Usage

```
:DISPlay[:WINDow[1]]:TEXT:STATe <b>
:DISPlay[:WINDow[1]]:TEXT:STATe?
```

`<b>`	Disable test message mode: `OFF` Enable test message mode: `ON`

Details

This command enables and disables the text message mode. When enabled, a defined message is displayed. When disabled, the message is removed from the display.

A user-defined text message is displayed only as long when the instrument is in remote. Taking the instrument out of remote (by pressing the LOCAL key or sending `GTL`), cancels the message and disables text message mode. Cycling power also disables test message mode.

This setting persists through `*RST` and `:SYSTem:PRESet`.

SCPI compliant.

Also see

*RST (on page 14-10)
:SYSTem:PRESet (on page 13-88)

FORMat subsystem

The commands for this subsystem select the data format that is used to transfer instrument readings over the remote interface.

:FORMat:BORDer

This command sets the byte order for the IEEE Std 754 binary formats.

Type	Affected by	Where saved	Default value
Command and query	Recall settings SYSTem:PRESet Instrument reset Power cycle	Save settings	SWAPped

Usage

```
:FORMat:BORDer <name>
:FORMat:BORDer?
```

`<name>`	The binary byte order: ■ Normal byte order: `NORMal` ■ Reverse byte order for binary formats: `SWAPped`

Details

This attribute selects the byte order in which data is written.

The ASCII data format can only be sent in the normal byte order. If the ASCII format is selected, the `SWAPped` selection is ignored.

When you select `NORMal` byte order, the data format for each element is sent as follows:

`Byte 1 Byte 2 Byte 3 Byte 4`

(Single precision)

`Byte 1    Byte 2` **...** `Byte 8`

(Double precision)

When you select `SWAPped`, the data format for each element is sent as follows:

`Byte 4 Byte 3 Byte 2 Byte 1`

(Single precision)

`Byte 8    Byte 7` **...** `Byte 1`

(Double precision)

The `#0` header is not affected by this command. The header is always sent at the beginning of the data string for each measurement conversion.

This command affects readings that are transferred from the buffer. Commands that are not transferred from the buffer, such as `:SENSe:DATA?` and `:CALC:DATA?`, are always sent in ASCII.

SCPI compliant.

Also see

:FORMat[:DATA] (on page 13-32)

:FORMat[:DATA]

This command specifies the data format that is used when transferring readings over the remote interface.

Type	Affected by	Where saved	Default value
Command and query	Recall settings SYSTem:PRESet Instrument reset Power cycle	Save settings	ASCii

Usage

```
:FORMat[:DATA] <type>
:FORMat[:DATA]?
```

`<type>`	The data format: ■ ASCII: `ASCii` ■ IEEE Std 754 single-precision: `SREal` ■ IEEE Std 754 double-precision: `DREal`

Details

This command affects readings that are transferred from the buffer. Commands that are not transferred from the buffer, such as `:SENSe:DATA?` and `:CALC:DATA?`, are always sent in ASCII.

For every reading conversion, the data string sent over the bus contains the elements specified by the `:ELEMents` command. The specified elements are sent in a particular order.

The ASCII data format is in a direct readable form for the operator. Most BASIC languages easily convert ASCII mantissa and exponent to other formats. However, some speed is compromised to accommodate the conversion.

An example of a reading in ASCII is:

`±1.23456789E±00VDC, 0INTCHAN`

The ASCII data elements in this example are:

- Reading: `±1.23456789E±00`. The reading includes the mantissa `(±1.23456789`) and the exponent (`E±00`). An overflow reading is displayed as +9.9E37 with no units.
- Units: `VDC`. Units can be:
  - DC volts: `VDC`
  - Temperature in °C: `C`
  - Temperature in °F: `F`
  - Temperature in K: `K`

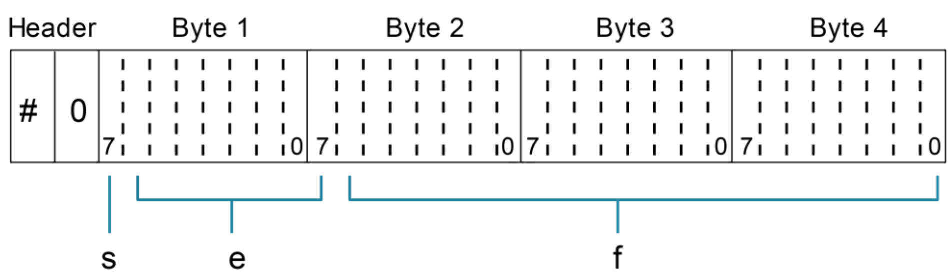
- Channel number: 0. The channel number can be:
  - Internal temperature sensor: 0
  - Channel 1: 1
  - Channel 2: 2
  - External channel number: 1 to 80
- Channel type: INTCHAN. The channel can be:
  - Internal channel: INTCHAN
  - External channel: EXTCHAN

The IEEE Std 754 binary formats use four bytes for single-precision values and eight bytes for double-precision values.

When data is written with any of the binary formats, the response message starts with #0 and ends with a new line. When data is written with the ASCII format, elements are separated with a comma and space.

SREal selects the binary IEEE Std 754 single-precision data format and is shown in the following figure. The figure shows the normal byte order format for each data element. For example, if three valid elements are specified, the data string for each reading conversion is made up of three 32-bit data blocks. The data string for each reading conversion is preceded by a 2-byte header that is the binary equivalent of an ASCII # sign and 0.

Figure 86: IEEE Std 754 single-precision data format (32 data bits)



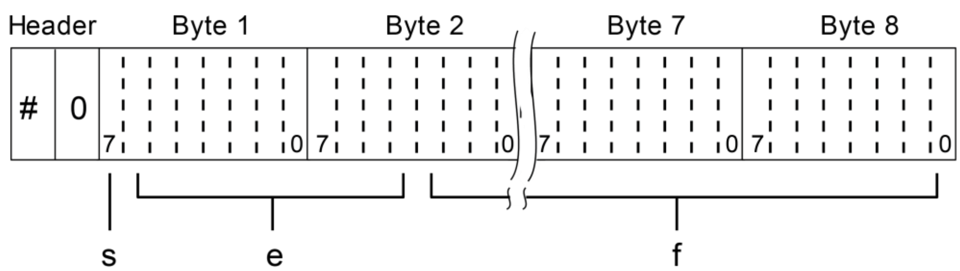
This figure shows:

- s = sign bit (0 = positive, 1 = negative)
- e = exponent bits (8)
- f = fraction bits (23)

The normal byte order is shown. For the swapped byte order, the bytes are sent in the order: Header, Byte 4, Byte 3, Byte 2, Byte 1. The Header is only sent once for each measurement conversion.

DREal selects the binary IEEE Std 754 double-precision data format and is shown in the following figure (normal byte order shown). This format is similar to the single-precision format except that it is 64 bits long.

Figure 87: IEEE Std 754 double precision data format (64 data bits)



This figure shows:

- s = sign bit (0 = positive, 1 = negative)
- e = exponent bits (11)
- f = fraction bits (52)

Bytes 3, 4, 5, and 6 are not shown. Normal byte order shown. For swapped byte order, the bytes are sent in the order Header, Byte 8, Byte 7 .... Byte 1. The Header is only sent once for each measurement conversion.

SCPI compliant.

Example

`FORM REAL`	Set the format to double-precision format.

Also see

:FORMat:ELEMents (on page 13-35)

:FORMat:ELEMents

This command specifies the elements to include in a data string for each measurement.

Type	Affected by	Where saved	Default value
Command and query	Recall settings SYSTem:PRESet Instrument reset Power cycle	Save settings	READing

Usage

```
:FORMat:ELEMents <item list>
:FORMat:ELEMents?
```

<item list>	Data elements: ■ Include the channel number: `CHANnel` ■ Include reading in the data string: `READing` ■ Include the timestamp in the data string: `TSTamp` ■ Include the units: `UNITs` You can specify any combination of these items; each item in the list must be separated by a comma

Details

`READing` includes the instrument reading. The resolution of the reading tracks the display resolution of the instrument. An overflow reading reads as `+9.9e37` with no units.

`CHANnel` correlates the instrument reading to the channel number. Channel 0 corresponds to the sensor used to measure the internal temperature of the 2182A. Channel 1 and Channel 2 correspond to the input channels of the instrument. For external scanning, the number corresponds to the channel number of the switching card.

`UNITs` attaches the function unit to the reading and the channel unit (internal or external) to the channel number. An external channel refers to the channel for an external switch system. This element is not available for the binary formats.

When using this command to add an element, you must include all elements that you want in the format. For example, if the reading is already specified and you want to add the channel, you must include the `READing` parameter:

```
:FORM:ELEM CHAN, READ
```

You can list the data elements in any order.

SCPI compliant.

Example

`FORM REAL`	Set the format to double-precision format.

Also see

:FORMat[:DATA] (on page 13-32)

OUTPut subsystem

These commands control the analog output.

:OUTPut:GAIN

This command specifies the gain factor for the analog output relative offset value.

Type	Affected by	Where saved	Default value
Command and query	Recall settings SYSTem:PRESet Instrument reset Power cycle	Save settings	1

Usage

```
:OUTPut:GAIN <NRf>
:OUTPut:GAIN?
```

<NRf>	The analog output gain factor (M): 1e−6 to 1e6

Details

Gain changes do not take affect until the next reading is triggered.

Example

The following program fragment assumes that you are using analog output to monitor a 1 mV signal on the 10 mV range. Analog output gain is set to 10 to increase sensitivity. Therefore, 1 mV results in a 1 V analog output. Finally, analog output relative offset is enabled to reference the 1 V analog output to zero.

```
CALL SEND (7, ":SYST:PRES", status%)   ' Restore System Preset
   ' defaults.
CALL SEND (7, ":SENS:VOLT:RANG 0.01", status%)   ' Select 10 mV range.
CALL SEND (7, ":OUTP:GAIN 10", status%)   ' Set analog output gain
   ' to 10.
CALL SEND (7, ":OUTP ON", status%)   ' Enable analog output.
CALL SEND (7, ":OUTP:REL ON", status%)   ' Enable analog output
   ' relative offset.
```

Also see

:OUTPut[:STATe] (on page 13-38)

:OUTPut:OFFSet

This command specifies the value for the analog output relative offset.

Type	Affected by	Where saved	Default value
Command and query	Recall settings SYSTem:PRESet Instrument reset Power cycle	Save settings	0

Usage

```
:OUTPut:OFFSet <NRf>
:OUTPut:OFFSet?
```

<NRf>	The relative offset (B): −1.2 to 1.2

Details

Offset changes do not take affect until the next reading is triggered.

Also see

:OUTPut:GAIN (on page 13-36)
:OUTPut:RELative (on page 13-37)

:OUTPut:RELative

This command uses the analog output voltage as the relative offset value.

Type	Affected by	Where saved	Default value
Command and query	Recall settings SYSTem:PRESet Instrument reset Power cycle	Save settings	OFF

Usage

```
:OUTPut:RELative <state>
:OUTPut:RELative?
```

<state>	Use the analog output relative offset value: ON Do not use the analog output relative offset value: OFF

Details

When this command is set to ON, the 2182A uses the analog output voltage as the relative offset value for the analog output.

If relative offset is already enabled, sending this command acquires a new relative offset value.

Example

The following program fragment assumes that you are using analog output to monitor a 1 mV signal on the 10 mV range. Analog output gain is set to 10 to increase sensitivity. Therefore, 1 mV results in a 1 V analog output. Finally, analog output relative offset is enabled to reference the 1 V analog output to zero.

```
CALL SEND (7, ":SYST:PRES", status%)   ' Restore System Preset
   ' defaults.
CALL SEND (7, ":SENS:VOLT:RANG 0.01", status%)   ' Select 10 mV range.
CALL SEND (7, ":OUTP:GAIN 10", status%)   ' Set analog output gain
   ' to 10.
CALL SEND (7, ":OUTP ON", status%)   ' Enable analog output.
CALL SEND (7, ":OUTP:REL ON", status%)   ' Enable analog output
   ' relative offset.
```

Also see

:OUTPut:GAIN (on page 13-36)

:OUTPut[:STATe]

This command enables or disables the analog output.

Type	Affected by	Where saved	Default value
Command and query	Recall settings SYSTem:PRESet Instrument reset Power cycle	Save settings	OFF

Usage

```
:OUTPut[:STATe] <state>
:OUTPut[:STATe]?
```

`<state>`	Enable the analog output: `ON` Disable the analog output: `OFF`

Details

`OFF` forces the analog output to 0 V immediately.

`ON` does not take effect until the next reading is triggered.

Example

The following program fragment assumes that you are using analog output to monitor a 1 mV signal on the 10 mV range. Analog output gain is set to 10 to increase sensitivity. Therefore, 1 mV results in a 1 V analog output. An analog output relative offset is enabled to reference the 1 V analog output to zero.

```
CALL SEND (7, ":SYST:PRES", status%)   ' Restore System Preset
   ' defaults.
CALL SEND (7, ":SENS:VOLT:RANG 0.01", status%)   ' Select 10 mV range.
CALL SEND (7, ":OUTP:GAIN 10", status%)   ' Set analog output gain
   ' to 10.
CALL SEND (7, ":OUTP ON", status%)   ' Enable analog output.
CALL SEND (7, ":OUTP:REL ON", status%)   ' Enable analog output
   ' relative offset.
```

Also see

:OUTPut:GAIN (on page 13-36)
:OUTPut:OFFSet (on page 13-37)

ROUTe subsystem

The ROUTe subsystem contains the commands for stepping and scanning.

:ROUTe:SCAN:INTernal:CCOunt

This command specifies the number of Channel 1 readings.

Type	Affected by	Where saved	Default value
Command and query	Recall settings SYSTem:PRESet Instrument reset Power cycle	Save settings	1

Usage

```
:ROUTe:SCAN:INTernal:CCOunt <n>
:ROUTe:SCAN:INTernal:CCOunt?
```

<n>	The number of readings: 1 to 1023

Example

The following program fragment performs a five measurement internal scan. The readings are stored in the buffer and displayed on the remote computer.

```
CALL SEND(7,"*RST",status%)   'Restore *RST defaults.
CALL SEND(7,":SAMP:COUN 5",status%)   'Set sample count to 5.
CALL SEND(7,":ROUT:SCAN:INT:CCO 4",status%)   'Set channel 1 count to 4.
CALL SEND(7,":ROUT:SCAN:LSEL INT",status%)   'Enable internal scan.
CALL SEND(7,":READ?",status%)   'Trigger scan and request
   'readings.
reading$ = SPACE$ (80)
CALL ENTER (reading$,length%,7,status%)   'Address 2182A to talk.
PRINT reading$   'Display the 5 readings.
```

Also see

Internal stepping and scanning (Channels 1 and 2) (on page 7-2)

:ROUTe:SCAN:LSELect

This command selects the type of scan operation.

Type	Affected by	Where saved	Default value
Command and query	Recall settings SYSTem:PRESet Instrument reset Power cycle	Save settings	NONE

Usage

```
:ROUTe:SCAN:LSELect <name>
:ROUTe:SCAN:LSELect?
```

`<name>`	The scan operation: ■ Disable all scan operations: `NONE` ■ Enable an internal scan: `INTernal` ■ Enable an external scan: `EXTernal`

Example

The following program fragment performs a five measurement internal scan. The readings are stored in the buffer and displayed on the remote computer.

```
CALL SEND(7,"*RST",status%)   'Restore *RST defaults.
CALL SEND(7,":SAMP:COUN 5",status%)   'Set sample count to 5.
CALL SEND(7,":ROUT:SCAN:INT:CCO 4",status%)   'Set channel 1 count to 4.
CALL SEND(7,":ROUT:SCAN:LSEL INT",status%)   'Enable internal scan.
CALL SEND(7,":READ?",status%)   'Trigger scan and request
   'readings.
reading$ = SPACE$ (80)
CALL ENTER (reading$,length%,7,status%)   'Address 2182A to talk.
PRINT reading$   'Display the 5 readings.
```

Also see

Stepping and scanning (on page 7-1)
TRIGger subsystem (on page 13-94)

:ROUTe:SCAN[:EXTernal]

This command specifies an external scan list.

Type	Affected by	Where saved	Default value
Command	Recall settings SYSTem:PRESet Instrument reset Power cycle	Save settings	@1:10

Usage

`:ROUTe:SCAN[:EXTernal] <list>`

`<list>`	The external scan list: 2 to 800; see **Details**

Details

The `<list>` parameter is formatted as follows:

`<list> = (@X:Y)`

Where:

- *X* is the minimum channel
- *Y* is the maximum channel

Example

`:ROUT:SCAN (@1:5)`	Specifies a scan list of channels 1 to 5.

Also see

External stepping and scanning (channel 1) (on page 7-7)

SENSe[1] subsystem

The SENSe[1] subsystem commands configure and control the measurement functions of the instrument.

:SENSe[1]:<function>:DIGits

This command determines the number of digits that are displayed for measurements on the front panel.

Type	Affected by	Where saved	Default value
Command and query	Recall settings SYSTem:PRESet Instrument reset Power cycle	Save settings	Voltage: 8 Temperature: 6

Usage

```
:SENSe[1]:<function>:DIGits <value>
:SENSe[1]:<function>:DIGits <DEF|MIN|MAX>
:SENSe[1]:<function>:DIGits?
:SENSe[1]:<function>:DIGits? <DEF|MIN|MAX>
```

`<function>`	The function the digits setting applies to: ■ Voltage (DCV1 and DCV2): `VOLTage[:DC]` ■ Temperature (TEMP 1 and TEMP 2): `TEMPerature`
`<value>`	Display digits: ■ 7½ (voltage only): `8` ■ 6½: `7` ■ 5½: `6` ■ 4½: `5` ■ 3½: `4`
`<DEF|MIN|MAX>`	The `DEFault`, `MINimum`, or `MAXimum` value

Details

This command affects how the reading for a measurement is displayed on the front panel of the instrument. It does not affect the number of digits returned in a remote command reading. It also does not affect the accuracy or speed of measurements.

The display digits setting is saved with the function setting, so if you use another function, then return to the function for which you set display digits, the display digits setting you set previously is retained.

The change in digits occurs the next time a measurement is made.

Example

`:SENS:VOLT:DIG 4` `:SENS:TEMP:DIG 5`	Select 3½-digit resolution for voltage readings. Select 5½-digit resolution for temperature readings.

Also see

None

:SENSe[1]:<function>:APERture

This command determines the aperture setting.

Type	Affected by	Where saved	Default value
Command and query	Recall settings SYSTem:PRESet Instrument reset Power cycle	Save settings	83.33 ms

Usage

```
:SENSe[1]:<function>:APERture <n>
:SENSe[1]:<function>:APERture?
```

`<function>`	Select the function to which the setting applies: ■ Voltage: `VOLTage[:DC]` ■ Temperature: `TEMPerature`
`<n>`	Specify integration rate in seconds: ■ Range for 60 Hz: 0.16667 ms to 1000 ms ■ Range for 50 Hz: 0.2000 ms to 1000 ms

Details

NOTE

The measurement time can also be set as NPLCs. Changing the NPLC value changes the aperture time and changing the aperture time changes the NPLC value.

Also see

:SENSe[1]:<function>:NPLCycles (on page 13-43)
Rate (on page 4-19)

:SENSe[1]:<function>:NPLCycles

This command sets the time that the input signal is measured.

Type	Affected by	Where saved	Default value
Command and query	Recall settings SYSTem:PRESet Instrument reset Power cycle	Save settings	5

Usage

```
:SENSe[1]:<function>:NPLCycles <n>
:SENSe[1]:<function>:NPLCycles?
```

`<function>`	Select the function to which the setting applies: ■ Voltage: `VOLTage[:DC]` ■ Temperature: `TEMPerature`
`<n>`	The number of power-line cycles for each measurement: ■ Range for 60 Hz: `0.01` to `60` ■ Range for 50 Hz: `0.01` to `50`

Details

SCPI compliant.

NOTE

The measurement time can also be set as an aperture time. Changing the NPLC value changes the aperture time and changing the aperture time changes the NPLC value.

Example

The following program fragment sets the voltage reading rate to 2 PLC and the temperature reading rate to 5 PLC.

```
CALL SEND(7,":SENS:VOLT:NPLC 2",status%)    'Set volts for 2 PLC.
CALL SEND(7,":SENS:TEMP:NPLC 5",status%)    'Set temperature for 5 PLC.
```

Also see

:SENSe[1]:<function>:APERture (on page 13-42)
Rate (on page 4-19)

:SENSe[1]:<function>[:CHANnel<X>]:DFILter:COUNt

This command set the filter count for measurements when using the selected function.

Type	Affected by	Where saved	Default value
Command and query	Recall settings SYSTem:PRESet Instrument reset Power cycle	Save settings	10

Usage

```
:SENSe[1]:<function>[:CHANnel1]:DFILter:COUNt <n>
:SENSe[1]:<function>:CHANnel2:DFILter:COUNt <n>
:SENSe[1]:<function>[:CHANnel1]:DFILter:COUNt?
:SENSe[1]:<function>:CHANnel2:DFILter:COUNt?
```

`<function>`	Select the function to which the setting applies: ■ Voltage: `VOLTage[:DC]` ■ Temperature: `TEMPerature`
`<n>`	The filter count: `0` to `100`

Details

The filter count specifies how many consecutive A/D conversions in the filter window to place in the memory stack. When the stack is full, the A/D conversions are averaged to calculate the final filtered reading. The filter count can be set from 1 to 100. If the filter count is 1, no averaging is done. Only readings in the filter window are displayed, stored, or transmitted.

Example

The following program fragment configures the filter for Channel 2 voltage (DCV2). It disables the analog filter and enables the digital filter (5% window, count 10, moving).

```
' Analog filter:
CALL SEND(7,":SENS:VOLT:CHAN2:LPAS OFF",status%)   'Disable analog filter.

' Digital filter:
CALL SEND(7,":SENS:VOLT:CHAN2:DFIL:WIND 5",status%)   'Set window to 5%.
CALL SEND(7,":SENS:VOLT:CHAN2:DFIL:COUN 10",status%)   'Set count to 10.
CALL SEND(7,":SENS:VOLT:CHAN2:DFIL:TCON MOV",status%)   'Select moving filter.
CALL SEND(7,":SENS:VOLT:CHAN2:DFIL:STAT ON",status%)   'Enable digital filter.
```

Also see

Digital filter (on page 4-21)
:SENSe[1]:<function>[:CHANnel<X>]:DFILter[:STATe] (on page 13-48)
:SENSe[1]:<function>[:CHANnel<X>]:DFILter:WINDow (on page 13-47)

:SENSe[1]:<function>[:CHANnel<X>]:DFILter:TCONtrol

This command sets the filter type to moving or repeating average.

Type	Affected by	Where saved	Default value
Command and query	Recall settings SYSTem:PRESet Instrument reset Power cycle	Save settings	MOVing

Usage

```
:SENSe:<function>[:CHANnel1]:DFILter:TCONtrol <type>
:SENSe:<function>:CHANnel2:DFILter:TCONtrol <type>
:SENSe:<function>[:CHANnel1]:DFILter:TCONtrol?
:SENSe:<function>[:CHANnel2]:DFILter:TCONtrol?
```

`<function>`	Select the function to which the setting applies: ■ Voltage: `VOLTage[:DC]` ■ Temperature: `TEMPerature`
`<type>`	The type of averaging: ■ Moving: `MOVing` ■ Repeating: `REPeat`

Details

When the moving average filter is selected, the measurements are added to the stack continuously on a first-in, first-out basis. As each measurement is made, the oldest measurement is removed from the stack. A new averaged sample is produced using the new measurement and the data that is now in the stack.

When the moving average filter is first selected, the stack is empty. When the first measurement is made, it is copied into all the stack locations to fill the stack. A true average is not produced until the stack is filled with new measurements.

When the repeating average filter is selected, a set of measurements are made. These measurements are stored in a measurement stack and averaged together to produce the averaged sample. Once the averaged sample is produced, the stack is flushed, and the next set of data is used to produce the next averaged sample. This type of filter is the slowest, since the stack must be completely filled before an averaged sample can be produced, but it provides more stable results.

For either method, the greater the number of measurements that are averaged, the slower the averaged sample rate, but the lower the noise error. Trade-offs between speed and noise are normally required to tailor the instrumentation to your measurement application.

Example

The following program fragment configures the filter for Channel 2 voltage (DCV2). It disables the analog filter and enables the digital filter (5% window, count 10, moving).

```
' Analog filter:
CALL SEND(7,":SENS:VOLT:CHAN2:LPAS OFF",status%)   'Disable analog filter.

' Digital filter:
CALL SEND(7,":SENS:VOLT:CHAN2:DFIL:WIND 5",status%)   'Set window to 5%.
CALL SEND(7,":SENS:VOLT:CHAN2:DFIL:COUN 10",status%)   'Set count to 10.
CALL SEND(7,":SENS:VOLT:CHAN2:DFIL:TCON MOV",status%)   'Select moving filter.
CALL SEND(7,":SENS:VOLT:CHAN2:DFIL:STAT ON",status%)   'Enable digital filter.
```

Also see

Digital filter (on page 4-21)
:SENSe:<function>[:CHANnel<X>]:DFILter[:STATe] (on page 13-48)

:SENSe[1]:<function>[:CHANnel<X>]:DFILter:WINDow

This command enables or disables the digital filter for measurements when using the selected function.

Type	Affected by	Where saved	Default value
Command and query	Recall settings SYSTem:PRESet Instrument reset Power cycle	Save settings	0.01

Usage

```
:SENSe[1]:<function>[:CHANnel1]:DFILter:WINDow <n>
:SENSe[1]:<function>:CHANnel2:DFILter:WINDow <n>
:SENSe[1]:<function>[:CHANnel1]:DFILter:WINDow?
:SENSe[1]:<function>:CHANnel2:DFILter:WINDow?
```

`<function>`	Select the function to which the setting applies: ■ Voltage: `VOLTage[:DC]` ■ Temperature: `TEMPerature`
`<n>`	The filter window in percent: 0 to 10

Details

The digital filter uses a window to control the filter threshold. While the input signal remains in the selected window, A/D conversions continue to be placed in the stack. If the signal changes to a value outside the window, the filter resets, and the filter starts processing again, starting with a new initial conversion value from the A/D converter.

Example

The following program fragment configures the filter for Channel 2 voltage (DCV2). It disables the analog filter and enables the digital filter (5% window, count 10, moving).

```
' Analog filter:
CALL SEND(7,":SENS:VOLT:CHAN2:LPAS OFF",status%)   'Disable analog filter.

' Digital filter:
CALL SEND(7,":SENS:VOLT:CHAN2:DFIL:WIND 5",status%)   'Set window to 5%.
CALL SEND(7,":SENS:VOLT:CHAN2:DFIL:COUN 10",status%)   'Set count to 10.
CALL SEND(7,":SENS:VOLT:CHAN2:DFIL:TCON MOV",status%)   'Select moving filter.
CALL SEND(7,":SENS:VOLT:CHAN2:DFIL:STAT ON",status%)   'Enable digital filter.
```

Also see

Digital filter (on page 4-21)
:SENSe:<function>[:CHANnel<X>]:DFILter[:STATe] (on page 13-48)

:SENSe[1]:<function>[:CHANnel<X>]:DFILter[:STATe]

This command enables or disables the digital filter for measurements when using the selected function.

Type	Affected by	Where saved	Default value
Command and query	Recall settings SYSTem:PRESet Instrument reset Power cycle	Save settings	ON

Usage

```
:SENSe[1]:<function>[:CHANnel1]:DFILter[:STATe] <state>
:SENSe[1]:<function>:CHANnel2:DFILter[:STATe] <state>
:SENSe[1]:<function>[:CHANnel1]:DFILter[:STATe]?
:SENSe[1]:<function>:CHANnel2:DFILter[:STATe]?
```

`<function>`	Select the function to which the setting applies: ■ Voltage: `VOLTage[:DC]` ■ Temperature: `TEMPerature`
`<state>`	The filter status; set to one of the following values: ■ Disable the analog filter: `OFF` ■ Enable the analog filter: `ON`

Details

The digital filter stabilizes noisy measurements. The displayed, stored, or transmitted reading is a windowed-average of a number of reading conversions (from 1 to 100).

In general, the digital filter places a specified number of consecutive A/D conversions (Filter Count) into a memory stack. The readings in the stack are then averaged to yield a single filtered reading. The filter can be set to use a moving or repeating average.

Example

The following program fragment configures the filter for Channel 2 voltage (DCV2). It disables the analog filter and enables the digital filter (5% window, count 10, moving).

```
' Analog filter:
CALL SEND(7,":SENS:VOLT:CHAN2:LPAS OFF",status%)   'Disable analog filter.

' Digital filter:
CALL SEND(7,":SENS:VOLT:CHAN2:DFIL:WIND 5",status%)   'Set window to 5%.
CALL SEND(7,":SENS:VOLT:CHAN2:DFIL:COUN 10",status%)   'Set count to 10.
CALL SEND(7,":SENS:VOLT:CHAN2:DFIL:TCON MOV",status%)   'Select moving filter.
CALL SEND(7,":SENS:VOLT:CHAN2:DFIL:STAT ON",status%)   'Enable digital filter.
```

Also see

Digital filter (on page 4-21)
:SENSe:<function>[:CHANnel<X>]:DFILter:COUNt (on page 13-45)
:SENSe:<function>[:CHANnel<X>]:DFILter:TCONtrol (on page 13-46)
:SENSe:<function>[:CHANnel<X>]:DFILter:WINDow (on page 13-47)

:SENSe[1]:<function>[:CHANnel<X>]:LPASs[:STATe]

This command enables or disables the analog filter for measurements for the selected function.

Type	Affected by	Where saved	Default value
Command and query	Recall settings SYSTem:PRESet Instrument reset Power cycle	Save settings	OFF

Usage

```
:SENSe[1]:<function>[:CHANnel1]:LPASs[:STATe] <state>
:SENSe[1]:<function>:CHANnel2:LPASs[:STATe] <state>
:SENSe[1]:<function>[:CHANnel1]:LPASs[:STATe]?
:SENSe[1]:<function>:CHANnel2:LPASs[:STATe]?
```

`<function>`	Select the function to which the setting applies: ■ Voltage: `VOLTage` ■ Temperature: `TEMPerature`
`<state>`	The filter status; set to one of the following values: ■ Disable the analog filter: `OFF` ■ Enable the analog filter: `ON`

Details

When you enable the low-pass analog filter, the normal-mode noise rejection ratio of the instrument is increased at 60 Hz. This filters out noise induced by the power line. The analog filter attenuates frequency at 20 dB/decade starting at 18 Hz.

A primary use of the analog filter is to keep the high-gain input stage of the 2182A from saturating due to the presence of high ac and dc voltage.

NOTE

The filter only attenuates ac voltages for the 10 mV range of the 2182A.

The analog filter adds approximately 125 ms of settling time between A/D conversions. The additional settling time may be required when using a high-impedance (≥100 kΩ) source in the test circuit. The increased settling time causes the reading rate of the 2182A to be greatly reduced. Therefore, if the analog filter is not needed, turn it off.

Example

This example produces a one-shot reading of dc volts with no trigger at the fastest rate.

```
*RST
:INITiate:CONTinuous OFF;:ABORt
:SENSe:FUNCtion 'VOLTage:DC'
:SENSe:VOLTage:DC:RANGe 10   // Use fixed range for fastest readings.
:SENSe:VOLTage:DC:NPLC 0.01   // Use lowest NPLC setting for fastest readings.
:DISPlay:ENABle OFF   // Turn off display to increase speed.
:SYSTem:AZERo:STATe OFF   // Disable autozero to increase speed, but may cause
   // drift over time.
:SENSe:VOLTage:DC:LPAS OFF   // Turn off analog filter for speed.
:SENSe:VOLTage:DC:DFIL OFF   // Turn off digital filter for speed.
:TRIGger:COUNt 1
:READ?
```

Also see

Analog filter (on page 4-21)

:SENSe[1]:<function>[:CHANnel<X>]:REFerence

This command contains the relative offset value.

Type	Affected by	Where saved	Default value
Command and query	Recall settings SYSTem:PRESet Instrument reset Power cycle	Save settings	0

Usage

```
:SENSe[1]:<function>[:CHANnel1]:REFerence <n>
:SENSe[1]:<function>:CHANnel2:REFerence <n>
:SENSe[1]:<function>[:CHANnel1]:REFerence?
:SENSe[1]:<function>:CHANnel2:REFerence?
```

`<function>`	Select the function to which the setting applies: ■ Voltage: `VOLTage[:DC]` ■ Temperature: `TEMPerature`
`<n>`	The relative offset value: ■ Voltage, channel 1: `-120` to `120` ■ Voltage, channel 2: `-12` to `12` ■ Temperature (°C): `-273` to `1800`

Details

When making measurements, you may need to subtract an offset value from a measurement.

The relative offset feature subtracts a set value or a baseline reading from measurement readings. When you enable relative offset, all measurements are recorded as the difference between the actual measured value and the relative offset value. The formula to calculate the offset value is:

Displayed value = Actual measured value − Relative offset value

When `:SENSe[1]:<function>[:CHANnel<X>]:REFerence:ACQuire` is sent, this is changed to the acquired value.

Voltage commands are SCPI compliant.

Example

This program fragment shows how to set a +1 V baseline for the DCV1 function. For this baseline value, a +1 V input is displayed as 0 V.

```
CALL SEND(7,":SYST:PRES",status%)    'Selects DCV function and
   'enables autorange.
CALL SEND(7,":SENS:VOLT:REF 1",status%)    'Sets a 1 V relative offset value.
CALL SEND(7,":SENS:VOLT:REF:STAT ON",status%)    'Enables relative offset for DCV1.
```

Also see

Relative offset (on page 4-25)
:SENSe[1]:<function>[:CHANnel<X>]:REFerence:ACQuire (on page 13-51)
:SENSe[1]:<function>[:CHANnel<X>]:REFerence:STATe (on page 13-52)

:SENSe[1]:<function>[:CHANnel<X>]:REFerence:ACQuire

This command acquires a measurement and stores it as the relative offset value.

Type	Affected by	Where saved	Default value
Command only	Not applicable	Not applicable	Not applicable

Usage

```
:SENSe[1]:<function>[:CHANnel1]:REFerence:ACQuire
:SENSe[1]:<function>:CHANnel2:REFerence:ACQuire
```

`<function>`	Select the function to which the setting applies: ■ Voltage: `VOLTage[:DC]` ■ Temperature: `TEMPerature`

Details

Uses the input signal on the selected channel as the relative offset value.

Example

This program fragment shows how to null out zero offset for the DCV1 function. Be sure to short the Channel 1 input.

```
CALL SEND(7,":SYST:PRES",status%)   'Selects DCV1 and enables
   'autorange.
CALL SEND(7,":FETCH?",status%)   'Fetches Channel 1 offset.
reading$ = SPACE$(80)
CALL ENTER(reading$,length%,7,status%)   'Gets offset reading.
CALL SEND(7,":SENS:VOLT:REF:ACQ",status%)   'Acquires relative offset value.
CALL SEND(7,":SENS:VOLT:REF:STAT ON",status%)   'Enables relative offset for DCV1.
```

Also see

Relative offset (on page 4-25)
:SENSe:<function>[:CHANnel<X>]:REFerence (on page 13-50)

:SENSe[1]:<function>[:CHANnel<X>]:REFerence:STATe

This command enables or disables the application of a relative offset value to the measurement.

Type	Affected by	Where saved	Default value
Command and query	Recall settings SYSTem:PRESet Instrument reset Power cycle	Save settings	Channel 1: OFF Channel 2, voltage: ON Channel 2, temperature: OFF

Usage

```
:SENSe[1]:<function>[:CHANnel1]:REFerence:STATe <state>
:SENSe[1]:<function>:CHANnel2:REFerence:STATe <state>
:SENSe[1]:<function>[:CHANnel1]:REFerence:STATe?
:SENSe[1]:<function>:CHANnel2:REFerence:STATe?
```

`<function>`	Select the function to which the setting applies: ■ Voltage: `VOLTage[:DC]` ■ Temperature: `TEMPerature`
`<X>`	The channel to which the setting applies, `1` or `2`
`<state>`	Disable the relative offset: `OFF` Enable the relative offset: `ON`

Details

When relative measurements are enabled, all subsequent measured readings are offset by the relative offset value. You can enter a relative offset value or have the instrument acquire a relative offset value.

Each returned measured relative reading is the result of the following calculation:

Displayed reading = Actual measured reading - Relative offset value

The voltage commands are SCPI compliant.

Example

This program fragment shows how to set a +1 V baseline for the DCV1 function. For this baseline value, a +1 V input is displayed as 0 V.

```
CALL SEND(7,":SYST:PRES",status%)    'Selects DCV function and
   'enables autorange.
CALL SEND(7,":SENS:VOLT:REF 1",status%)    'Sets a 1 V relative offset value.
CALL SEND(7,":SENS:VOLT:REF:STAT ON",status%)    'Enables relative offset for DCV1.
```

Also see

Relative offset (on page 4-25)
:SENSe:<function>[:CHANnel<X>]:REFerence (on page 13-50)

:SENSe[1]:CHANnel

This command selects the measurement channel.

Type	Affected by	Where saved	Default value
Command and query	Recall settings SYSTem:PRESet Instrument reset Power cycle	Save settings	1

Usage

```
:SENSe[1]:CHANnel "<channel>"
:SENSe[1]:CHANnel?
```

`<channel>`	Select measurement channel: ■ Internal temperature sensor (see **Details**): `0` ■ Channel 1: `1` ■ Channel 2: `2`

Details

Channel 0 is the internal temperature sensor. When a temperature function is selected, reading Channel 0 returns the internal temperature reading. When a voltage function is selected, reading Channel 0 returns the voltage reading of the internal temperature sensor.

Example

The following program fragment switches to DCV1 (channel 1 voltage) and makes a reading.

```
CALL SEND(7, "sens:func 'volt'", status%)
   CALL SEND(7, "sens:chan 1; :read?", status%)
   reading%$ = SPACE$(80)
   CALL ENTER(reading$, length%, 7, status%
   PRINT reading$
```

Also see

None

:SENSe[1]:DATA:FRESh?

This command triggers a measurement and returns the 2182A reading.

Type	Affected by	Where saved	Default value
Query only	Not applicable	Not applicable	Not applicable

Usage

```
:SENSe[1]:DATA:FRESh?
```

Details

This command returns a new reading. It waits for a reading to complete if a reading is in progress.

The `:CALC:DATA:FRESh?` query is similar to the `:SENSe:DATA:FRESh?` query, but applies to readings that have math applied to them, such as mx+b scaling.

SCPI compliant.

Also see

Calculations that you can apply to measurements (on page 4-28)
:CALCulate:DATA (on page 13-7)

:SENSe[1]:DATA[:LATest]?

This command returns the last 2182A reading.

Type	Affected by	Where saved	Default value
Query only	Not applicable	Not applicable	Not applicable

Usage

```
:SENSe[1]:DATA[:LATest]?
```

Details

This command can return data that was captured before changes to the measurement settings take place, which would invalidate the old reading.

The `:CALC:DATA:LATest?` query is similar to the `:DATA:LAT?` query, but applies to readings that have math, such as mx+b, applied to them.

SCPI compliant.

Also see

:CALCulate:DATA? (on page 13-7)

:SENSe[1]:FUNCtion

This command selects the active measure function, voltage or temperature.

Type	Affected by	Where saved	Default value
Command and query	Recall settings SYSTem:PRESet Instrument reset Power cycle	Save settings	VOLT

Usage

```
:SENSe[1]:FUNCtion "<function>"
:SENSe[1]:FUNCtion?
```

`<function>`	A string that contains the measure function: `VOLTage` or `TEMPerature`

Details

SCPI compliant.

Example

`SENS:FUNC "TEMP"`	Make the temperature function the active function.

Also see

None

:SENSe[1]:HOLD:COUNt

This command selects the reading hold count.

Type	Affected by	Where saved	Default value
Command and query	Recall settings SYSTem:PRESet Instrument reset Power cycle	Save settings	5

Usage

```
:SENSe:HOLD:COUNt <n>
:SENSe:HOLD:COUNt?
```

`<n>`	The hold count: `2` to `100`

Also see

Reading hold (on page 9-7)

:SENSe[1]:HOLD:STATe

This command selects the reading hold state.

Type	Affected by	Where saved	Default value
Command and query	Recall settings SYSTem:PRESet Instrument reset Power cycle	Save settings	OFF

Usage

```
:SENSe[1]:HOLD:STATe <b>
:SENSe[1]:HOLD:STATe?
```

<b>	Disable reading hold: OFF Enable reading hold: ON

Details

This is set to OFF when :SENSe[1]:VOLTage[:DC]:RATio is set to ON.

Also see

Reading hold (on page 9-7)
:SENSe[1]:VOLTage[:DC]:RATio (on page 13-63)

:SENSe[1]:HOLD:WINDow

This command selects the reading hold window.

Type	Affected by	Where saved	Default value
Command and query	Recall settings SYSTem:PRESet Instrument reset Power cycle	Save settings	1

Usage

```
:SENSe[1]:HOLD:WINDow <n>
:SENSe[1]:HOLD:WINDow?
```

<n>	The hold window (%): 0.01 to 20

Also see

Reading hold (on page 9-7)

:SENSe[1]:TEMPerature:RJUNction:RSELect

This command defines the type of the thermocouple reference junction.

Type	Affected by	Where saved	Default value
Command and query	Recall settings SYSTem:PRESet Instrument reset Power cycle	Save settings	INTernal

Usage

```
:SENSe[1]:TEMPerature:RJUNction:RSELect <type>
:SENSe[1]:TEMPerature:RJUNction:RSELect?
```

`<type>`	The type of reference junction: ■ `SIMulated` ■ `INTernal`

Example

The following program fragments measure voltage on Channel 1 and temperature on Channel 2. Temperature is configured using a simulated reference junction (typically an ice bath) and a type K thermocouple.

```
' Configure Temperature:
CALL SEND(7,":SENS:TEMP:TRANS TC",status%)   'Select thermocouple
      'sensor.
CALL SEND(7,":SENS:TEMP:RJUN:RSEL SIM",status%)   'Select simulated reference.
CALL SEND(7,":SENS:TEMP:RJUN:RSEL:SIM 0",status%)   'Set reference to 0°C.
CALL SEND(7,":SENS:TEMP:TC K",status%)   'Set for type K
      thermocouple.
CALL SEND(7,":UNIT:TEMP F",status%)   'Read in °F.

' Measure voltage on Channel 1:
CALL SEND(7,":SENS:CHAN 1",status%)   'Select Channel 1.
CALL SEND(7,":SENS:FUNC 'VOLT'",status%)   'Select DCV1.
CALL SEND(7,":SENS:DATA:FRES?",status%)   'Request a fresh
      'reading.
reading$ = SPACE$(80)
CALL ENTER(reading$, length%, 7, status%)   'Address 2182A to talk.
PRINT reading$   'Display reading on computer.

' Measure temperature on Channel 2:
CALL SEND(7,":SENS:CHAN 2",status%)   'Select Channel 2.
CALL SEND(7,":SENS:FUNC 'TEMP'",status%)   'Select TEMP 2.
CALL SEND(7,":SENS:DATA:FRES?",status%)   'Request a fresh
      'reading.
reading$ = SPACE$(80)
CALL ENTER(reading$, length%, 7,status%)   'Address 2182A to talk.
PRINT reading$   'Display reading on computer.
```

Also see

:SENSe[1]:TEMPerature:RJUNction:SIMulated (on page 13-58)

:SENSe[1]:TEMPerature:RJUNction:SIMulated

This command sets the simulated reference temperature of the thermocouple reference junction.

Type	Affected by	Where saved	Default value
Command and query	Recall settings SYSTem:PRESet Instrument reset Power cycle	Save settings	23

Usage

```
:SENSe[1]:TEMPerature:RJUNction:SIMulated <tempValue>
:SENSe[1]:TEMPerature:RJUNction:SIMulated?
```

`<tempValue>`	The temperature in Celsius: `0` to `60`

Details

This command applies to the temperature function when the reference junction is set to simulated. It allows you to set the simulated reference temperature value.

Also see

:SENSe:TEMPerature:RJUNction:RSELect (on page 13-57)

:SENSe[1]:TEMPerature:RTEMperature?

This command queries the internal temperature in °C.

Type	Affected by	Where saved	Default value
Query only	Not applicable	Not applicable	Not applicable

Usage

```
:SENSe[1]:TEMPerature:RTEMperature?
```

Also see

None

:SENSe[1]:TEMPerature:TCouple[:TYPE]

This command indicates the thermocouple type.

Type	Affected by	Where saved	Default value
Command and query	Recall settings SYSTem:PRESet Instrument reset Power cycle	Save settings	J

Usage

```
:SENSe[1]:TEMPerature:TCouple[:TYPE] <type>
:SENSe[1]:TEMPerature:TCouple[:TYPE]?
```

<type>	Thermocouple type: B, E, J, K, N, R, S, or T

Example

The following program fragments measure voltage on Channel 1 and temperature on Channel 2. Temperature is configured using a simulated reference junction (typically an ice bath) and a type K thermocouple.

```
' Configure Temperature:
CALL SEND(7,":SENS:TEMP:TRANS TC",status%)   'Select thermocouple
       'sensor.
CALL SEND(7,":SENS:TEMP:RJUN:RSEL SIM",status%)   'Select simulated reference.
CALL SEND(7,":SENS:TEMP:RJUN:RSEL:SIM 0",status%)   'Set reference to 0°C.
CALL SEND(7,":SENS:TEMP:TC K",status%)   'Set for type K
       thermocouple.
CALL SEND(7,":UNIT:TEMP F",status%)   'Read in °F.

' Measure voltage on Channel 1:
CALL SEND(7,":SENS:CHAN 1",status%)   'Select Channel 1.
CALL SEND(7,":SENS:FUNC 'VOLT'",status%)   'Select DCV1.
CALL SEND(7,":SENS:DATA:FRES?",status%)   'Request a fresh
       'reading.
reading$ = SPACE$(80)
CALL ENTER(reading$, length%, 7, status%)   'Address 2182A to talk.
PRINT reading$   'Display reading on computer.

' Measure temperature on Channel 2:
CALL SEND(7,":SENS:CHAN 2",status%)   'Select Channel 2.
CALL SEND(7,":SENS:FUNC 'TEMP'",status%)   'Select TEMP 2.
CALL SEND(7,":SENS:DATA:FRES?",status%)   'Request a fresh
       'reading.
reading$ = SPACE$(80)
CALL ENTER(reading$, length%, 7,status%)   'Address 2182A to talk.
PRINT reading$   'Display reading on computer.
```

Also see

:SENSe[1]:TEMPerature:TRANsducer (on page 13-60)

:SENSe[1]:TEMPerature:TRANsducer

This command sets the transducer type.

Type	Affected by	Where saved	Default value
Command and query	Recall settings SYSTem:PRESet Instrument reset Power cycle	Save settings	TCouple

Usage

```
:SENSe[1]:TEMPerature:TRANsducer <type>
:SENSe[1]:TEMPerature:TRANsducer?
```

`<type>`	The type of transducer: ■ Thermocouple: `TCouple` ■ Internal: `INTernal`

Example

The following program fragments measure voltage on Channel 1 and temperature on Channel 2. Temperature is configured using a simulated reference junction (typically an ice bath) and a type K thermocouple.

```
' Configure Temperature:
CALL SEND(7,":SENS:TEMP:TRANS TC",status%)   'Select thermocouple
       'sensor.
CALL SEND(7,":SENS:TEMP:RJUN:RSEL SIM",status%)   'Select simulated reference.
CALL SEND(7,":SENS:TEMP:RJUN:RSEL:SIM 0",status%)   'Set reference to 0°C.
CALL SEND(7,":SENS:TEMP:TC K",status%)   'Set for type K
       thermocouple.
CALL SEND(7,":UNIT:TEMP F",status%)   'Read in °F.

' Measure voltage on Channel 1:
CALL SEND(7,":SENS:CHAN 1",status%)   'Select Channel 1.
CALL SEND(7,":SENS:FUNC 'VOLT'",status%)   'Select DCV1.
CALL SEND(7,":SENS:DATA:FRES?",status%)   'Request a fresh
       'reading.
reading$ = SPACE$(80)
CALL ENTER(reading$, length%, 7, status%)   'Address 2182A to talk.
PRINT reading$   'Display reading on computer.

' Measure temperature on Channel 2:
CALL SEND(7,":SENS:CHAN 2",status%)   'Select Channel 2.
CALL SEND(7,":SENS:FUNC 'TEMP'",status%)   'Select TEMP 2.
CALL SEND(7,":SENS:DATA:FRES?",status%)   'Request a fresh
       'reading.
reading$ = SPACE$(80)
CALL ENTER(reading$, length%, 7,status%)   'Address 2182A to talk.
PRINT reading$   'Display reading on computer.
```

Also see

:SENSe:FUNCtion (on page 13-55)

:SENSe[1]:VOLTage[:DC]:CHANnel2:LQMode

This command enables or disables low charge injection mode.

Type	Affected by	Where saved	Default value
Command and query	Recall settings SYSTem:PRESet Instrument reset Power cycle	Save settings	OFF

Usage

```
:SENSe[1]:VOLTage[:DC]:CHANnel2:LQMode <state>
:SENSe[1]:VOLTage[:DC]:CHANnel2:LQMode?
```

<state>	Disables low charge injection mode: OFF or 0 Enables low charge injection mode: ON or 1

Details

You cannot enable low charge injection mode from the front panel. However, it can be disabled from the front panel by restoring factory default conditions.

Example

This program fragment enables low charge injection for Channel 2.

```
CALL SEND(7,":SENS:VOLT:CHAN2:LQM ON",status%)    ' Enables low charge
   ' injection.
```

Also see

Pumpout current (low charge injection mode) (on page 4-27)

:SENSe[1]:VOLTage[:DC]:DELTa

This command enables or disables the delta measurement function.

Type	Affected by	Where saved	Default value
Command and query	Recall settings SYSTem:PRESet Instrument reset Power cycle	Save settings	OFF

Usage

```
:SENSe[1]:VOLTage[:DC]:DELTa <b>
:SENSe[1]:VOLTage[:DC]:DELTa?
```

<b>	Disable delta measurements: OFF Enable delta measurements: ON

Details

Basic delta calculation:

$$\text{Delta} = \frac{V1t1 - V1t2}{2}$$

Where:

- V1t1 is the voltage measurement on the positive phase of the current source.
- V1t2 is the voltage measurement on the negative phase of the current source.

Enabling ratio disables delta and conversely, enabling delta disables ratio.

Delta is only available for voltage measurements.

To double the speed of delta measurements, disable front autozero.

When this is set to `ON`, `:SENSe[1]:VOLTage[:DC]:RATio` and `:SENSe[1]:HOLD:STATe` are set to `OFF` and `:SENS:VOLT:DC:CHAN1:DFILT:TCON` is set to `MOV`.

Example

The following program fragment uses a SourceMeter with the 2182A to perform delta measurements. External triggering using Trigger Link is used to synchronize the source-measure operations between the two instruments. The front-panel autozero is disabled to double the speed of delta.

Three delta measurements are made: A source value at 10 µA, one at 20 µA, and one at 50 µA. The readings are stored in the buffer of the 2182A.

```
CALL SEND (7, ":SYST:PRES", status%)   ' 2182A - System preset
   ' defaults.
CALL SEND (7, ":TRIG:DEL 1", status%)   ' 2182A - 1 s delay.
CALL SEND (7, ":SENS:VOLT:DELTA ON", status%)   ' 2182A - Enable delta.
CALL SEND (7, ":SYST:FAZ OFF", status%)   ' Disable front autozero to
   ' double delta speed.
CALL SEND (7, ":TRIG:SOUR EXT", status%)   ' 2182A - External triggering.
CALL SEND (7, ":TRAC:POIN 3", status%)   ' 2182A - Buffer size 3.
CALL SEND (7, ":TRAC:FEED:CONT NEXT", status%)   ' 2182A - Enable buffer.
CALL SEND (24, "*RST", status%)   ' Reset SourceMeter to defaults.
CALL SEND (24, ":TRIG:SOUR TLIN", status%)   ' SourceMeter - Select trigger link.
CALL SEND (24, ":TRIG:DIR SOUR", status%)   ' SourceMeter - Enable source bypass.
CALL SEND (24, ":TRIG:OUTP SOUR", status%)   ' SourceMeter - Output trigger after
   ' source.
CALL SEND (24, ":TRIG:COUN 6", status%)   ' SourceMeter - Trig count 6.
CALL SEND (24, ":SOUR:FUNC CURR", status%)   ' SourceMeter - Source current.
CALL SEND (24, ":FUNC 'VOLT'", status%)   ' SourceMeter - Measure voltage
CALL SEND (24, ":VOLT:NPLC 0.01", status%)   ' SourceMeter - Fast measurements
CALL SEND (24, ":SOUR:LIST:CURR 10e-6, -10e-6,
20e-6, -20e-6, 50e-6, -50e-6",status%)   ' SourceMeter - Current list values
CALL SEND (24, ":OUTP ON", status%)   ' SourceMeter - Turn output on.
CALL SEND (24, ":SOUR:CURR:MODE LIST", status%)   ' SourceMeter - Enable list
   mode.
CALL SEND (24, ":INIT", status%)   ' SourceMeter - Start sweep.
```

Also see

Delta (on page 6-1)
Ratio (on page 4-31)
:SENSe[1]:HOLD:STATe (on page 13-56)
:SENSe[1]:VOLTage[:DC]:RATio (on page 13-63)
:SENSe[1]:<function>[:CHANnel<X>]:DFILter:TCONtrol (on page 13-46)
:SYSTem:FAZero[:STATe] (on page 13-83)

:SENSe[1]:VOLTage[:DC]:RATio

This command enables or disables the ratio function (V1/V2).

Type	Affected by	Where saved	Default value
Command and query	Recall settings SYSTem:PRESet Instrument reset Power cycle	Save settings	OFF

Usage

```
:SENSe[1]:VOLTage[:DC]:RATio <b>
:SENSe[1]:VOLTage[:DC]:RATio?
```

`<b>`	Disable ratio: `OFF` Enable ratio: `ON`

Details

When this is set `ON`, `:SENSe[1]:HOLD:STATe` and `:SENSe[1]:VOLTage[:DC]:DELTa` are set to `OFF`.

Enabling delta automatically sets this to `OFF`.

Example

The following program fragment enables ratio and displays the result on the computer.

```
CALL SEND (7, ":SENS:VOLT:FUNC 'VOLT'", status%)   ' Select voltage function.
CALL SEND (7, ":SENS:VOLT:CHAN 1", status%)   ' Select Channel 1 (DCV1) for
   ' range control.
CALL SEND (7, ":SENS:VOLT:RATIO", status%)   ' Enable ratio.
CALL SEND (7, ":SENS:DATA:FRESH?", status%)   ' Request a fresh reading.
reading$ = SPACE$(80)
CALL ENTER(reading$, length%, 7, status%)   ' Address 2182A to talk.
PRINT reading$   ' Display ratio reading on monitor.
```

Also see

Ratio (on page 4-31)
:SENSe[1]:HOLD:STATe (on page 13-56)
:SENSe[1]:VOLTage[:DC]:DELTa (on page 13-61)

:SENSe[1]:VOLTage[:DC][:CHANnel<X>]:RANGe[:UPPer]

This command determines the positive full-scale measure range.

Type	Affected by	Where saved	Default value
Command and query	Recall settings SYSTem:PRESet Instrument reset Power cycle	Save settings	Channel 1: 120 Channel 2: 12

Usage

```
:SENSe[1]:VOLTage[:DC][:CHANnel1]:RANGe[:UPPer] <n>
:SENSe[1]:VOLTage[:DC]:CHANnel2:RANGe[:UPPer] <n>
:SENSe[1]:VOLTage[:DC][:CHANnel1]:RANGe[:UPPer]?
:SENSe[1]:VOLTage[:DC]:CHANnel2:RANGe[:UPPer]?
```

`<n>`	The expected reading (voltage): ■ Channel 1: `0` to `120` ■ Channel 2: `0` to 1`2`

Details

The DCV1 function has five measurement ranges: 10 mV, 100 mV, 1 V, 10 V, and 100 V. The DCV2 function has three measurement ranges: 100 mV, 1 V, and 10 V.

Setting a range turns autorange off.

SCPI compliant.

Example

The following program fragment enables autoranging for DCV1 and sets DCV2 to the 1 V range:

```
CALL SEND(7,":SENS:VOLT:RANG:AUTO ON",status%)   'Enable autorange for DCV1.
CALL SEND(7,":SENS:VOLT:CHAN2:RANG 0.5",status%)   'Set DCV2 to 1 V range.
```

Also see

Range (on page 4-15)
:SENSe[1]:VOLTage[:DC][:CHANnel<X>]:RANGe:AUTO (on page 13-65)

:SENSe[1]:VOLTage[:DC][:CHANnel<X>]:RANGe:AUTO

This command determines if the measurement range is set manually or automatically for the selected measure function.

Type	Affected by	Where saved	Default value
Command and query	Recall settings SYSTem:PRESet Instrument reset Power cycle	Save settings	ON (1)

Usage

```
:SENSe[1]:VOLTage[:DC][:CHANnel1]:RANGe:AUTO <state>
:SENSe[1]:VOLTage[:DC]:CHANnel2:RANGe:AUTO <state>
:SENSe[1]:VOLTage[:DC][:CHANnel1]:RANGe:AUTO?
:SENSe[1]:VOLTage[:DC]:CHANnel2:RANGe:AUTO?
```

`<state>`	Set the measurement range manually: `OFF` or `0` Set the measurement range automatically: `ON` or `1`

Details

Autorange selects the best range in which to measure the signal that is applied to the input terminals of the instrument. When autorange is enabled, the range increases at 120 percent of range. The range decreases occur when the reading at 10 percent of nominal range. For example, if you are on the 1 V range and autorange is enabled, the instrument autoranges up to the 10 V range when the measurement exceeds 1.2 V. It autoranges down to the 100 mV range when the measurement falls below 1 V.

When this command is set to `OFF`, you must set the range. If you do not set the range, the instrument remains at the range that was last selected by autorange.

When this command is set to `ON`, the instrument automatically goes to the most sensitive range to perform the measurement.

If `:SENSe:VOLTage[:CHANnel<X>]:RANGe[:UPPer]` is set to a value, this is automatically set to `OFF`.

SCPI compliant.

Example

The following program fragment enables autoranging for DCV1 and sets DCV2 to the 1 V range:

```
CALL SEND(7,":SENS:VOLT:RANG:AUTO ON",status%)   'Enable autorange for DCV1.
CALL SEND(7,":SENS:VOLT:CHAN2:RANG 0.5",status%)   'Set DCV2 to 1 V range.
```

Also see

:SENSe:VOLTage[:CHANnel<X>]:RANGe[:UPPer] (on page 13-64)

STATus subsystem

The STATus subsystem controls the status registers of the instrument. For additional information on the status model, see Status model (on page 16-1).

Commands in this subsystem are not affected by `*RST` or `:SYSTem:PRESet`.

:STATus:MEASurement:CONDition?

This command reads condition registers.

Type	Affected by	Where saved	Default value
Query only	Not applicable	Not applicable	Not applicable

Usage

```
:STATus:MEASurement:CONDition?
```

Details

This command reads content of the measurement condition register. Each set of event registers (except the standard event register set) has a condition register. A condition register is similar to its corresponding event register, except that it is a real-time register that constantly updates to reflect the current operating status of the instrument. For register bit descriptions, refer to `:STATus:MEASurement[:EVENt]?`.

After sending this command and addressing the 2182A to talk, a decimal value is sent to the computer. The binary equivalent of this decimal value indicates which bits in the register are set.

For example, if sending `:STAT:MEAS:COND?` returns a decimal value of 512 (binary 0000001000000000), bit B9 of the Measurement Condition Register is set, indicating that the trace buffer is full.

Also see

:STATus:MEASurement[:EVENt]? (on page 13-68)

:STATus:MEASurement:ENABle

This command sets or reads the contents of the Measurement Event Enable Register of the status model.

Type	Affected by	Where saved	Default value
Command and query	STATus:PRESet	Not applicable	See Details

Usage

```
:STATus:MEASurement:ENABle <NRf>
:STATus:MEASurement:ENABle?
```

`<NRf>`	The status of the Measurement Status Register; see **Details**

Details

This command sets the contents of the measurement event enable register. Send the command with the decimal equivalent of the binary value that determines the state (0 or 1) of each bit in the appropriate register.

The event enable register is used as a mask for events. When a bit in an event enable register is cleared (0), the corresponding bit in the event register is masked, so it cannot set the corresponding summary bit of the next register set in the status structure. Conversely, when a bit in an event enable register is set (1), the corresponding bit in the event register is unmasked. When the unmasked bit in the event register sets, the summary bit of the next register set in the status structure sets.

The decimal weighting of the bits is shown in the following table. The sum of the decimal weights of the bits to set is sent as the <NRf> parameter. Set <NRf> to 0 to clear all bits. Set <NRf> to 65535 to set all bits.

Measurement event enable register

Bit position	**B9**	**B8**	**B7**	**B6**	**B5**	**B4**	**B3**	**B2**	**B1**	**B0**
Event	BFL	BHF	BAV	—	RAV	HL2	LL2	HL1	LL1	ROF
<n> (Decimal Weighting)	512 (2^9)	256 (2^8)	128 (2^7)	—	32 (2^5)	16 (2^4)	8 (2^3)	4 (2^2)	2 (2^1)	1 (2^0)
Value	0/1	0/1	0/1	—	0/1	0/1	0/1	0/1	0/1	0/1

Bits B10 through B15 are not shown because they are not used.

Values are:

- 1 = Event Bit Set
- 0 = Event Bit Cleared

Events are listed in the following table.

ROF	Reading overflow
LL1	Low limit 1
HL1	High limit 1
LL2	Low limit 2
HL2	High limit 2
RAV	Reading available
BAV	Buffer available
BHF	Buffer half full
BFL	Buffer full

Power-up and `:STATus:PRESet` clear all bits of the registers; `*CLS` has no effect.

Example

`:STAT:MEAS:ENAB 544`	Sets the BFL and RAV bits of the Measurement Event Enable Register, where: ■ BFL (bit B9) is decimal 512 ■ RAV (bit B5) is decimal 32 So `<n>` is 512+32 = 544

Also see

Measurement Event Register (on page 16-4)

:STATus:MEASurement[:EVENt]?

This command reads and clears the Measurement Event Register of the status model.

Type	Affected by	Where saved	Default value
Query only	Not applicable	Not applicable	Not applicable

Usage

```
:STATus:MEASurement[:EVENt]?
```

Details

This attribute reads the Measurement Event Register of the status model and then clears the register.

The instrument returns a decimal value that corresponds to the binary-weighted sum of all bits set in the register. For details, see Measurement Event Register (on page 16-4).

Measurement Event Register:

Bit B0, Reading Overflow (ROF): Set bit indicates that the reading exceeds the measurement range of the instrument.

Bit B1, Low Limit1 (LL1): Set bit indicates that the reading is less than the Low Limit 1 setting.

Bit B2, High Limit1 (HL1): Set bit indicates that the reading is greater than the High Limit 1 setting.

Bit B3, Low Limit 2 (LL2): Set bit indicates that the reading is less than the Low Limit 2 setting.

Bit B4, High Limit 2 (HL2): Set bit indicates that the reading is greater than the High Limit 2 setting.

Bit B5, Reading Available (RAV): Set bit indicates that a reading was made and processed.

Bit B6: Not used.

Bit B7, Buffer Available (BAV): Set bit indicates that there are at least two readings in the trace buffer.

Bit B8, Buffer Half Full (BHF): Set bit indicates that the trace buffer is half full.

Bit B9, Buffer Full (BFL): Set bit indicates that the trace buffer is full.

Bit position	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
Event	BFL	BHF	BAV	—	RAV	HL2	LL2	HL1	LL1	ROF
<n> (Decimal Weighting)	512 (2^9)	256 (2^8)	128 (2^7)	—	32 (2^5)	16 (2^4)	8 (2^3)	4 (2^2)	2 (2^1)	1 (2^0)
Value	0/1	0/1	0/1	—	0/1	0/1	0/1	0/1	0/1	0/1

Bits B10 through B15 are not shown because they are not used.

Values are:

- 1 = Event Bit Set
- 0 = Event Bit Cleared

Events are listed in the following table.

ROF	Reading overflow
LL1	Low limit 1
HL1	High limit 1
LL2	Low limit 2
HL2	High limit 2
RAV	Reading available
BAV	Buffer available
BHF	Buffer half full
BFL	Buffer full

Power-up and `*CLS` clears all bits. `:STATus:PRESet` has no effect.

Example

`:STAT:MEAS?`	Assume that reading the Measurement Event Register results in an acquired decimal value of `544`. The binary equivalent is `0000001000100000`. For this binary value, bits B5 and B9 of the Measurement Event Register are set.

Also see

None

:STATus:OPERation:CONDition?

This command reads the operation condition registers.

Type	Affected by	Where saved	Default value
Query only	Not applicable	Not applicable	Not applicable

Usage

```
:STATus:OPERation:CONDition?
```

Details

This command reads content of the operation condition register. Each set of event registers (except the standard event register set) has a condition register. A condition register is similar to its corresponding event register, except that it is a real-time register that constantly updates to reflect the current operating status of the instrument. For register bit descriptions, refer to `:STATus:OPERation[:EVENt]?`.

After sending this command and addressing the 2182A to talk, a decimal value is sent to the computer. The binary equivalent of this decimal value indicates which bits in the register are set.

For example, if sending `:STAT:MEAS:COND?` returns a decimal value of 512 (binary 0000001000000000), bit B9 of the Measurement Condition Register is set, indicating that the trace buffer is full.

Also see

:STATus:OPERation[:EVENt]? (on page 13-71)

:STATus:OPERation:ENABle

This command sets or reads the contents of the Operation Event Enable Register of the status model.

Type	Affected by	Where saved	Default value
Command and query	STATus:PRESet	Not applicable	0

Usage

```
:STATus:OPERation:ENABle <NRf>
:STATus:OPERation:ENABle?
```

`<NRf>`	The status of the operation status register; see **Details**

Details

When one of these bits is set, when the corresponding bit in the Operation Event Register or Operation Condition Register is set, the OSB bit in the Status Byte Register is set. Send the command with the decimal equivalent of the binary value that determines the state (`0` or `1`) of each bit in the appropriate register.

The event enable register is used as a mask for events. When a bit in an event enable register is cleared (`0`), the corresponding bit in the event register is masked, so it cannot set the corresponding summary bit of the next register set in the status structure. Conversely, when a bit in an event enable register is set (`1`), the corresponding bit in the event register is unmasked. When the unmasked bit in the event register sets, the summary bit of the next register set in the status structure sets.

The decimal weighting of the bits is shown in the following table. The sum of the decimal weights of the bits to set is sent as the `<NRf>` parameter. Set `<NRf>` to `0` to clear all bits. Set `<NRf>` to `65535` to set all bits.

Operation event enable register

Bit position	**B10**	**B9**	**B8**	**B7, B6**	**B5**	**B4**	**B3**	**B2**	**B1**	**B0**
Event	Idle	—	Filt	—	Trig	Meas	—	—	—	Cal
<n> (Decimal Weighting)	1024 (2^{10})	—	256 (2^8)	—	32 (2^5)	16 (2^4)	—	—	—	1 (2^0)
Value	0/1	—	0/1	—	0/1	0/1	—	—	—	0/1

Bits B11 through B15 are not shown because they are not used.

Values are:

- 1 = Enable operation event
- 0 = Disable (mask) operation event

Events are listed in the following table.

Idle	Idle state of the 2182A
Filt	Filter settled
Trig	Trigger layer
Meas	Measuring
Cal	Calibrating

Power-up and `:STATus:PRESet` clear all bits of the registers; `*CLS` has no effect.

Also see

Operation event registers (on page 16-3)

:STATus:OPERation[:EVENt]?

This command reads and clears the Operation Event Register of the status model.

Type	Affected by	Where saved	Default value
Query only	Not applicable	Not applicable	Not applicable

Usage

```
:STATus:OPERation[:EVENt]?
```

Details

This attribute reads the Operation Event Register of the status model and then clears the register.

The instrument returns a decimal value that corresponds to the binary-weighted sum of all bits set in the register. For details, see Operation event registers (on page 16-3).

Operation Event Register:

Bit B0, Calibrating (Cal): Set bit indicates that the instrument is calibrating.

Bits B1 through B3: Not used.

Bit B4, Measuring (Meas): Set bit indicates that the instrument is performing a measurement.

Bit B5, Trigger Layer (Trig): Set bit indicates that the instrument is waiting in the trigger layer of the trigger model.

Bits B6 and B7: Not used.

Bit B8, Filter Settled (Filt): Set bit indicates that the filter has settled.

Bit B9: Not used.

Bit B10, Idle: Set bit indicates that the instrument is in the idle state.

Bit position	B10	B9	B8	B7, B6	B5	B4	B3	B2	B1	B0
Event	Idle	—	Filt	—	Trig	Meas	—	—	—	Cal
<n> (Decimal Weighting)	1024 (2^{10})	—	256 (2^8)	—	32 (2^5)	16 (2^4)	—	—	—	1 (2^0)
Value	0/1	—	0/1	—	0/1	0/1	—	—	—	0/1

Bits B11 through B15 are not shown because they are not used.

Values are:

- 1 = Enable operation event
- 0 = Disable (mask) operation event

Events are listed in the following table.

Idle	Idle state of the 2182A
Filt	Filter settled
Trig	Trigger layer
Meas	Measuring
Cal	Calibrating

Power-up and `*CLS` clears all bits. `:STATus:PRESet` has no effect.

Also see

None

:STATus:PRESet

This function returns the registers to their default conditions.

Type	Affected by	Where saved	Default value
Command only	Not applicable	Not applicable	Not applicable

Usage

```
:STATus:PRESet
```

Details

When this command is sent, all bits of the following registers are cleared to zero (0):

- Questionable Event Enable Register
- Measurement Event Enable Register
- Operation Event Enable Register

NOTE

Registers not included in the above list are not affected by this command.

Example

`:STAT:PRES`	Returns the registers to their default conditions.

Also see

None

:STATus:QUEStionable:CONDition?

This command reads the Questionable Condition Register of the status model.

Type	Affected by	Where saved	Default value
Query only	Not applicable	Not applicable	Not applicable

Usage

```
:STATus:QUEStionable:CONDition?
```

Details

This command reads content of the questionable condition register. Each set of event registers (except the standard event register set) has a condition register. A condition register is similar to its corresponding event register, except that it is a real-time register that constantly updates to reflect the current operating status of the instrument. For register bit descriptions, refer to `:STATus:QUEStionable[:EVENt]?`.

After sending this command and addressing the 2182A to talk, a decimal value is sent to the computer. The binary equivalent of this decimal value indicates which bits in the register are set.

For example, if sending `:STAT:MEAS:COND?` returns a decimal value of 512 (binary 0000001000000000), bit B9 of the Measurement Condition Register is set, indicating that the trace buffer is full.

Example

`:STAT:QUES:COND?`	Reads the Questionable Condition Register.

Also see

:STATus:QUEStionable[:EVENt]? (on page 13-75)

:STATus:QUEStionable:ENABle

This command sets or reads the contents of the Questionable Event Enable Register of the status model.

Type	Affected by	Where saved	Default value
Command and query	STATus:PRESet	Not applicable	0

Usage

```
:STATus:QUEStionable:ENABle <NRf>
:STATus:QUEStionable:ENABle?
```

`<NRf>`	The status of the Questionable Status Register; see **Details**

Details

This command sets or reads the contents of the questionable event enable register. Send the command with the decimal equivalent of the binary value that determines the state (`0` or `1`) of each bit in the appropriate register.

The event enable register is used as a mask for events. When a bit in an event enable register is cleared (`0`), the corresponding bit in the event register is masked, so it cannot set the corresponding summary bit of the next register set in the status structure. Conversely, when a bit in an event enable register is set (`1`), the corresponding bit in the event register is unmasked. When the unmasked bit in the event register sets, the summary bit of the next register set in the status structure sets.

The decimal weighting of the bits is shown in the following table. The sum of the decimal weights of the bits to set is sent as the `<NRf>` parameter. Set `<NRf>` to `0` to clear all bits. Set `<NRf>` to `65535` to set all bits.

Questionable event enable register

Bit position	**B9**	**B8**	**B7**	**B6**	**B5**	**B4**	**B3**	**B2**	**B1**	**B0**
Event	ACAL	Cal	—	—	—	Temp	—	—	—	—
<n> (Decimal Weighting)	512 (2^9)	256 (2^8)	—	—	—	16 (2^4)	—	—	—	—
Value	0/1	0/1	—	—	—	0/1	—	—	—	—

Bits B10 through B15 are not shown because they are not used.

Values are:

- 1 = Event Bit Set
- 0 = Event Bit Cleared

Events are listed in the following table.

ACAL	ACAL summary
Cal	Calibration summary
Temp	Temperature summary

Power-up and `:STATus:PRESet` clear all bits of the registers; `*CLS` has no effect.

Example

`:stat:meas:enab 544`	Sets the BFL and RAV bits of the Measurement Event Enable Register, where: ▪ BFL (bit B9) is decimal 512 ▪ RAV (bit B5) is decimal 32 ▪ so `<n>` is 512+32 = 544

Also see

Enable registers (on page 16-8)
:STATus:PRESet (on page 13-73)

:STATus:QUEStionable[:EVENt]?

This command reads and clears the Questionable Event Register of the status model.

Type	Affected by	Where saved	Default value
Query only	Not applicable	Not applicable	Not applicable

Usage

```
:STATus:QUEStionable[:EVENt]?
```

Details

This attribute reads the Questionable Event Register of the status model and then clears the register.

The instrument returns a decimal value that corresponds to the binary-weighted sum of all bits set in the register. For details, see Questionable status registers* (on page 16-5).

Questionable Event Register:

Bits B0 through B3: Not used.

Bit B4, Temperature Summary (Temp): Set bit indicates that an invalid reference junction measurement has occurred for thermocouple temperature measurements.

Bits B5 through B7: Not used.

Bit B8, Calibration Summary (Cal): Set bit indicates that an invalid calibration constant was detected during the power-up sequence. The instrument will instead use a default calibration constant. This error will clear after successful calibration of the instrument.

Bit B9, ACAL Summary (ACAL): Set bit indicates that an invalid ACAL was performed. This error will clear after a successful ACAL is performed.

Bits B10 through B15: Not used.

NOTE

Whenever a questionable event occurs, the ERR annunciator turns on. The annunciator turns off when the questionable event clears.

Bit position	**B9**	**B8**	**B7**	**B6**	**B5**	**B4**	**B3**	**B2**	**B1**	**B0**
Event	ACAL	Cal	—	—	—	Temp	—	—	—	—
<n> (Decimal Weighting)	512 (2^9)	256 (2^8)	—	—	—	16 (2^4)	—	—	—	—
Value	0/1	0/1	—	—	—	0/1	—	—	—	—

Bits B10 through B15 are not shown because they are not used.

Values are:

- 1 = Event Bit Set
- 0 = Event Bit Cleared

Events are listed in the following table.

ACAL	ACAL summary
Cal	Calibration summary
Temp	Temperature summary

Power-up and `*CLS` clears all bits. `:STATus:PRESet` has no effect.

Also see

Status model (on page 16-1)

:STATus:QUEue:CLEar

This function clears all messages from the error queue.

Type	Affected by	Where saved	Default value
Command only	Not applicable	Not applicable	Not applicable

Usage

```
:STATus:QUEue:CLEar
```

Also see

Queues (on page 16-6)

:STATus:QUEue:DISable

This function specifies the messages that are not to be placed in the error queue.

Type	Affected by	Where saved	Default value
Command and query	Power cycle	Not applicable	Not applicable

Usage

```
:STATus:QUEue:DISable <list>
:STATus:QUEue:DISable?
```

`<list>`	The messages to disable; see **Details**

Details

The query reads the disabled messages.

On power-up, all error messages are enabled and go into the error queue as they occur. This command specifies the messages you want to disable. Disabled messages are prevented from entering the queue. To disable all messages from entering the error queue, send the command:

```
:STAT:QUE:DIS ()
```

NOTE

Status messages are not enabled and do not go into the queue.

Messages are specified by number. Refer to "Status and error messages (on page 11-4)" in the *Model 2182A User's Manual* for message numbers.

Power-up clears the list of messages. `*CLS` and `:STATus:PRESet` have no effect.

Example

`:STATus:QUEue:DISable -110`	Disable single message.
`:STATus:QUEue:DISable -110, -140, -222`	Disable multiple messages (separated by commas).
`:STATus:QUEue:DISable -110:-222`	Disable a range of messages (start and end separated by a colon).
`:STATus:QUEue:DISable -110:-222, -230`	Disable a range (separated by a colon) and a single entry (separated by a comma).

Also see

Queues (on page 16-6)
:STATus:QUEue:ENABle (on page 13-78)

:STATus:QUEue:ENABle

This function returns the most recent error queue message.

Type	Affected by	Where saved	Default value
Command and query	Power cycle	Not applicable	Not applicable

Usage

```
:STATus:QUEue:ENABle <list>
:STATus:QUEue:ENABle?
```

`<list>`	A specified list of messages to enable for the error queue; see **Details**

Details

On power-up, all error messages are enabled and go into the error queue as they occur. This command specifies which messages you want to enable. Messages that are not specified are disabled and prevented from entering the queue.

NOTE

Status messages are not enabled and do not go into the queue.

When this command is sent, all messages are disabled, then the specified messages are enabled. The query contains all the enabled messages. To enable all messages to enter the error queue, send the command:

```
:STAT:QUE:ENAB ()
```

Messages are specified by numbers. A list of messages is provided in "Status and error messages (on page 11-4)" in the *Model 2182A User's Manual*.

The list of messages is cleared on power up. `*CLS` and `:STATus:PRESet` have no effect.

Example

`:STATus:QUEue:ENABle -110`	Enable single message.
`:STATus:QUEue:ENABle -110, -140, -222`	Enable multiple messages (separated by commas).
`:STATus:QUEue:ENABle -110:-222`	Enable a range of messages (start and end separated by a colon).
`:STATus:QUEue:ENABle -110:-222, -230`	Enable a range (separated by a colon) and a single entry (separated by a comma).

Also see

Queues (on page 16-6)
:STATus:QUEue:DISable (on page 13-77)

:STATus:QUEue[:NEXT]?

This function returns the oldest error queue message from the Event Queue and removes it.

Type	Affected by	Where saved	Default value
Query only	Not applicable	Not applicable	Not applicable

Usage

```
:STATus:QUEue[:NEXT]?
```

Details

As error and status messages occur, they are placed in the error queue. This command reads those messages. The error queue is a first-in, first-out (FIFO) register that can hold up to 10 messages. Each time you read the queue, the oldest message is read, and that message is then removed from the queue.

If the queue is full, the `350, Queue Overflow` message occupies the last memory location in the register. When the error queue is empty, the message `0, No error` is placed in the error queue.

The messages in the queue are preceded by a number. Negative (–) numbers indicate SCPI-defined messages and positive (+) numbers indicate messages defined by Keithley. Refer to "Status and error messages (on page 11-4)" in the *Model 2182A User's Manual* for the list of messages.

NOTE

The `:STATus:QUEue[:NEXT]?` query command performs the same function as the `:SYSTem:ERRor?` query command.

Example

`STAT:QUE?`
Returns information on the oldest error in the event log. For example, if you sent a command without a parameter, the return is: `-109,"Missing parameter;1;2017/05/06 12:57:04.484"`

Also see

:SYSTem:ERRor? (on page 13-82)

SYSTem subsystem

This subsystem contains commands that affect the overall operation of the instrument, such as passwords, beepers, communications, event logs, and time.

:SYSTem:AZERo[:STATe]

This command enables and disables autozero.

Type	Affected by	Where saved	Default value
Command and query	Recall settings SYSTem:PRESet Instrument reset Power cycle	Save settings	ON

Usage

```
:SYSTem:AZERo[:STATe] <state>
:SYSTem:AZERo[:STATe]?
```

<state>	Disable autozero: `OFF` or `0` Enable autozero: `ON` or `1`

Details

When autozero is disabled, the zero and gain reference measurements are not performed. This increases the measurement speed. However, the zero and gain reference points eventually drift, resulting in inaccurate readings of the input signal. It is recommended that autozero only be disabled for short periods.

Example

This program fragment disables autozero.

```
CALL SEND(7,":SYST:AZER OFF",status%)    ' Disables autozero.
```

Also see

Autozero modes (on page 4-17)

:SYSTem:BEEPer[:STATe]

This command enables or disables the instrument beeper for limit tests and HOLD.

Type	Affected by	Where saved	Default value
Command and query	Recall settings SYSTem:PRESet Instrument reset Power cycle	Save settings	ON

Usage

```
:SYSTem:BEEPer[:STATe] <b>
:SYSTem:BEEPer[:STATe]?
```

`<b>`	Disable the beeper: `OFF` or `0` Enable the beeper: `ON` or `1`

Details

SCPI compliant.

Also see

None

:SYSTem:CLEar

This command clears messages from the error queue.

Type	Affected by	Where saved	Default value
Command only	Not applicable	Not applicable	Not applicable

Usage

```
:SYSTem:CLEar
```

Details

SCPI compliant.

Also see

:SYSTem:ERRor? (on page 13-82)

:SYSTem:ERRor?

This command returns the oldest unread error message from the event log and removes it from the log.

Type	Affected by	Where saved	Default value
Query only	Not applicable	Not applicable	Not applicable

Usage

```
:SYSTem:ERRor?
```

Details

As error and status messages occur, they are placed in the error queue. This command reads those messages. The error queue is a first-in, first-out (FIFO) register that can hold up to 10 messages. Each time you read the queue, the oldest message is read, and that message is then removed from the queue.

If the queue is full, the `350, Queue Overflow` message occupies the last memory location in the register. When the error queue is empty, the message `0, No error` is placed in the error queue.

The messages in the queue are preceded by a number. Negative (–) numbers indicate SCPI-defined messages and positive (+) numbers indicate messages defined by Keithley. Refer to "Status and error messages (on page 11-4)" in the *Model 2182A User's Manual* for the list of messages.

The error queue is cleared at power-up and when `*CLS` is sent. `*RST`, `:SYSTem:PRESet`, and `:STATus:PRESet` have no effect on the error queue.

NOTE

The `:SYSTem:ERRor?` query command is similar to the `:STATus:QUEue?` command.

Example

SYST:ERR?
Returns information on the oldest error in the event log. For example, if you sent a command without a parameter, the return is: `-109,"Missing parameter;1;2017/05/06 12:57:04.484"`

Also see

Queues (on page 16-6)
:STATus:QUEue[:NEXT]? (on page 13-79)

:SYSTem:FAZero[:STATe]

This command enables or disables front autozero.

Type	Affected by	Where saved	Default value
Command and query	Recall settings SYSTem:PRESet Instrument reset Power cycle	Save settings	ON

Usage

```
:SYSTem:FAZero[:STATe] <b>
:SYSTem:FAZero[:STATe]?
```

<b>	Disable front autozero: OFF Enable front autozero: ON

Details

When front autozero is enabled, the instrument performs two A/D measurement cycles for each reading. The first one is a normal measurement cycle. The second measurement cycle is performed with the polarity of the front-end amplifier reversed. This current-reversal measurement technique cancels internal offsets in the amplifier. When front autozero is disabled, the second measurement cycle is not performed.

You can double the speed for delta measurements by disabling front autozero. Delta uses its own current-reversal technique to cancel offsets. Therefore, the dual measurement technique of front autozero is not required.

Example

This programming fragment disables front autozero.

```
CALL SEND (7, ":SYST:FAZ OFF", status%)
```

Also see

Autozero modes (on page 4-17)
Delta (on page 6-1)
:SENSe[1]:CHANnel<n>:VOLTage[:DC]:DELTa (on page 13-61)

:SYSTem:KCLick

This command allows you to enable or disable the key click sound.

Type	Affected by	Where saved	Default value
Command and query	Recall settings SYSTem:PRESet Instrument reset Power cycle	Save settings	ON

Usage

```
:SYSTem:KCLick <state>
:SYSTem:KCLick?
```

`<state>`	Disable the key click: `OFF` or `0` Enable the key click: `ON` or `1`

Details

This command enables or disables the key click . When enabled, when you press a key on the front panel, a sound is output.

You can also enable or disable the keyclick from the front panel by pressing **SHIFT** then **LOCAL**.

Example

`:SYSTem:KCLick ON`	Enable the key clicks.

Also see

None

:SYSTem:KEY

This command allows you to simulate a front-panel key press using remote commands.

Type	Affected by	Where saved	Default value
Command and query	Recall settings SYSTem:PRESet Instrument reset Power cycle	Save settings	Not applicable

Usage

```
:SYSTem:KEY <NRf>
:SYSTem:KEY?
```

`<NRf>`	The number that corresponds to the key press; see **Details**

Details

The queue for `:SYSTem:KEY?` holds one key-press value. When `:SYSTem:KEY?` is sent over the bus and the 2182A is addressed to talk, the key-press code number for the last key pressed (either physically or with `:SYSTem:KEY?`) is sent to the computer.

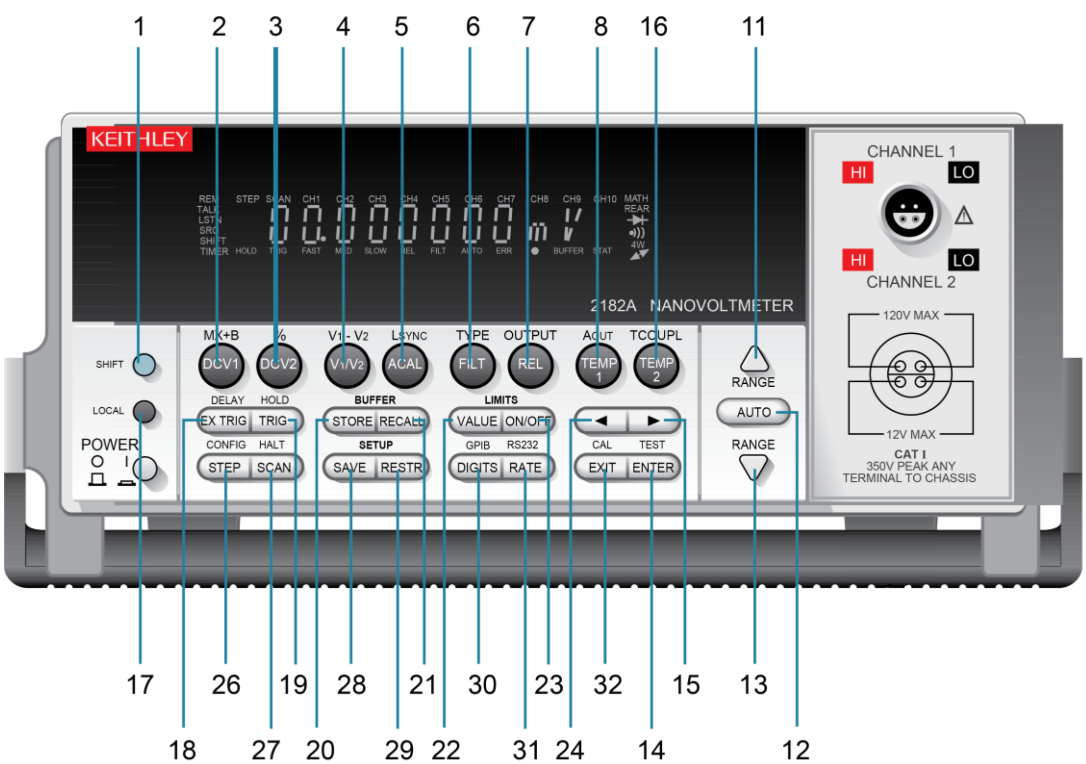
NOTE

Do not use `:SYSTem:KEY` to put the 2182A into local mode. Use `GTL` instead.

The value of `<NRf>` that corresponds to each key is listed in the following table. They are also shown in the figure that follows the table.

<NRf>	Key	<NRf>	Key
1	SHIFT	17	LOCAL
2	DCV1	18	EX-TRIG
3	DCV2	19	TRIG
4	V1/V2 (ratio)	20	STORE
5	ACAL	21	RECALL
6	FILT	22	VALUE
7	REL	23	ON/OFF
8	TEMP 1	24	◀
9	—	25	—
10	—	26	STEP
11	▲	27	SCAN
12	AUTO	28	SAVE
13	▼	29	RESTR
14	ENTER	30	DIGITS
15	▶	31	RATE
16	TEMP 2	32	EXIT

Figure 88: Key-press codes



SCPI compliant.

Example

`:SYST:KEY 4`	Simulate a RATIO front-panel key press.

Also see

None

:SYSTem:LFRequency?

This query contains the power line frequency setting that is used for NPLC calculations.

Type	Affected by	Where saved	Default value
Query only	Power cycle	Not applicable	Not applicable

Usage

```
:SYSTem:LFRequency?
```

Details

The instrument automatically detects the power line frequency when the instrument is powered on. Power line frequency can be 50 Hz, 60 Hz, or 400 Hz. If the line frequency is 400 Hz, `50` is returned.

Example

`:SYST:LFR?`	Check the line frequency.

Also see

Instrument power (on page 2-1)

:SYSTem:LSYNc[:STATe]

This command enables or disables line synchronization.

Type	Affected by	Where saved	Default value
Command and query	Recall settings SYSTem:PRESet Instrument reset Power cycle	Save settings	OFF

Usage

```
:SYSTem:LSYNc[:STATe] <state>
:SYSTem:LSYNc[:STATe]?
```

`<state>`	Disable line cycle synchronization: `OFF` or `0` Enable line cycle synchronization: `ON` or `1`

Details

When enabled, A/D conversions are synchronized with the power line frequency. This reduces noise at the expense of speed. Line cycle synchronization is not available for integration rates <1 PLC, regardless of the `:SYStem:LSYNC` setting.

Example

This program fragment enables line cycle synchronization.
`CALL SEND(7,":SYST:LSYNC ON",status%)    ' Enables LSYNC.`

Also see

Line cycle synchronization (on page 4-27)

:SYSTem:POSetup

This command selects the defaults when you power on the instrument.

Type	Affected by	Where saved	Default value
Command and query	Not applicable	Nonvolatile memory	RST

Usage

```
:SYSTem:POSetup <name>
:SYSTem:POSetup?
```

`<name>`	Which setup to restore when you power on the instrument: ■ `*RST` defaults: `RST` ■ `:SYSTem:PRESet` defaults: `PRESet` ■ Stored setup: `SAV0`

Details

When you select `RST`, the instrument restores settings to their default values when the instrument is powered on.

When you select `PRESet`, the instrument restores settings to the `:SYSTem:PRESet` default conditions when the instrument is powered on.

When you select `SAV0`, the settings in the saved setup are applied when the instrument is powered on. The settings are saved using the `*SAV` command.

Default settings are listed in the SCPI command descriptions.

Example

`SYST:POS SAV1`	Set the instrument to restore the settings that are saved in stored setup 1 when the instrument is powered on.

Also see

*RST (on page 14-10)
*SAV (on page 14-11)
:SYSTem:PRESet (on page 13-88)

:SYSTem:PRESet

This command returns the instrument to states optimized for front-panel operation.

Type	Affected by	Where saved	Default value
Command only	Not applicable	Not applicable	Not applicable

Usage

```
:SYSTem:PRESet
```

Details

SCPI compliant.

Example

`:SYST:PRES`	Set the instrument to states optimized for front-panel operation.

Also see

*RST (on page 14-10)

:SYSTem:TSTamp:RESet

This command resets the 2182A timestamp.

Type	Affected by	Where saved	Default value
Command only	Not applicable	Not applicable	Not applicable

Usage

```
:SYSTem:TSTamp:RESet
```

Details

Resets the timestamp to `0`.

Also see

None

:SYSTem:VERSion?

This command returns the present SCPI version.

Type	Affected by	Where saved	Default value
Query only	Not applicable	Not applicable	Not applicable

Usage

```
:SYSTem:VERSion?
```

Details

This query command returns the SCPI version.

SCPI compliant.

Example

`SYSTem:VERSion?`	Query the version. An example of a return is: `1996.0`

Also see

None

TRACe subsystem

The TRACe subsystem contains commands that control the reading buffers.

You can use either `:TRACe` or `:DATA` as the root command.

`:SYSTem:PRESet` and `*RST` have no effect on the commands in this subsystem.

:TRACe:CLEar

This command clears readings from the buffer.

Type	Affected by	Where saved	Default value
Command only	Not applicable	Not applicable	Not applicable

Usage

```
:TRACe:CLEar
:DATA:CLEar
```

Details

When the readings are cleared, `:TRACe:FEED:CONTrol` is set to `NEV`.

SCPI compliant.

Example

This programming fragment demonstrates a simple method for making and displaying a specified number of readings. The number of readings is specified by the `:SAMPle:COUNt` command. When `:READ?` is asserted, the specified number of readings is made. After all the readings are made, they are sent to the computer and stored in the buffer.
This example makes 10 readings on the `DCV1` function and displays them on the computer.

```
' Reset controls, clear buffer and place 2182A in idle.
CALL SEND(7, "*RST", status%)
CALL SEND(7, "TRAC:CLE", status%)
CALL SEND(7, "SAMPLE:COUN 10", status%)
CALL SEND(7, "FORM:ELEM read,unit", status%)
CALL SEND(7, "READ?", status%)
reading$ = SPACE$ (300)
CALL ENTER(reading$, length%, 16, status%)
PRINT reading$
```

Also see

:TRACe:FEED:CONTrol (on page 13-92)

:TRACe:DATA?

This command reads all readings in the buffer.

Type	Affected by	Where saved	Default value
Query only	Not applicable	Not applicable	Not applicable

Usage

```
:TRACe:DATA?
:DATA:DATA?
```

Details

SCPI compliant.

Example

This programming fragment sets a buffer size of 20 readings, stores readings, and requests the stored readings.

```
' Store Readings:
CALL SEND(7,":TRAC:POIN 20",status%)    'Set buffer size to 20.
CALL SEND(7,":TRAC:FEED SENS",status%)    'Store raw input readings.
CALL SEND(7,":TRAC:FEED:CONT NEXT", status%)    'Start storing readings.
CALL SEND(7,":TRAC:DATA?",status%)    'Request all stored
   'readings.
```

Also see

None

:TRACe:FEED

This command selects raw input readings or readings with calculations applied.

Type	Affected by	Where saved	Default value
Command and query	Not applicable	Not applicable	Not applicable

Usage

```
:TRACe:FEED <name>
:DATA:FEED <name>
:TRACe:FEED?
:DATA:FEED?
```

`<name>`	The source of the readings that are stored in the reading buffer: ■ Store raw input readings: `SENSe[1]` ■ Store results of the mx+b or percent calculation: `CALCulate` ■ Do not store results in the buffer: `NONE`

Details

This command controls the source of the readings that are stored in the buffer.

SCPI compliant.

Example

This programming fragment sets a buffer size of 20 readings, stores readings, and requests the stored readings.

```
' Store Readings:
CALL SEND(7,":TRAC:POIN 20",status%)    'Set buffer size to 20.
CALL SEND(7,":TRAC:FEED SENS",status%)    'Store raw input readings.
CALL SEND(7,":TRAC:FEED:CONT NEXT", status%)    'Start storing readings.
CALL SEND(7,":TRAC:DATA?",status%)    'Request all stored
   'readings.
```

Also see

Calculations that you can apply to measurements (on page 4-28)
CALCulate[1] subsystem (on page 13-6)

:TRACe:FEED:CONTrol

This command selects starts and stops buffer storage.

Type	Affected by	Where saved	Default value
Command and query	TRACe:CLEar TRACe:POINts	Not applicable	Not applicable

Usage

```
:TRACe:FEED:CONTrol <name>
:DATA:FEED:CONTrol <name>
:TRACe:FEED:CONTrol?
:DATA:FEED:CONTrol?
```

`<name>`	The buffer control mode: ■ Start the storage process: `NEVer` ■ Stop the storage process: `NEXT`

Details

When storage is complete, buffer control automatically returns to `NEVer`.

This setting is set to `NEVer` when `:TRACe:POINts` is changed and when `:TRACe:CLEar` is sent.

SCPI compliant.

Example

This programming fragment sets a buffer size of 20 readings, stores readings, and requests the stored readings.

```
' Store Readings:
CALL SEND(7,":TRAC:POIN 20",status%)    'Set buffer size to 20.
CALL SEND(7,":TRAC:FEED SENS",status%)    'Store raw input readings.
CALL SEND(7,":TRAC:FEED:CONT NEXT", status%)    'Start storing readings.
CALL SEND(7,":TRAC:DATA?",status%)    'Request all stored
   'readings.
```

Also see

:TRACe:CLEar (on page 13-89)
:TRACe:POINts (on page 13-93)

:TRACe:FREE?

This command reads the status of storage memory.

Type	Affected by	Where saved	Default value
Query only	Not applicable	Not applicable	Not applicable

Usage

```
:TRACe:FREE?
:DATA:FREE?
```

Details

After sending this command and addressing the 2182A to talk, two values separated by commas are sent to the computer. The first value indicates how many bytes of memory are available, and the second value indicates how many bytes are reserved to store readings.

SCPI compliant.

Also see

None

:TRACe:POINts

This command sets the number of readings a buffer can store.

Type	Affected by	Where saved	Default value
Command and query	Not applicable	Not applicable	Not applicable

Usage

```
:TRACe:POINts <newSize>
:DATA:POINts <newSize>
:TRACe:POINts?
:DATA:POINts?
```

`<newSize>`	The new size for the buffer: `2` to `1024`

Details

Changing this setting changes `:TRACe:FEED:CONTrol` to `NEV`.

SCPI compliant.

Example

This programming fragment sets a buffer size of 20 readings, stores readings, and requests the stored readings.

```
' Store Readings:
CALL SEND(7,":TRAC:POIN 20",status%)    'Set buffer size to 20.
CALL SEND(7,":TRAC:FEED SENS",status%)    'Store raw input readings.
CALL SEND(7,":TRAC:FEED:CONT NEXT", status%)    'Start storing readings.
CALL SEND(7,":TRAC:DATA?",status%)    'Request all stored
   'readings.
```

Also see

:TRACe:FEED:CONTrol (on page 13-92)

TRIGger subsystem

The commands in this subsystem configure and control the trigger operations, including the trigger model.

:ABORt

This command stops all trigger model commands on the instrument.

Type	Affected by	Where saved	Default value
Command only	Not applicable	Not applicable	Not applicable

Usage

```
:ABORt
```

Details

When an `:ABORt` command is sent with continuous initiation disabled, the 2182A goes into the idle state. With continuous initiation enabled, operation continues at the top of the trigger model.

SCPI compliant.

Also see

Trigger model operation (on page 9-2)

:INITiate:CONTinuous

This command enables or disables continuous initiation.

Type	Affected by	Where saved	Default value
Command and query	SYSTem:PRESet Reset	Not applicable	See **Details**

Usage

```
:INITiate:CONTinuous <b>
:INITiate:CONTinuous?
```

`<b>`	Disable continuous initiation: `OFF` Enable continuous initiation: `ON`

Details

With continuous initiation enabled, you cannot use the `READ?` command or set the sample count (`SAMPle:COUNt`) to more than one.

Defaults for continuous initiation:

- `SYSTem:PRESet` enables continuous initiation.
- `*RST` disables continuous initiation.

The `:INITiate` commands remove the 2182A from the idle state. The device operations of `:INITiate` do not complete until the 2182A returns to idle. You can send the `*WAI` command after the `:INITiate` command to wait until the 2182A returns to idle before executing subsequent commands.

This is an overlapped command.

SCPI compliant.

Example

```
CALL SEND (7, ":SYST:PRES", status%)
CALL SEND (7, ":INIT:CONT OFF; :ABORT", status%)   'Place 2182A in idle.
CALL SEND (7, ":TRIG:COUN 1;SOUR TIM, status%)
CALL SEND (7, ":SAMP:COUN 30", status%)   'Program for 30
   'measurements then stop.
CALL SEND (7, ":INIT; *WAI", status%)   'Start measurements and send
   '*WAI
CALL SEND (7, ":DATA?", status%)   'Query a reading.
reading$ = SPACE$(80)
CALL ENTER (reading$, length%, 7, status%)   'Get a response when 2182A
   'goes into idle.
PRINT reading$   'Display the reading.
```

Also see

*WAI (on page 14-16)

:INITiate[:IMMediate]

This command starts the trigger model.

Type	Affected by	Where saved	Default value
Command only	Not applicable	Not applicable	Not applicable

Usage

```
:INITiate[:IMMediate]
```

Details

`:INITiate` removes the 2182A from the idle state. The device operations of `:INITiate` do not complete until the 2182A returns to idle. You can send the `*WAI` command after the `:INITiate` command to wait until the 2182A returns to idle before executing subsequent commands.

If the trigger model is running, sending this command causes an error and the command is ignored.

This is an overlapped command.

SCPI compliant.

Example

```
INIT
*WAI
```

Starts the trigger model and then waits until the commands are complete to accept new commands.

Also see

:ABORt (on page 13-94)
*OPC? (on page 14-8)
*WAI (on page 14-16)
Trigger model (remote operation) (on page 9-2)

:SAMPle:COUNt

This command sets the sample count.

Type	Affected by	Where saved	Default value
Command and query	Recall settings SYSTem:PRESet Instrument reset Power cycle	Save settings	1

Usage

```
:SAMPle:COUNt <n>
:SAMPle:COUNt?
```

<n>	Sample count: 1 to 1024

Details

A sample count >1 specifies how many readings are automatically stored in the buffer.

If continuous initiation is enabled, you cannot set the sample count greater than one.

Example

`SAMP:COUN 10`	Sets the sample count to 10.

Also see

:INITiate:CONTinuous (on page 13-94)

:TRIGger[:SEQuence[1]]:COUNt

This command sets the timer count.

Type	Affected by	Where saved	Default value
Command and query	SYSTem:PRESet Instrument reset	Save settings	See **Details**

Usage

```
:TRIGger[:SEQuence[1]]:COUNt <n>
:TRIGger[:SEQuence[1]]:COUNt?
```

<n>	Trigger count: 1 to 9999 or INF (infinite)

Details

Defaults for trigger count:

- SYSTem:PRESet sets the count to INF.
- *RST sets the count to 1.

SCPI compliant.

Example

:TRIG:COUN INF	Set the trigger count to infinite.

Also see

:SYSTem:PRESet (on page 13-88)
*RST (on page 14-10)

:TRIGger[:SEQuence[1]]:DELay

This command sets the trigger delay.

Type	Affected by	Where saved	Default value
Command and query	Recall settings SYSTem:PRESet Instrument reset Power cycle	Save settings	0

Usage

```
:TRIGger[:SEQuence[1]]:DELay <n>
:TRIGger[:SEQuence[1]]:DELay?
```

<n>	The delay in seconds: 0 to 999999.999

Details

SCPI compliant.

Example

`TRIG:DEL 10`	Set a 10 second delay.

Also see

None

:TRIGger[:SEQuence[1]]:DELay:AUTO

This command enables or disables an automatic delay.

Type	Affected by	Where saved	Default value
Command and query	Recall settings SYSTem:PRESet Instrument reset Power cycle	Save settings	ON; see **Details**

Usage

```
:TRIGger[:SEQuence[1]]:DELay:AUTO <b>
:TRIGger[:SEQuence[1]]:DELay:AUTO?
```

`<b>`	Disable the automatic delay: `OFF` Enable the automatic delay: `ON`

Details

Disabling automatic delay sets the delay time to `0`.

The automatic delay is 5 ms for the 100 V range and 1 ms for other ranges.

SCPI compliant.

Example

Makes a one-shot reading using an external trigger with the automatic delay enabled.

```
*RST
:INITiate:CONTinuous OFF;:ABORt
:TRIGger:SOURce EXTernal
:TRIGger:DELay:AUTO ON   // Note: Automatic trigger delay only takes effect with
   // trigger source set for BUS or EXTernal.
:SENSe:FUNCtion 'VOLTage:DC'
:SENSe:VOLTage:DC:RANGe:AUTO ON
:INITiate
(external trigger)
:SENSe:DATA:FRESh?
(Enter reading)   // This step times out if the trigger has not occurred.
```

Also see

:TRIGger[:SEQuence[1]]:DELay (on page 13-97)

:TRIGger[:SEQuence[1]]:SIGNal

This command bypasses the control source.

Type	Affected by	Where saved	Default value
Command only	Not applicable	Not applicable	Not applicable

Usage

```
:TRIGger[:SEQuence[1]]:SIGNal
```

Details

This command bypasses the control source if you do not want to wait for the programmed event to occur. The instrument must be waiting at the control source for the event when this command is sent. Otherwise, an error occurs and the command is ignored.

SCPI compliant.

Example

`:TRIG:SIGN`	Bypass the control source event.

Also see

None

:TRIGger[:SEQuence[1]]:SOURce

This command selects the trigger control source, which holds operation until the programmed event occurs.

Type	Affected by	Where saved	Default value
Command and query	Recall settings SYSTem:PRESet Instrument reset Power cycle	Save settings	IMMediate

Usage

```
:TRIGger[:SEQuence[1]]:SOURce <name>
:TRIGger[:SEQuence[1]]:SOURce?
```

`<name>`	The control source: ■ `IMMediate` ■ `MANual` ■ `TIMer` ■ `EXTernal` ■ `BUS`

Details

The control source options are as follows:

- `IMMediate`: Event detection is immediately satisfied, allowing operation to continue.
- `MANual`: Event detection is satisfied by pressing the TRIG key. The 2182A must be in LOCAL mode for it to respond to the TRIG key. Press the LOCAL key or send `LOCAL 7` over the bus to remove the instrument from the remote mode.
- `TIMer`: Event detection is immediately satisfied on the initial pass through the loop. Each subsequent detection is satisfied when the programmed timer interval elapses. The timer source is only available during step/scan operation. The timer resets to its initial state when the instrument goes into the normal mode of operation or into the idle state.
- `EXTernal`: Event detection is satisfied when an input trigger through the TRIGGER LINK connector is received by the 2182A.
- `BUS`: Event detection is satisfied when the 2182A receives a bus trigger (`GET` or `*TRG`).

When `BUS` or `EXTernal` is selected, you cannot use the `:READ?` command. Use `:FETCh?` or `:DATA:FRESh?` instead.

SCPI compliant.

Example

```
*RST
:INITiate:CONTinuous OFF;:ABORt
:TRIGger:SOURce BUS
:SENSe:FUNCtion 'VOLTage:DC'
:SENSe:VOLTage:DC:RANGe:AUTO ON
:TRIGger:COUNt 1
:INITiate
*TRG // Triggers reading. Can also use GPIB GET command.
:SENSe:DATA:FRESh?
```

Makes a one-shot reading on dc volts using the bus trigger and autoranging.

Also see

:FETCh? (on page 13-3)
:READ? (on page 13-5)
:SENSe[1]:DATA:FRESh? (on page 13-54)
:TRIGger[:SEQuence[1]]:TIMer (on page 13-101)
Trigger model (remote operation (on page 9-2)

:TRIGger[:SEQuence[1]] :TIMer

This command sets the timer interval.

Type	Affected by	Where saved	Default value
Command and query	Recall settings SYSTem:PRESet Instrument reset Power cycle	Save settings	0.1

Usage

```
:TRIGger[:SEQuence[1]]:TIMer <interval>
:TRIGger:[SEQuence[1]]:TIMer <DEF|MIN|MAX>
:TRIGger[:SEQuence[1]]:TIMer?
:TRIGger:[SEQuence[1]]:TIMer? <DEF|MIN|MAX>
```

<interval>	Timer interval in seconds: 0 to 999999.99

Details

A delay is the period after the timer is triggered and before the timer generates a trigger event. Each time the timer is triggered, it uses this delay.

Reading this command returns the delay interval that is used the next time the timer is triggered.

SCPI compliant.

Also see

None

UNIT subsystem

The UNIT subsystem selects measurement units for temperature readings.

:UNIT:TEMPerature

This command sets the units of measurement that are displayed on the front panel of the instrument and stored in the reading buffer.

Type	Affected by	Where saved	Default value
Command and query	Recall settings SYSTem:PRESet Instrument reset Power cycle	Save settings	C

Usage

```
:UNIT:TEMPerature <unitOfMeasure>
:UNIT:TEMPerature?
```

<unitOfMeasure>	The unit of measurement: ■ Celsius: C ■ Fahrenheit: F ■ Kelvin: K

Details

SCPI compliant.

Example

The following program fragment configures the 2182A to use a simulated reference junction (typically an ice bath) and a type K thermocouple with temperature displayed in Fahrenheit.

```
' Configure Temperature:
CALL SEND(7,":SENS:TEMP:TRANS TC",status%)   'Select thermocouple
   'sensor.
CALL SEND(7,":SENS:TEMP:RJUN:RSEL SIM",status%)   'Select simulated reference.
CALL SEND(7,":SENS:TEMP:RJUN:RSEL:SIM 0",status%)   'Set reference to 0°C.
CALL SEND(7,":SENS:TEMP:TC K",status%)   'Set for type K thermocouple.
CALL SEND(7,":UNIT:TEMP F",status%)   'Read in °F.
```

Also see

Measuring voltage and temperature (on page 4-14)

Common commands

In this section:

Introduction

The common commands are device commands that are common to all devices on the bus. These commands are designated and defined by the IEEE-488.2 standard.

*CLS

This command clears the event registers and queues.

Type	Affected by	Where saved	Default value
Command only	Not applicable	Not applicable	Not applicable

Usage

```
*CLS
```

Details

Use the `*CLS` command to clear (reset to 0) the bits of the following registers in the 2182A:

- Standard Event Register
- Operation Event Register
- Error Queue
- Questionable Event Register
- Measurement Event Register

This command also forces the instrument into the operation complete command idle state and operation complete query idle state.

Also see

Status model (on page 16-1)

*ESE

This command sets and queries bits in the Status Enable register of the Standard Event Register.

Type	Affected by	Where saved	Default value
Command and query	Not applicable	Not applicable	See **Details**

Usage

```
*ESE <NRf>
*ESE?
```

`<NRf>`	The value of the Status Enable register: ■ `0`: Clear register ■ `1`: Set OPC (B0) ■ `4`: Set QYE (B2) ■ `8`: Set DDE (B3) ■ `16`: Set EXE (B4) ■ `32`: Set CME (B5) ■ `64`: Set URQ (B6) ■ `128`: Set PON (B7) ■ `255`: Set all bits

Details

The `*ESE` command programs the standard event enable register. This command is sent with the decimal equivalent of the binary value that determines the state (0 or 1) of the bits in the register. This register is cleared on power up.

This register is used as a mask for the standard event status register. When a standard event is masked, the occurrence of that event does not set the event summary bit (ESB) in the status byte register. Conversely, when a standard event is unmasked (enabled), the occurrence of that event sets the ESB bit.

A cleared bit (`0`) in the enable register prevents (masks) the ESB bit in the status byte register from setting when the corresponding standard event occurs. A set bit (`1`) in the enable register enables the ESB bit to set when the corresponding standard event occurs.

The standard event enable register is shown in the following table and includes the decimal weight of each bit. The sum of the decimal weights of the bits to be set is the parameter value that is sent with the `*ESE` command.

If a command error (CME) occurs, bit B5 of the standard event status register sets. If a query error (QYE) occurs, bit B2 of the standard event status register sets. Since both events are unmasked (enabled), the occurrence of any one of them causes the ESB bit in the status byte register to set.

The standard event status register can be read by using the *ESE? query command.

Standard event enable register

Bit position	B7	B6	B5	B4	B3	B2	B1	B0
Event	PON	URQ	CME	EXE	DDE	QYE	—	OPC
Decimal Weighting	128 (2^7)	64 (2^6)	32 (2^5)	16 (2^4)	8 (2^3)	4 (2^2)	—	1 (2^0)
Value	0/1	0/1	0/1	0/1	0/1	0/1	—	0/1

Bits B8 through B15 are not shown since they are not used.

Values are:

- 1 = Enable Standard Event
- 0 = Disable (Mask) Standard Event

Events are:

- PON = Power On
- URQ = User Request
- CME = Command Error
- EXE = Execution Error
- DDE = Device-dependent Error
- QYE = Query Error
- OPC = Operation Complete

Example

`*ESE 36`	Sets the CME and QYE bits of the Standard Event Register, where: ▪ CME (bit B5) = Decimal 32 ▪ QYE (bit B2) = Decimal 4 ▪ The Status Enable register is 36

Also see

*CLS (on page 14-1)
Status model (on page 16-1)

*ESR?

This command acquires the value of the Standard Event Register.

Type	Affected by	Where saved	Default value
Query only	Not applicable	Not applicable	Not applicable

Usage

```
*ESR?
```

Details

This command acquires the value of the standard event status register (see the following table). The binary equivalent of the returned decimal value determines which bits in the register are set. This register is cleared on power-up or when `*CLS` is sent.

A set bit in this register indicates that a particular event has occurred. For example, for an acquired decimal value of 48, the binary equivalent is 00110000. For this binary value, bits B4 and B5 of the standard event status register are set. These set bits indicate that a device-dependent error and command error have occurred.

Standard event status register

Bit position	B7	B6	B5	B4	B3	B2	B1	B0
Event	PON	URQ	CME	EXE	DDE	QYE	—	OPC
Decimal Weighting	128 (2^7)	64 (2^6)	32 (2^5)	16 (2^4)	8 (2^3)	4 (2^2)	—	1 (2^0)
Value	0/1	0/1	0/1	0/1	0/1	0/1	—	0/1

Bits B8 through B15 are not shown since they are not used.

Values are:

- 1 = Event Bit Set
- 0 = Event Bit Cleared

Events are:

- PON = Power On
- URQ = User Request
- CME = Command Error
- EXE = Execution Error
- DDE = Device-dependent Error
- QYE = Query Error
- OPC = Operation Complete

The bits of the standard event status register are described as follows:

Bit B0, operation complete (OPC): Set bit indicates that all pending selected device operations are completed and the 2182A is ready to accept new commands; this bit only sets in response to the `*OPC?` query command.

Bit B1: Not Used

Bit B2, query error (QYE): Set bit indicates that you attempted to read data from an empty output queue.

Bit B3, device-dependent error (DDE): Set bit indicates that an instrument operation did not execute properly due to some internal condition (such as Settings Conflict errors, and buffer sizing and mismatch errors).

Bit B4, execution error (EXE): Set bit indicates that the 2182A detected an error when trying to execute a command.

Bit B5, command error (CME): Set bit indicates that a command error has occurred. Command errors include:

- IEEE-488.2 syntax error: 2182A received a message that does not follow the defined syntax of the IEEE-488.2 standard.
- Semantic error: 2182A received a command that was misspelled or received an optional IEEE-488.2 command that is not implemented.
- The instrument received a group execute trigger (GET) inside a program message.

Bit B6, user request (URQ): Set bit indicates that the LOCAL key on the 2182A front panel was pressed.

Bit B7, power on (PON): Set bit indicates that the 2182A was turned off and turned back on since the last time this register was read.

Example

```
SYST:PRES    ' Return 2182A to default setup.
INIT:CONT OFF    ' Disables continuous initiation.
ABORt     ' Aborts operation. Places 2182A in idle.
INIT:IMM    ' Initiate one trigger cycle.
*OPC    ' Sends the OPC command
*ESR?    ' Reads the Standard Event Status Register.
ABORt    ' Aborts operation. Places 2182A in idle.
*ESR?    ' Reads the Standard Event Status Register.
SYST:PRES    ' Returns 2182A to default setup.
```

The first group of commands send the `*OPC` command after the `:INITiate` command and verifies that the OPC bit in the Standard Event Status Register does not set while the instrument continues to make measurements (not in idle). The second group of commands returns the 2182A to the idle state and verifies that the OPC bit did set.

After addressing the 2182A to talk, the returned value of 0 denotes that the bit (bit 0) is not set, indicating that the `:INITiate` operation is not complete.

After addressing the 2182A to talk, the returned value of 1 denotes that the bit (bit 1) is set, indicating that the `:INITiate` operation is complete.

Also see

None

*IDN?

This command retrieves the identification string of the instrument.

Type	Affected by	Where saved	Default value
Query only	Not applicable	Not applicable	Not applicable

Usage

```
*IDN?
```

Details

The identification string includes the manufacturer, model number, serial number, and firmware revision of the instrument. The string is formatted as follows:

```
KEITHLEY INSTRUMENTS,MODEL nnnn,xxxxxxxx,yyyyyy/zzzzz
```

Where:

- *nnnn* is the model number
- *xxxxxxxx* is the serial number
- *yyyyyy/zzzzz* is the firmware revision levels of the digital board ROM and display board ROM

Example

`*IDN?`	Output: `KEITHLEY INSTRUMENTS,MODEL 2182A,01234567/1.0.0i`

Also see

None

*OPC

This command sets the operation complete (OPC) bit after all pending commands, including overlapped commands, have been executed.

Type	Affected by	Where saved	Default value
Command only	Not applicable	Not applicable	Not applicable

Usage

```
*OPC
```

Details

After the *OPC command is sent, the Operation Complete bit (bit B0) of the Standard Event Status Register sets immediately after the last pending command is completed. If the corresponding bit (bit B0) in the Standard Event Enable Register and bit 5 (Event Summary Bit) of the Service Request Enable Register is set, the RQS/MSS (Request for Service/Master Summary Status) bit in the Status Byte Register is set.

When used with the initiate immediately command (:INITiate), the OPC bit in the Standard Event Status Register is not set until the 2182A returns to the idle state. The :INIT command operation is not considered to be finished until the 2182A returns to the idle state. See the description of *WAI for more information on command execution.

When used with the *TRG command, the OPC bit is not set until the operations associated with the *TRG command (and the initiate command) are finished. The *TRG command is considered to be finished when the device action block completes or when operation stops a control source to wait for an event.

To use the *OPC exclusively with the *TRG command, first force the completion of the initiate command so that only the *TRG command is pending. To do this, send the :ABORt command to place the instrument in idle, which completes the initiate command. If continuous initiation is on, operation continues into the trigger model. After sending the *TRG command, the OPC bit sets when the *TRG command is finished.

NOTE

For RS-232 operation, use *OPC or *OPC? with commands that respond slowly, such as *RCL, *RST, *SAV, and SYST:PRES, and when performing the standard deviation calculation on a large buffer, CALC2:IMM, and CALC2:IMM?.

Example

```
SYST:PRES   ' Return 2182A to default setup.
INIT:CONT OFF   ' Disables continuous initiation.
ABORt    ' Aborts operation. Places 2182A in idle.
INIT:IMM   ' Initiate one trigger cycle.
*OPC   ' Sends the OPC command
*ESR?   ' Reads the Standard Event Status Register.

ABORt   ' Aborts operation. Places 2182A in idle.
*ESR?   ' Reads the Standard Event Status Register.

SYST:PRES   ' Returns 2182A to default setup.
```

The first group of commands send the `*OPC` command after the `:INITiate` command and verifies that the OPC bit in the Standard Event Status Register does not set while the instrument continues to make measurements (not in idle). The second group of commands returns the 2182A to the idle state and verifies that the OPC bit did set.

After addressing the 2182A to talk, the returned value of 0 denotes that the bit (bit 0) is not set, indicating that the `:INITiate` operation is not complete.

After addressing the 2182A to talk, the returned value of 1 denotes that the bit (bit 1) is set, indicating that the `:INITiate` operation is complete.

Also see

*WAI (on page 14-16)
:INITiate[:IMMediate] (on page 13-95)
Triggering (on page 9-1)

*OPC?

This command places a `1` in the output queue after all pending operations are completed.

Type	Affected by	Where saved	Default value
Query only	Not applicable	Not applicable	Not applicable

Usage

```
*OPC?
```

Details

When the 2182A is addressed to talk, the `1` in the Output Queue is sent to the computer.

The `1` in the Output Queue sets the Message Available (MAV) bit (B4) of the Status Byte Register. If the corresponding bit (B4) in the Service Request Enable Register is set, the Request for Service/Master Summary Status (RQS/MSS) bit in the Status Byte Register is set. The execution of `OPC?` is not completed until it has placed the `1` in the Output Queue.

If `*OPC?` is used with the initiate immediately command (`:INITiate`), the `1` is not placed into the Output Queue until the 2182A returns to the idle state. The `:INIT` command operation is not considered finished until the 2182A goes back into the idle state. See the description of `*WAI` for more information on command execution.

To use *OPC? exclusively with the *TRG command, force the completion of the initiate command so that only the *TRG command is pending. To do this, send the :ABORt command to place the instrument in idle, which completes the initiate command. If continuous initiation is on, operation continues into the trigger model. After sending the *TRG command, an ASCII 1 is placed in the Output Queue and the MAV bit sets when the *TRG command is finished.

After *OPC? executes, you cannot send additional commands to the 2182A until the pending overlapped commands are finished. For example, :INITiate:CONTinuous ON followed by *OPC? locks up the instrument and requires a device clear (DCL or SDC) before it accepts any more commands.

The following commands take a long time to process and may benefit from using *OPC or OPC?:

- *RST
- :SYST:PRES
- *RCL
- *SAV
- :CALC2:IMM and :CALC2:IMM? when performing the standard deviation calculation on a large buffer

RS-232 operation can also benefit from *OPC?.

Example

```
SYST:PRES      ' Returns 2182A to default setup.
INIT:CONT OFF  ' Disables continuous initiation.
ABORt          ' Aborts operation. Places 2182A in idle.
TRIG:COUN 1    ' These two commands configure the 2182A
SAMP:COUN 5    ' to make five measurements.
INIT           ' Starts measurement process.
*OPC?          ' Sends the OPC? command
```

Uses *OPC? to signal the end of a measurement process.
After the measurements are made and the instrument returns to the idle state, an ASCII 1 is placed in the Output Queue. The 2182A is addressed to talk, which sends the 1 from the Output Queue to the computer. Displays the 1 on the monitor.

Also see

:INITiate[:IMMediate] (on page 13-95)
*OPC (on page 14-7)
*TRG (on page 14-15)
*WAI (on page 14-16)

*RCL

This command returns to the setup stored in memory.

Type	Affected by	Where saved	Default value
Command only	Not applicable	Not applicable	Not applicable

Usage

```
*RCL 0
```

Details

This command returns the 2182A to a setup configuration stored at a memory location. The `*SAV` command stores a setup configuration at a memory location.

The 2182A is shipped from the factory with `:SYSTem:PRESet` defaults loaded into the available setup memories. If a recall error occurs, the setup memories default to the `:SYSTem:PRESet` values.

Only one setup configuration can be saved and recalled.

NOTE

For RS-232 operation and in some cases, GPIB operation, use `*OPC` or `*OPC?` with `*RCL`, which is a slow responding command.

Also see

:SYSTem:PRESet (on page 13-88)
*OPC (on page 14-7)
*OPC? (on page 14-8)
*SAV (on page 14-11)

*RST

This command resets the instrument settings to their default values.

Type	Affected by	Where saved	Default value
Command only	Not applicable	Not applicable	Not applicable

Usage

```
*RST
```

Details

Returns the instrument to default settings, cancels all pending commands, and cancels the response to any previously received `*OPC` and `*OPC?` commands.

NOTE

For RS-232 operation (and in some cases, GPIB operation), use `*OPC` or `*OPC?` with `*RST`, which is a command that responds slowly.

Also see

*OPC (on page 14-7)
*OPC? (on page 14-8)

*SAV

This command saves the present setup into memory.

Type	Affected by	Where saved	Default value
Command only	Not applicable	Not applicable	Not applicable

Usage

```
*SAV 0
```

Details

Any control that is affected by `*RST` can be saved by the `*SAV` command. Use the `*RCL` command to restore the instrument to the saved setup configuration.

Only one setup configuration can be saved and recalled.

NOTE

For RS-232 operation and in some GPIB operations, use `*OPC` or `*OPC?` with `*SAV`, which is a slow responding command.

Also see

*OPC (on page 14-7)
*OPC? (on page 14-8)
*RCL (on page 14-10)
:SYSTem:POSetup (on page 13-87)

*SRE

These commands program and read the service request enable register.

Type	Affected by	Where saved	Default value
Command and query	Not applicable	Not applicable	See Details

Usage

```
*SRE <NRf>
*SRE?
```

<NRf>	0: Clears enable register 1: Set MSB bit (bit 0) 4: Set EAV bit (bit 2) 8: Set QSB bit (bit 3) 16: Set MAV bit (bit 4) 32: Set ESB bit (bit 5) 128: Set OSB bit (bit 7) 255: Sets all bits

Details

The *SRE command programs the service request enable register. This command is sent with the decimal equivalent of the binary value that determines the state (0 or 1) of each bit in the register. This register is cleared on power-up.

This enable register is used with the status byte register to generate service requests (SRQ). With a bit in the service request enable register set, an SRQ occurs when the corresponding bit in the status byte register is set by an appropriate event. For details on register structure, refer to Status model (on page 16-1).

The service request enable register is shown in the following table. The sum of the decimal weights of the bits to set is the value that is sent with the *SRE command.

You can read the contents of the service request enable register using the *SRE? query command.

Service request enable register

Bit position	**B7**	**B6**	**B5**	**B4**	**B3**	**B2**	**B1**	**B0**
Event	OSB	—	ESB	MAV	QSB	EAV	—	MSB
Decimal Weighting	128 (2^7)	—	32 (2^5)	16 (2^4)	8 (2^3)	4 (2^2)	—	1 (2^0)
Value	0/1	—	0/1	0/1	0/1	0/1	—	0/1

Values are:

- 1 = Enable Service Request Event
- 0 = Disable (Mask) Service Request Event

Events are:

- OSB = Operation Summary Bit
- ESB = Event Summary Bit
- MAV = Message Available
- QSB = Questionable Summary Bit
- EAV = Error Available
- MSB = Measurement Summary Bit

Example

`*SRE 48`
Set the ESB and MAV bits of the service request enable register, where: ▪ ESB (bit B5) = `32` ▪ MAV (bit B4) = `16` ▪ `<NRf> = 48`

Also see

None

*STB?

This command reads the status byte register.

Type	Affected by	Where saved	Default value
Query only	Not applicable	Not applicable	Not applicable

Usage

`*STB?`

Details

The `*STB?` query command acquires the value of the status byte register. The status byte register is shown in the following table. The binary equivalent of the decimal value determines which bits in the register are set.

All bits except bit 6 in this register are set by other event registers and queues. Bit 6 sets when one or more enabled conditions occur.

The `*STB?` query command does not clear the status byte register. This register can only be cleared by clearing the related registers and queues. Register and queue structure are explained in Status structure (on page 16-1).

For example, for an acquired decimal value of 48, the binary equivalent is 00110000. This binary value indicates that bits 4 and 5 of the status byte register are set.

Status bye register

Bit position	B7	B6	B5	B4	B3	B2	B1	B0
Event	OSB	MSS, RQS	ESB	MAV	QSB	EAV	—	MSB
Decimal Weighting	128 (2^7)	64 (2^6)	32 (2^5)	16 (2^4)	8 (2^3)	4 (2^2)	—	1 (2^0)
Value	0/1	0/1	0/1	0/1	0/1	0/1	—	0/1

Values are:

- 1 = Event Bit Set
- 0 = Event Bit Cleared

Events are:

- OSB = Operation Summary Bit
- MSS = Master Summary Status
- RQS = Request Service
- ESB = Event Summary Bit
- MAV = Message Available
- QSB = Questionable Summary Bit
- EAV = Error Available
- MSB = Measurement Summary Bit

The bits of the status byte register are described as follows:

Bit 0, measurement status (MSB): Set bit indicates that a measurement event has occurred. The event can be identified by reading the measurement event status register using the `:STATus:MEASurement?` command (see SCPI command reference (on page 13-1)).

Bit 1: Not used.

Bit 2, error available (EAV): Set bit indicates that an error or status message is present in the error queue. The message can be read using one of the following SCPI commands (see SCPI command reference (on page 13-1)):

`:SYSTem:ERRor?`

`:STATus:QUEue?`

Bit 3, questionable summary bit (QSB): Set bit indicates that a calibration error has occurred.

Bit 4, message available (MAV): Set bit indicates that a message is present in the output queue. The message is sent to the computer when the 2182A is addressed to talk.

Bit 5, event summary bit (ESB): Set bit indicates that an enabled standard event has occurred. The event can be identified by reading the standard event status register using the *ESE? query command.

Bit 6, master summary status (MSS) / request service (RQS): Set bit indicates that one or more enabled status byte conditions have occurred. The MSS bit can be read using the STB? query command, or the occurrence of a service request (RQS bit set) can be detected by performing a Serial Poll.

Bit 7, operation summary bit (OSB): Set bit indicates that an enabled operation event has occurred. The event can be identified by reading the Operation Event Status Register using the `:STATus:OPERation?` query command (see SCPI command reference (on page 13-1) for details).

Also see

None

*TRG

This command sends a bus trigger to the 2182A.

Type	Affected by	Where saved	Default value
Command only	Not applicable	Not applicable	Not applicable

Usage

```
*TRG
```

Details

Use the `*TRG` command to issue a GPIB trigger to the 2182A. It has the same effect as a group execute trigger (GET).

Use the `*TRG` command as an event to control operation. The 2182A reacts to this trigger if BUS is the programmed control source. The control source is programmed from the `TRIGger` subsystem.

Also see

Triggering (on page 9-1)

*TST?

This command runs self test and reads the result.

Type	Affected by	Where saved	Default value
Query only	Not applicable	Not applicable	Not applicable

Usage

```
*TST?
```

Details

This query command performs a checksum test on ROM and places the coded result (0 or 1) in the output queue. When the 2182A is addressed to talk, the coded result is sent from the output queue to the computer.

A returned value of zero (0) indicates that the test passed. A value of one (1) indicates that the test has failed.

Also see

None

*WAI

This command postpones the execution of subsequent commands until all previous overlapped commands are finished.

Type	Affected by	Where saved	Default value
Command only	Not applicable	Not applicable	Not applicable

Usage

```
*WAI
```

Details

There are two types of instrument commands:

- **Overlapped commands:** Commands that allow the execution of subsequent commands while instrument operations of the overlapped command are still in progress.
- **Sequential commands:** Commands whose operations must finish before the next command is executed.

The *WAI command suspends the execution of commands until the instrument operations of all previous overlapped commands are finished. The *WAI command is not needed for sequential commands.

The 2182A has three overlapped commands:

- :INITiate
- :INITiate:CONTinuous ON
- *TRG

The :INITiate commands remove the 2182A from the idle state. The device operations of :INITiate do not complete until the 2182A returns to idle. You can send the *WAI command after the :INITiate command to wait until the 2182A returns to idle before executing subsequent commands.

The *TRG command issues a bus trigger that can be used to provide the arm, scan, and measure events for the trigger model. By sending the *WAI command after the *TRG command, subsequent commands execute when the trigger model moves in response to *TRG and settles at its next state.

Example

```
CALL SEND (7, ":SYST:PRES", status%)
CALL SEND (7, ":INIT:CONT OFF; :abort", status%)   'Place 2182A in idle.
CALL SEND (7, ":TRIG:COUN 1;SOUR TIM, status%)
CALL SEND (7, ":SAMP:COUN 30", status%)   'Program for 30
   'measurements then stop
CALL SEND (7, ":INIT; *WAI", status%)   'Start measurements and send
   '*WAI
CALL SEND (7, ":DATA?", status%)   'Query a reading.
reading$ = SPACE$(80)
CALL ENTER (reading$, length%, 7, status%)   'Get a response when 2182A
   'goes into idle.
PRINT reading$   'Display the reading.
```

Also see

*OPC (on page 14-7)
*OPC? (on page 14-8)
*TRG (on page 14-15)

Section 15

Program examples

In this section:

Introduction

All examples presume QuickBASIC version 4.5 or higher and a CEC IEEE-488 interface card with CEC driver version 2.11 or higher, with the 2182A at address 7 on the IEEE-488 bus.

Changing function and range

The 2182A has independent range control for each of its two voltage measurement functions. This means, for example, that autorange can be turned on for DCV1 while leaving it off for DCV2.

Another difference is in the relationship of parameters to the range command. In other instruments, a single number was used to denote each range. The parameter of the SCPI `RANGe` command is given as "the maximum value to measure." The instrument interprets this parameter and goes to the appropriate range. When you query the range with `RANGe?`, the instrument returns the full-scale value of its present range.

The following example program illustrates changing the function and range. It sets the range for several functions and then makes readings on each of those functions.

The 2182A rounds the range parameter to an integer before choosing the appropriate range. For example, sending `SENSe:VOLTage:CHANnel1:RANGe 20.45` sets channel 1 of the 2182A to the 100 V range.

```
'Example program to demonstrate changing voltage function and range,
'taking readings on DCV1 and DCV2.
'For QuickBASIC 4.5 and CEC PC488 interface card
'Edit the following line to where the QuickBASIC
'libraries are on your computer.
'$INCLUDE: 'c:'b45\ieeeqb.bi'
'Initialize the CEC interface as address 21.
CALL initialize(21, 0)
'Reset the SENSe1 subsystem settings and the trigger
'model. Each READ? causes one trigger.
CALL SEND("*rst", status%)
'Set DCV1 for 100V range and DCV2 for 10 V range.
CALL SEND(7, "sens:volt:chan1:rang 100"status%)
CALL SEND(7, "sens:volt:chan2:rang 10", status%)
'Switch to DCV2 (Channel 2 volts) and make a reading.
CALL SEND(7, "sens:func 'volt'", status%)
CALL SEND(7, "sens:chan 2; :read?", status%)
reading%$ = SPACE$(80)
CALL ENTER(reading$, length%, 7, status%)
PRINT reading$
'Switch to DCV1 (Channel 1 volts) and make reading.
CALL SEND(7, "sens:func 'volt'", status%)
CALL SEND(7, "sens:chan 1; :read?", status%)
reading%$ = SPACE$(80)
CALL ENTER(reading$, length%, 7, status%
PRINT reading$
```

One-shot triggering

Other voltmeters generally have two types of triggering: One-shot and continuous. In one-shot, each activation of the selected trigger source triggers one reading. In continuous, the voltmeter is idle until the trigger source is activated, at which time it begins making readings at a specified rate. Typical trigger sources are:

- IEEE-488 talk
- IEEE-488 Group Execute Trigger (GET)
- "X" command
- External trigger (BNC on the rear panel)

Arming the instrument to respond to triggers is implicit in non-SCPI voltmeters. Sending a command to a non-SCPI voltmeter to change any of the trigger controls causes the instrument to arm itself for triggers.

The SCPI trigger model implemented in the 2182A gives:

- Explicit control over the trigger source (the TRIGger subsystem).
- A way to complete disable triggers.

Changing any of the settings in the TRIGger subsystem does not automatically arm the 2182A for triggers.

The following program sets up the 2182A to make one reading each time it receives an external trigger pulse.

```
'Example program to demonstrate one-shot external triggering.
'For QuickBASIC 4.5 and CEC PC488 interface card.
'Edit the following line to where the QuickBASIC
'libraries are on your computer.
'$INCLUDE: 'c:'b45\ieeeqb.bi'
'Initialize the CEC interface as address 21.
CALL initialize(21, 0)
'Reset controls and put trigger model in IDLE state.
CALL SEND(7, "*rst", status%)
CALL SEND(7, "trig:sour ext;coun inf", status%)
'Start everything.
CALL SEND(7, "init", status%)
```

After the 2182A receives the INITiate command, it stops at the control source in the trigger model, waiting for a trigger pulse. Each time a pulse arrives at the Trigger Link connector, the 2182A makes one reading. Because TRIGger:COUNt is set to INFinity, the instrument never enters the idle state. You can send the ABORt command to put the instrument in the idle state, disabling triggers until another INITiate command is sent.

Generating SRQ on buffer full

When your program must wait until the 2182A has completed an operation, it is more efficient to program the 2182A to assert the IEEE-488 SRQ line when it is finished instead of sending repeated serial polls to the instrument. An IEEE-488 controller typically addresses the instrument to talk and then unaddresses it each time it performs a serial poll. Repeated polling of the 2182A generally reduces its overall reading throughput.

The 2182A provides a status bit for most operations it performs. It can be programmed to assert the IEEE-488 SRQ line whenever a status bit becomes true or false. The IEEE-488 controller (your computer) can examine the state of the SRQ line without performing a serial poll, thereby detecting when the 2182A has completed its task without interrupting it in the process.

The following example program segment sets up the 2182A to assert SRQ when the reading buffer has completely filled and then arms the reading buffer, initiates readings, and waits for the 2182A to indicate that the buffer is full.

The following example code is not a complete program. The commands to configure the trigger model and the reading buffer, shown in the example Storing readings in buffer (on page 15-4), are not shown. The example shown here can be modified for any event in the 2182A status reporting system.

```
'Reset STATus subsystem (not affected by *RST).
CALL SEND(7, "stat:pres;*cls", status%)
CALL SEND(7, "stat:meas:enab 512", status%)   'Enable BFL.
CALL SEND(7, "*sre 1"' status%)   'Enable MSB.
CALL SEND(7, "trac:feed:cont next", status%)
'Start everything.
CALL SEND(7, "init", status%)
WaitSRQ:
IF (NOT(srq%()) THEN GOTO WaitSRQ
CALL SPOLL(7, poll%, status%)
IF (poll% AND 64)=0 THEN GOTO WaitSRQ
```

After the program has detected an asserted SRQ line, it serial polls the 2182A to determine if it is the device requesting service. This is necessary for the following reasons:

- Serial polling the 2182A causes it to stop asserting the SRQ line.
- In test systems that have more than one IEEE-488 instrument programmed to assert SRQ, your program must determine which instrument is actually requesting service.

Once an event register has caused a service request, it cannot cause another service request until you clear it by reading it (in the above code example, using `STATus:MEASurement[:EVENt]?`) or by sending the `*CLS` command.

Storing readings in the buffer

The command in the 2182A that control the reading buffer include:

- The size of the buffer (in readings).
  `TRACe:POINts <NRf>`
- Where the data is coming from (before or after the CALCulate1 math post-processing).
  `TRACe:FEED SENSe1` store unprocessed readings.
  `TRACe:FEED CALCulate1` store math processed readings.
- Select buffer control mode.
  `TRACe:FEED:CONTrol NEVer` to immediately stop storing readings.
- `TRACe:FEED:CONTrol NEXT` to start now and stop when buffer is full.

The following example program sets up the 2182A to make 20 readings as fast as it can into the buffer, and then reads the data after the buffer is filled.

```
'Example program to demonstrate the reading buffer.
'For QuickBASIC 4.5 and CEC PC488 interface card.
'Edit the following line to where the QuickBASIC
'libraries are on your computer.
'$INCLUDE: 'c:'b45\ieeeqb.bi'
'Initialize the CEC interface as address 21.
CALL initialize(21, 0)
'Reset controls and put trigger model in IDLE state.
CALL SEND(7, "*rst", status%)
'Reset STATus subsystem (not affected by *RST).
CALL SEND(7, "stat:pres;*cls", status%)
CALL SEND(7, "stat:meas:enab 512", status%)   'Enable BFL.
CALL SEND(7, "*sre 1", status%)   'Enable MSB.
CALL SEND(7, "trig:coun 20", status%)
'TRACe subsystem is not affected by *RST.
CALL SEND(7, "trac:poin 20", status%)
CALL SEND(7, "trac:feed sens1;feed:cont next", status%)
'Start everything.
CALL SEND(7, "init", status%)
'Initialize reading$ while the 2182A is busy making readings.
reading$ = SPACE$(4000)
WaitSRQ:
IF (NOT(srq%)) THEN GOTO WaitSRQ
CALL SPOLL(7, poll%, status%)
IF (poll% AND 64)=0 THEN GOTO WaitSRQ
CALL SEND(7, "stat:meas?", status%)
CALL ENTER(S$, length%, 16, status%)

CALL SEND(7, "form:elem read,unit" status%)
CALL SEND(7, "trac:data?", status%)
CALL ENTER(reading$, length%, 16, status%)
PRINT reading$
'Repeat buffer storage.
CALL SEND(7, "feed:cont next", status%)
CALL SEND(7, "init", status%)
WaitSRQ:
IF (NOT(srq%()) THEN GOTO WaitSRQ
CALL SPOLL(7, poll%, status%)
IF (poll% AND 64)=0 THEN GOTO WaitSRQ
```

Controlling the 2182A using the RS-232 COM2 port

This example program illustrates the use of the 2182A interfaced to the RS-232 COM2 port. The 2182A is set up to make 100 readings at the fastest possible rate (200 readings per second). The readings are made, sent across the serial port, and displayed on the screen.

```
' Example program controlling the 2182A using the RS-232 COM2 port.
' For QuickBASIC 4.5 and CEC PC488 interface card.
RD$=SPACE$(1500)   ' Set string space.
CLS   ' Clear screen.
PRINT "Set COM2 baud rate to 19200"
PRINT "Set no flow control, and CR as Terminator"
' Configure serial port parameters.
ComOpen$="COM2:19200,N,8,1,ASC,CD0,CS0,DS0,LF,OP0,RS,TB8192,RB8192"
OPEN ComOpen$ FOR RANDOM AS #1
' Model 2182A setup commands.
' Note serial communications only operate with SCPI mode.
PRINT #1, "*RST"   ' Clear registers.
PRINT #1, "*CLS"   ' Clear Model 2182A.
PRINT #1, ":INIT:CONT OFF;:ABORT"   ' Init off.
PRINT #1, ":SENS:FUNC 'VOLT:DC"   ' DCV.
PRINT #1, ":SENS:CHAN 1"   ' Channel 1.
PRINT #1, ":SYST:AZER:STAT OFF"   ' Auto zero off.
PRINT #1, ":SENS:VOLT:CHAN1:LPAS:STAT OFF"   ' Analog filter off.
PRINT #1, ":SENS:VOLT:CHAN1:DFIL:STAT OFF"   ' Digital filter off.
PRINT #1, ":SENS:VOLT:DC:NPLC 0.01"   ' NPLC = 0.1.
PRINT #1, ":SENS:VOLT:CHAN1:RANG 10"   ' 10 V range.
PRINT #1, ":SENS:VOLT:DC:DIG 4"   ' 4 digit.
PRINT #1, ":FORM:ELEM READ"   ' Reading only.
PRINT #1, ":TRIG:COUN 1"   ' Trig count 1.
PRINT #1, ":SAMP:COUN 100"   ' Sample count 100.
PRINT #1, ":TRIG:DEL 0"   ' No trigger dela.y
PRINT #1, ":TRIG:SOUR IMM"   ' Immediate trigger.
PRINT #1, ":DISP:ENAB OFF"   ' No display.
PRINT #1, ":INIT"   ' Send init.
SLEEP 1   ' Wait one second.
PRINT #1, ":FETCH?"   ' Read query.
LINE INPUT #1, RD$   ' Get data.
PRINT RD$   ' Display data.
PRINT #1, ":DISP:ENAB ON"   ' Turn on display.
PRINT #1, ":SYST:AZER:STAT ON"   ' Auto zero on.
'Clean up and quit.
finish:
CLOSE #1   ' Close file.
CLEAR   ' Interface clear.
END
```

Status model

In this section:

Status structure

The status model consists of status register sets and queues. You can monitor the status model to view instrument events and configure the status model to control the events.

As you work with the status model, be aware that the result applies to the Status Byte Register. All the status register sets and queues flow into the Status Byte Register. Your test program can read this register to determine if a service request (SRQ) has occurred, and if so, which event caused it.

The Status Byte Register, register sets, and queues include:

- Standard Event Register
- Questionable Event Register
- Operation Event Register
- Output Queue
- Error Queue

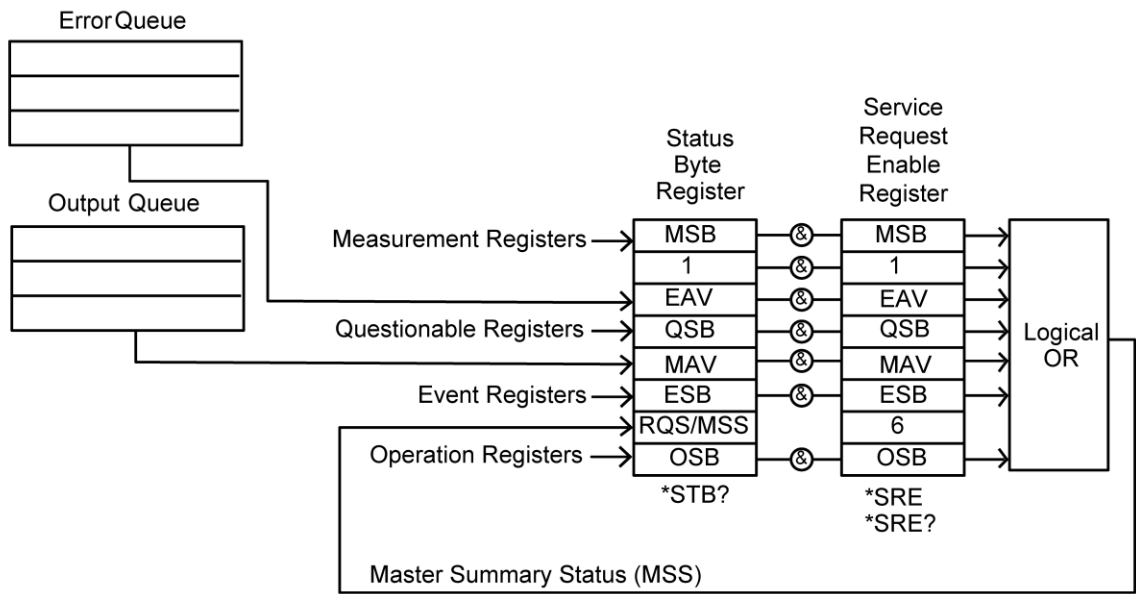
The following figures show the structure of the status model.

NOTE

The status structures registers are configured and controlled by STATus subsystem commands (on page 13-66), *ESE (on page 14-2), *ESR? (on page 14-4), *SRE (on page 14-12), and *STB (on page 14-13).

Status byte register overview

Figure 89: Status byte register

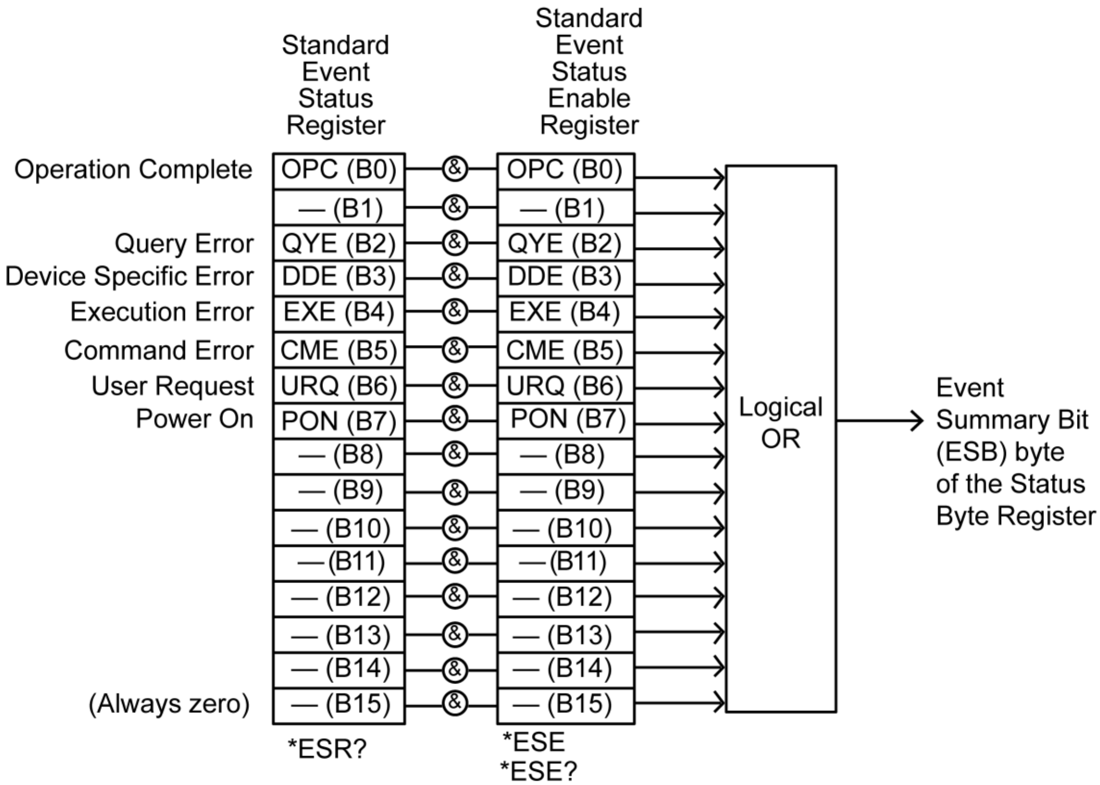


Abbreviation	Description
MSB	Measurement Summary Bit
EAV	Error Available
QSB	Questionable Summary Bit
MAV	Message Available
ESB	Event Summary Bit
RQS/MSS	Request for Service/Master Summary Status
OSB	Operation Summary Bit

NOTE

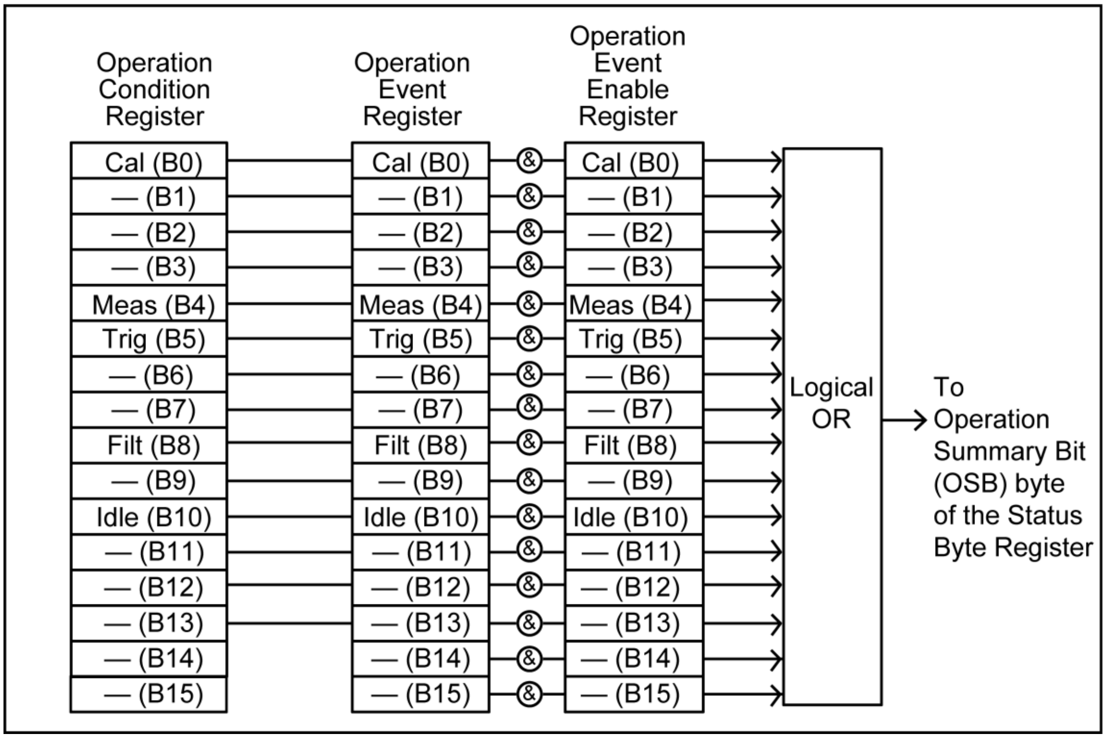
The RQS bit is in the serial poll byte. The MSS bit is in the `*STB?` response.

Figure 90: Standard event status registers



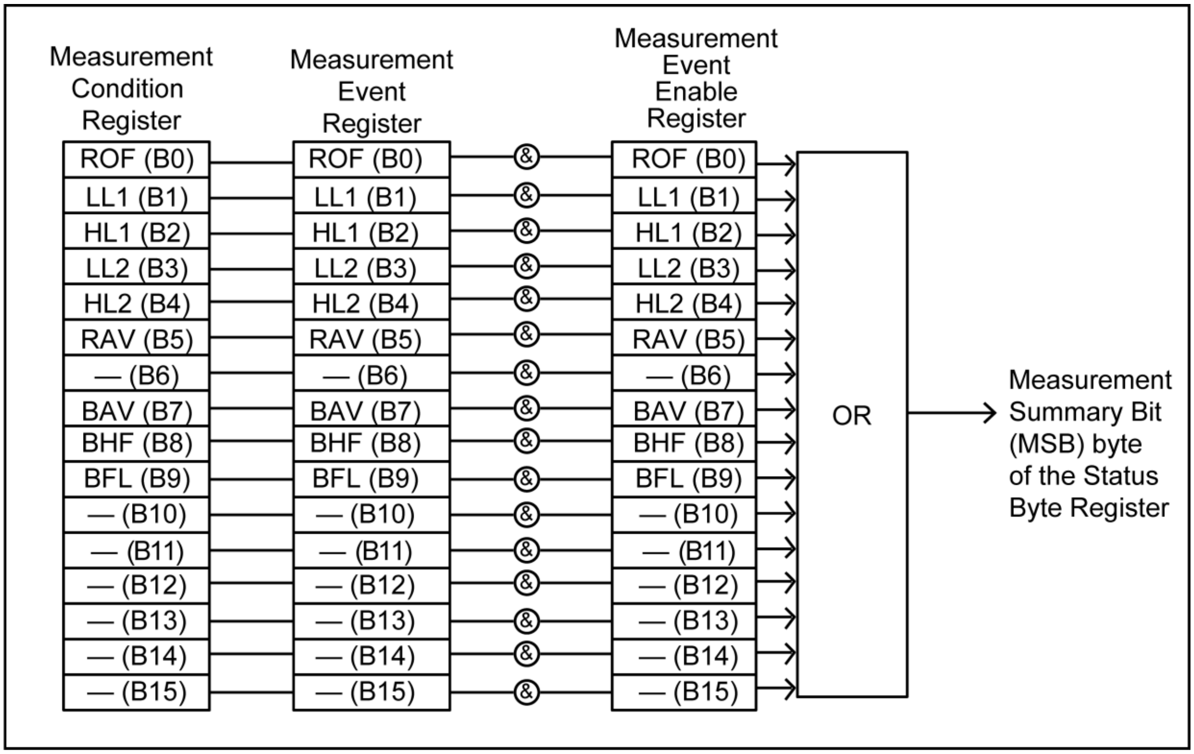
PON	Power on	&	Logical AND
URQ	User request	OR	Logical OR
CME	Command error		
EXE	Execution error		
DDE	Device-dependent error		
QYE	Query error		
OPC	Operation complete		

Figure 91: Operation event status registers



Cal	Calibrating	&	Logical AND
Meas	Measuring	OR	Logical OR
Trig	Trigger layer		
Filt	Filter settled		
Idle	Idle state of the 2182A		

Figure 92: Measurement event status registers

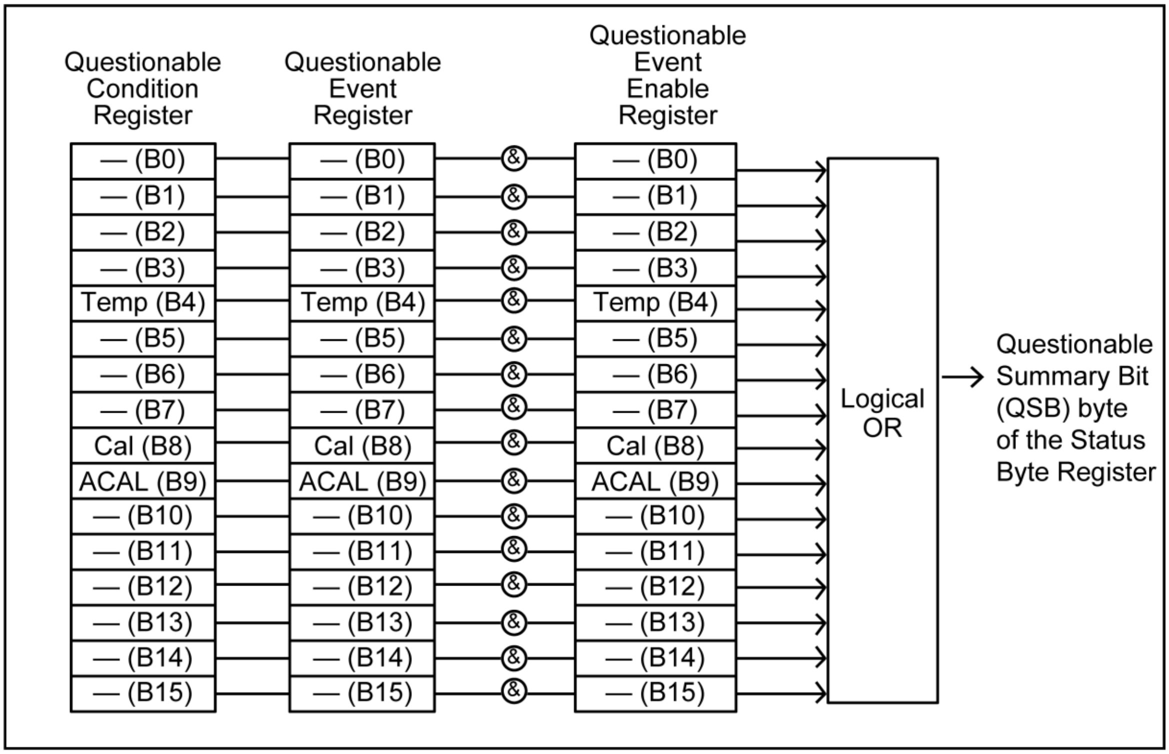


ROF	Reading overflow	&	Logical AND
LL1	Low limit 1	OR	Logical OR
HL1	High limit 1		
LL2	Low limit 2		
HL2	High limit 2		
RAV	Reading available		
BAV	Buffer available		
BHF	Buffer half full		
BFL	Buffer full		

NOTE

B15 is always zero.

Figure 93: Questionable status registers



Temp	Temperature summary	&	Logical AND
Cal	Calibration summary	OR	Logical OR
ACAL	ACAL summary		

Queues

The 2182A queues are first-in, first-out (FIFO) registers. The Output Queue holds reading and response messages. The Error Queue holds error and status messages.

Refer to Status structure (on page 16-1) for a figure that shows how the queues are structured with the other registers.

The output queue holds data that pertains to the normal operation of the instrument. For example, when a query command is sent, the response message is placed in the output queue.

When data is placed in the output queue, the Message Available (MAV) bit in the Status Byte Register is set. A data message is cleared from the output queue when it is read. The output queue is considered cleared when it is empty. An empty output queue clears the MAV bit in the Status Byte Register.

Read a message from the output queue by addressing the 2182A to talk after the appropriate query is sent.

The error queue holds error and status messages. When an error or status event occurs, a message that defines the error or status is placed in the error queue. This queue holds up to ten messages.

When a message is placed in the error queue, the Error Available (EAV) bit in the Status Byte Register is set. An error message is cleared from the error queue when it is read. The error queue is considered cleared when it is empty. An empty error queue clears the EAV bit in the Status Byte Register. To read an error message from the error queue, send either of the following SCPI query commands and then address the 2182A to talk:

- `:SYSTem:ERRor?`
- `:STATus:QUEue?`

Condition registers

Some register sets have a condition register. A condition register is a real-time, read-only register that is constantly updated to reflect the present operating conditions of the instrument. For example, while a measurement is made, bit B4 (Meas) of the Operation Condition Register is set. When the measurement is completed, bit B4 clears.

Use the `:CONDition?` query commands in the `STATus` subsystem to read the condition registers. See STATus subsystem (on page 13-66) for more information on these commands.

Event registers

Each status register set has an event register. An event register is a latched, read-only register whose bits are set by the corresponding condition register. Once a bit in an event register is set, it remains set (latched) until the register is cleared by a specific clearing operation. The bits of an event register are logically ANDed with the bits of the corresponding enable register and applied to an OR gate. The output of the OR gate is applied to the Status Byte Register.

Use the `*ESR?` command to read the Standard Event Register. All other event registers are read using the `EVENt?` query commands in the `STATus` subsystem. See *ESR? (on page 14-4) and STATus subsystem (on page 13-66) for more information.

An event register is cleared when it is read. The following operations clear all event registers:

- Cycling power
- Sending `*CLS`

Enable registers

Each status register set has an enable register, as shown in the figures in Status structure (on page 16-1). An enable register is programmed by you and serves as a mask for the corresponding event register. An event bit is masked when the corresponding bit in the enable register is cleared (`0`). When masked, a set bit in an event register cannot set a bit in the Status Byte Register (1 AND 0 = 0).

To use the Status Byte Register to detect events (serial poll), you must unmask the events by setting the appropriate bits of the enable registers to `1`.

To program and query the Standard Event Status Register, use the `*ESE` and `*ESE?`. All other enable registers are programmed and queried using the `:ENABle` and `:ENABle?` commands in the `STATus` subsystem. See *ESE (on page 14-2) and STATus subsystem (on page 13-66) for more information.

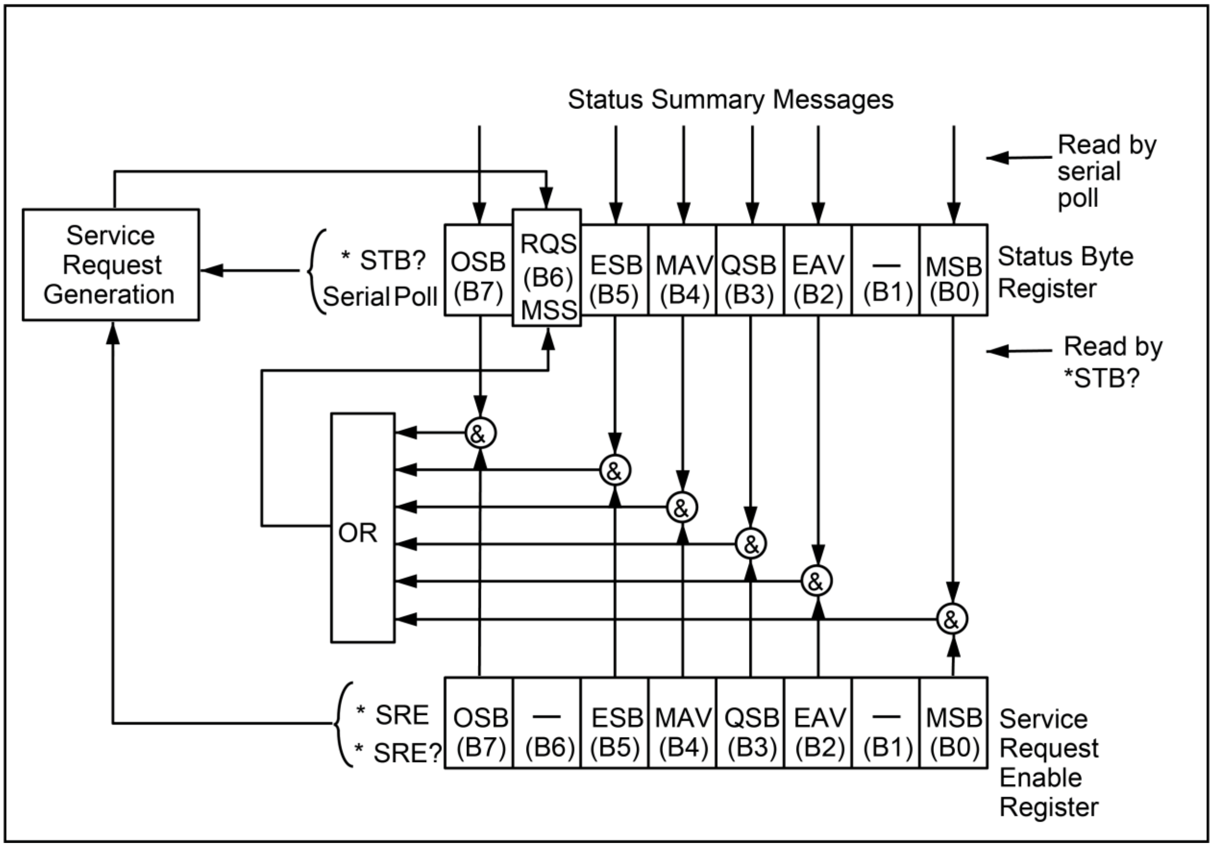
An enable register is not cleared when it is read. The following operations affect the enable registers:

- Cycling power: Clears all enable registers
- `:STATus:PRESet` clears the following enable registers:
  - Operation Event Enable Register
  - Questionable Event Enable Register
  - Measurement Event Enable Register
- `*ESE 0`: Clears the Standard Event Status Enable Register

Status byte and service request (SRQ)

Service request is controlled by the 8-bit registers Status Byte Register and Service Request Enable Register. The following figure shows the structure of these registers.

Figure 94: Status byte and service request



Status byte and service request definitions

OSB	Operation Summary bit	&	Logical AND
MSS	Master Summary Status	OR	Logical OR
RQS	Request for Service		
ESB	Event Summary bit		
MAV	Message available		
QSB	Questionable Summary bit		
EAV	Error available		
MSB	Measurement Summary bit		

Status byte register

The summary messages from the status registers and queues set or clear the appropriate bits (B0, B2, B3, B4, B5, and B7) of the Status Byte Register. These bits do not latch, and their states (0 or 1) are solely dependent on the summary messages (0 or 1). For example, if the Standard Event Status Register is read, its register clears. As a result, its summary message resets to 0, which clears the ESB bit in the Status Byte Register.

Bit B6 in the Status Byte Register is one of the following:

- The Master Summary Status (MSS) bit, sent in response to the `*STB?` command, indicates the status of any set bits with corresponding enable bits set.
- The Request for Service (RQS) bit, sent in response to a serial poll, indicates which device was requesting service by polling on the SRQ line.

For a description of the other bits in the Status Byte Register, see Common commands (on page 14-1).

The IEEE-488.2 standard uses the `*STB?` common query command to read the Status Byte Register.

When reading the Status Byte Register using the `*STB?` command, bit B6 is called the MSS bit. None of the bits in the Status Byte Register are cleared when using the `*STB?` command to read it.

The IEEE-488.1 standard has a serial poll sequence that also reads the Status Byte Register and is better suited to detecting a service request (SRQ). When using the serial poll, bit B6 is called the RQS bit. Serial polling causes bit B6 (RQS) to reset. Refer to Serial poll and SRQ (on page 16-11).

The following operations clear all bits of the Status Byte Register:

- Cycling power.
- Sending the `*CLS` common command

The MAV bit may or may not be cleared.

Service request enable register

This register is programmed by you and serves as a mask for the Status Summary Message bits (B0, B2, B3, B4, B5, and B7) of the Status Byte Register. When masked, a set summary bit in the Status Byte Register cannot set bit B6 (MSS/RQS) of the Status Byte Register. Conversely, when unmasked, a set summary bit in the Status Byte Register sets bit B6.

A Status Summary Message bit in the Status Byte Register is masked when the corresponding bit in the Service Request Enable Register is cleared (0). When the masked summary bit in the Status Byte Register sets, it is ANDed with the corresponding cleared bit in the Service Request Enable Register. The logic 1 output of the AND gate is applied to the input of the OR gate, so it sets the MSS/RQS bit in the Status Byte Register.

The individual bits of the Service Request Enable Register can be set or cleared by using the `*SRE <NRf>` common command.

To read the Service Request Enable Register, use the `*SRE?` query command. The Service Request Enable Register clears when power is cycled or when `*SRE 0` is sent.

Serial poll and SRQ

Any enabled event summary bit that goes from 0 to 1 sets RQS and generates a service request (SRQ). In your test program, you can periodically read the Status Byte Register to check if a service request (SRQ) occurred and what caused it. If an SRQ occurred, the program can, for example, branch to an appropriate subroutine that services the request. Typically, service requests (SRQs) are managed by the serial poll sequence of the 2182A. If an SRQ did not occur, bit B6 (RQS) of the Status Byte Register remains cleared, and the program proceeds normally after the serial poll is performed. If an SRQ did occur, bit B6 of the Status Byte Register sets, and the program can branch to a service subroutine when the SRQ is detected by the serial poll.

The serial poll automatically resets RQS of the Status Byte Register. This allows subsequent serial polls to monitor bit B6 for an SRQ occurrence generated by other event types. After a serial poll, the same event can cause another SRQ, even if the event register that caused the first SRQ has not been cleared.

A serial poll clears RQS but does not clear MSS. The MSS bit stays set until all Status Byte event summary bits are cleared.

Section 17

Model 182 emulation commands

In this section:

Introduction

You can configure the 2182A to accept the device-dependent (DDC) commands of the Keithley Instruments Model 182 Sensitive Digital Voltmeter. The commands for controlling the 2182A with the 182 language are provided in the following topic. For details on 182 operation, refer to the *Model 182 Instruction Manual*.

Since the architecture of the 2182A differs from that of the Model 182, some commands cannot be used. Be sure to refer to the notes for information on command restrictions.

CAUTION

The 182 language is intended to be used only over the IEEE-488 bus. Using front-panel controls with this language may cause erratic operation and results cannot be guaranteed.

NOTE

When switching from the 182 language to the SCPI language, the instrument goes into the idle state and stays there. You can take the instrument out of idle by pressing the TRIG key or by sending an initiate command.

182 device-dependent command summary

Display ASCII String

Commands:

- `A0`: Restore display to normal.
- `A1,string`: Display string (up to 12 characters).

The maximum number of characters for the A1 command string is 12. The `A2` and `A3` commands are not supported.

Display Resolution

Commands:

- `B0`: 5½-digit resolution.
- `B1`: 6½-digit resolution.
- `B2`: 3½-digit resolution.
- `B3`: 4½-digit resolution.

Calibration

`C` (Calibration) commands are not supported by the 182 language. You must use the SCPI language to calibrate the 2182A over the bus.

Filter Damping

Commands:

`D0`: Configure filter damping off (same as `P2`).

`D1`: Configure filter damping on (same as `P3`).

Reading Source

Commands:

- `F0`: Latest reading from A/D converter.
- `F1`: One reading from buffer.
- `F2`: All readings in buffer.
- `F3`: Maximum value in buffer.
- `F4`: Minimum value in buffer.

Reading Format

Commands:

- `G0`: Reading only 3.
- `G1`: Reading with prefix.
- `G2`: Reading with buffer location.
- `G3`: Reading with buffer location and prefix.

The `G4` through `G7` commands are not supported by the 182 language. The 182 does not use timestamps.

Immediate Trigger

Command:

- `H0`: Initiate manual trigger 4.

The H1 command is not supported by the 182 language.

Buffer Configuration

Commands:

- `I0`: Disable buffer 5.
- `I1,value`: Buffer on, value = buffer size.

The minimum buffer size for the 2182A is 2, while the minimum buffer size for the 182 is 1. Therefore, to set minimum buffer size, `I1,0` is valid for the 182 and `I1,2` is valid for the 2182A. The `I2` command (circular buffer) is not supported.

Analog Output Relative Offset

Commands:

- `J0`: Disable analog output relative offset.
- `J1`: Enable analog output relative using last reading as the relative value (the `J1` command for the 182 uses the next reading as the relative value).
- `J2,value`: Enable analog output relative offset using value (±1e−9 to ±120).
- `J3`: Enable analog output relative offset and use the present relative offset value.

EOI, Bus Hold-off

Commands:

- `K0`: Enable EOI, enable bus hold-off on X.
- `K1`: Disable EOI, enable bus hold-off on X.
- `K2`: Enable EOI, disable bus hold-off on X.
- `K3`: Disable EOI, disable bus hold-off on X.

Save/Recall Setup

Commands:

- `L0`: Save current setup as power-on.
- `L1`: Recall factory default setup.
- `L2`: Recall power-on setup.

SRQ Mask

Commands:

- `M0`: Disable SRQ.
- `M1`: Reading done.
- `M2`: Buffer half full.
- `M4`: Buffer full.
- `M8`: Reading overflow.
- `M16`: Ready for command.
- `M32`: Error.
- `M128`: Ready for trigger.

Enable/Disable Filter

Commands:

- `N0`: Enable Filter.
- `N1`: Disable Filter.

Analog Filter Configuration

Commands:

- `O0`: Configure analog filter off.
- `O1`: Configure analog filter on.

Digital Filter Configuration

Commands:

- `P0`: Configure digital filter off.
- `P1`: Configure fast response.
- `P2`: Configure medium response.
- `P3`: Configure slow response.

Trigger Interval

Command:

- `Qvalue`: Interval = value in ms (10 ms to 999999 ms).

Range

Commands:

- `R0`: Enable autorange.
- `R1`: 10 mV range.
- `R2`: 100 mV range.
- `R3`: 1 V range.
- `R4`: 10 V range.
- `R5`: 100 V range.
- `R6`: No operation.
- `R7`: No operation.
- `R8`: Disable autorange.

Integration Period

Commands:

- `S0`: Line cycle integration period.
- `S1`: 3 ms integration period.
- `S2`: 100 ms integration period.
- `S3`: 1 s integration period. The `S3` command is added to include a 1 s integration period.

Trigger Mode

Commands:

- `T0`: Multiple on talk.
- `T1`: One-shot on talk.
- `T2`: Multiple on GET.
- `T3`: One-shot on GET.
- `T4`: Multiple on X.
- `T5`: One-shot on X.
- `T6`: Multiple on external.
- `T7`: One-shot on external.
- `T8`: Multiple on manual (TRIG key) or bus `H0X`.
- `T9`: One-shot on manual (TRIG key) or bus `H0X`.
- `T10`: Disable all triggers.

The front-panel TRIG key is always operational, even when `T10` is sent to disable all triggers.

Alternate Output

Commands:

- `U0`: Send machine status.
- `U1`: Send error conditions. For the 2182A, the `U1` error status word only supports the IDDC (Invalid Device-Dependent Command) error (bit 0), the IDDCO (Invalid Device-Dependent Command Option) error (bit 2), and the Uncalibrated Error (bit 8). All other bits remain at zero.
- `U2`: Send firmware revision. The 2182A uses two processors, while the Model 182 uses three processors. Therefore, the `U2` command is formatted to only provide revision levels for the Front-Panel Processor and the Main Processor.
- `U3`: Send buffer length.
- `U4`: Send buffer average.
- `U5`: Send buffer standard deviation.
- `U6`: Send reading relative value.
- `U7`: Send analog output relative value.
- `U8`: Send analog output gain value.
- `U9`: Send trigger interval.
- `U10`: Send trigger delay.
- `U12`: Send calibration lock status.
- `U13`: Send Model 181-like machine status.

`U11` and `U14` are not supported.

Analog Output

Command:

- `V0,gain`: Analog output gain (0.001 to 999999.999)

The `V1` command is not supported.

Trigger Delay

Commands:

- `W0`: Disable trigger delay.
- `Wvalue`: Enable trigger delay; delay = value (1 ms to 999999 ms)

Execute

Commands:

- `X`: Execute other device-dependent commands.

Terminators

Commands:

- `Y0`: <CR LF>.
- `Y1`: <LF CR>.
- `Y2`: <CR>.
- `Y3`: <LF>.
- `Y10`: <CR LF>.
- `Y13`: <LF CR>.

Reading Relative

Commands:

- `Z0`: Disable reading relative.
- `Z1`: Enable reading relative using next reading. The 2182A uses the last reading as the relative value. The `Z1` command for the 182 uses the next reading as the relative value.
- `Z2,value`: Enable reading relative using value (±1e-9 to ±120).
- `Z3`: Enable reading relative, use present relative value.

Section 18

IEEE-488 bus overview

In this section:

Introduction

The IEEE-488 bus is a communication system between two or more electronic devices. A device can be either an instrument or a computer. When a computer is used on the bus, it serves to supervise the communication exchange between all the devices and is known as the controller. Supervision by the controller consists of determining which device talks and which device listens. As a talker, a device outputs information and as a listener, a device receives information. To simplify the task of keeping track of the devices, a unique address number is assigned to each one.

On the bus, only one device can talk at a time and is addressed to talk by the controller. The device that is talking is known as the active talker. The devices that need to listen to the talker are addressed to listen by the controller. Each listener is then referred to as an active listener. Devices that do not need to listen are instructed to unlisten. The reason for the unlisten instruction is to optimize the speed of bus information transfer since the task of listening takes up bus time. Through the use of control lines, a handshake sequence takes place in the transfer process of information from a talker to a listener. This handshake sequence helps ensure the credibility of the information transfer. The basic handshake sequence between an active controller (talker) and a listener is as follows:

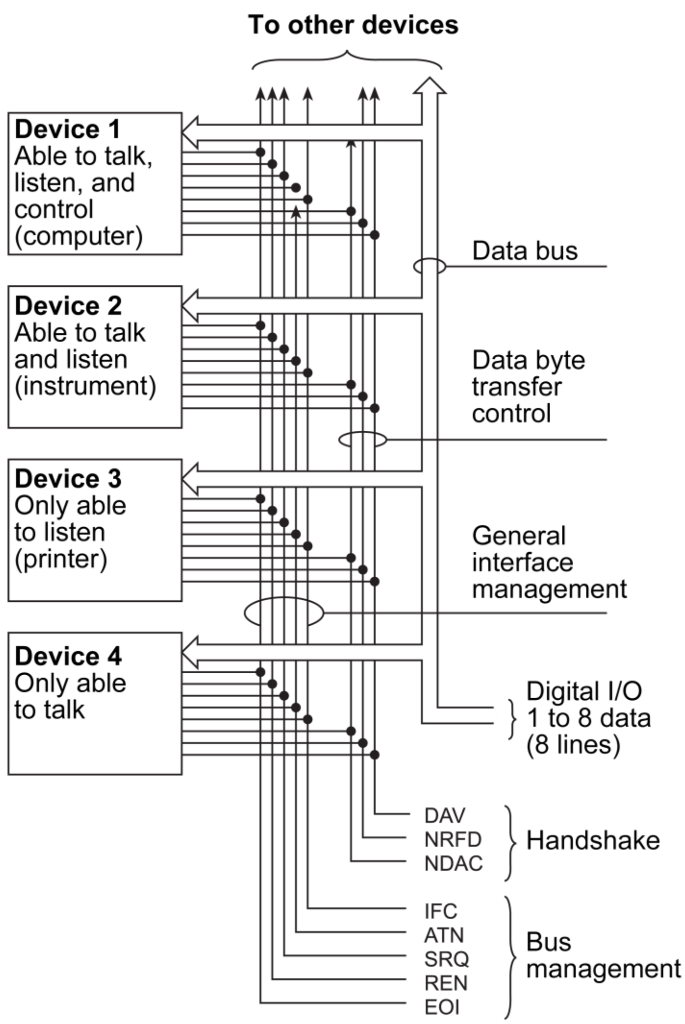
1. The listener indicates that it is ready to listen.
2. The talker places the byte of data on the bus and indicates that the data is available to the listener.
3. The listener, aware that the data is available, accepts the data and indicates that the data has been accepted.
4. The talker, aware that the data has been accepted, stops sending data and indicates that data is not being sent.
5. The listener, aware that there is no data on the bus, indicates that it is ready for the next byte of data.

Bus description

The IEEE-488 bus, which is also referred to a General Purpose Interface Bus (GPIB), is a parallel transfer medium that optimizes data transfer without using an excessive number of bus lines. In keeping with this goal, the bus has only eight data lines that are used for both data and with most commands. Five bus management lines and three handshake lines round out the complement of bus signal lines.

A typical setup for controlled operation is shown in the following figure. Generally, a system contains one controller and a number of other instruments to which the commands are given. Device operation is categorized into three operators: Controller, talker, and listener. The controller controls the instruments on the bus. The talker sends data and a listener receives data. Depending on the type of instrument, any particular device can be a talker only, a listener only, or both a talker and listener.

Figure 95: IEEE-488 bus configuration



There are system controllers and basic controllers. Both can control other instruments, but only the system controller has absolute authority in the system. In a system with more than one controller, only one controller may be active at any given time. Certain protocol is used to pass control from one controller to another.

The IEEE-488 bus is limited to 15 devices, including the controller. Therefore, any number of talkers and listeners up to that limit may be present on the bus at one time. Although several devices may be commanded to listen simultaneously, the bus can have only one active talker.

A device is placed in the talk or listen state by sending an appropriate talk or listen command. These talk and listen commands are derived from the primary address of the instrument. The primary address may have any value between 0 and 31. The actual listen address value sent out over the bus is obtained by ORing the primary address with `$20`. For example, if the primary address is `$27`, the actual listen address is `$47` (`$47` = `$27` + `$2`0). The talk address is obtained by ORing the primary address with `$40`. With the present example, the talk address derived from a primary address of `27` decimal is `$67` (`$67` = `$27` + `$40`).

The IEEE-488 standards also include another addressing mode called secondary addressing. Secondary addresses are in the range of `$60` to `$7F`. The 2182A does not use secondary addressing.

Once a device is addressed to talk or listen, the appropriate bus transactions take place. For example, if the instrument is addressed to talk, it places its data string on the bus one byte at a time. The controller reads the information and the appropriate software can be used to direct the information to the correct location.

Bus lines

The signal lines on the IEEE-488 bus are grouped into the categories data lines, management lines, and handshake lines. The data lines handle bus data and commands. The management and handshake lines ensure that proper data transfer and operation takes place. Each bus line is active low, with approximately zero volts representing a logic 1 (true). The following paragraphs describe the operation of these lines.

Data lines

The IEEE-488 bus uses eight data lines that transfer data one byte at a time. DIO1 (data input/output) through DIO8 (data input/output) are the eight bidirectional data lines used to transmit both data and multiline commands. The data lines operate with low true logic.

Bus management lines

The bus management lines help to ensure proper interface control and management. These lines send the uniline commands.

ATN (attention): The ATN line is one of the more important management lines because the state of this line determines how information on the data bus is interpreted.

IFC (interface clear): The IFC line controls clearing of instruments from the bus.

REN (remote enable):The REN line places the instrument on the bus in the remote mode.

EOI (end or identify): The EOI is usually used to mark the end of a multibyte data transfer sequence.

SRQ (service request): This line is used by devices when they require service from the controller.

Handshake lines

The bus handshake lines operate in an interlocked sequence. This method ensures reliable data transmission regardless of the transfer rate. Generally, data transfer occurs at a rate determined by the slowest active device on the bus.

One of the three handshake lines is controlled by the source (the talker sending information). The other two lines are controlled by accepting devices (the listener or listeners receiving the information). The three handshake lines are:

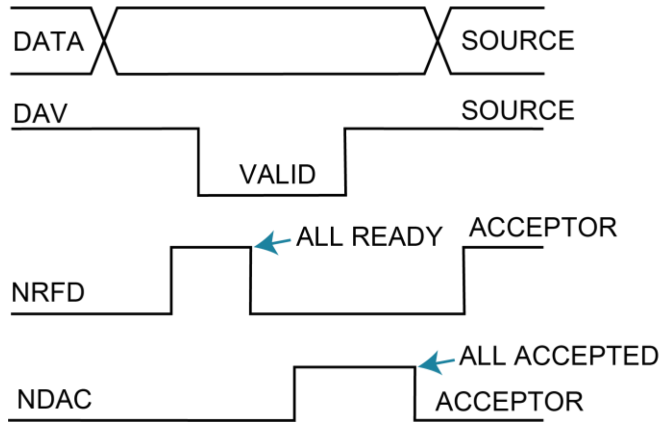
- DAV (data valid): The source controls the state of the DAV line, which indicates whether data bus information is valid for any listening lines.
- NRFD (not ready for data): The acceptor controls the state of NRFD. It signals the transmitting device to pause the byte transfer sequence until the accepting device is ready.
- NDAC (not data accepted): NDAC is also controlled by the accepting device. The state of NDAC tells the source whether or not the device has accepted the data byte.

The complete handshake sequence for one data byte is shown in the following figure. Once data is placed on the data lines, the source checks to see that NRFD is high, indicating that all active devices are ready. At the same time, NDAC should be low from the previous byte transfer. If these conditions are not met, the source must wait until NDAC and NRFD have the correct status. If the source is a controller, NRFD and NDAC must be stable for at least 100 ns after ATN is set true. Because of the possibility of a bus interruption, many controllers have time-out routines that display messages if case the transfer sequence stops for any reason.

Once all NDAC and NRFD are properly set, the source sets DAV low, signaling to accepting devices that the byte on the data lines is now valid. NRFD then goes low, and NDAC goes high once all devices have accepted the data. Each device releases NDAC at its own rate, but NDAC is not released to go high until all devices have accepted the data byte.

The sequence described above is used to transfer data, talk and listen addresses, and multiline commands. The state of the ATN line determines what the data bus contains.

Figure 96: IEEE-488 handshake sequence



Bus commands

The instrument may be given a number of special bus commands through the IEEE-488 interface. This section briefly describes the purpose of the bus commands, which are grouped into the following categories:

- Uniline commands: Sent by setting the associated bus lines true. For example, to assert `REN` (remote enable), the `REN` line is set low (true).
- Multiline commands: General bus commands that are sent over the data lines with the `ATN` line true (low).
- Common commands: Commands that are common to all devices on the bus; sent with `ATN` high (false).
- `SCPI` commands: Commands that are particular to each device on the bus; sent with `ATN` (false).

These bus commands and their general purposes are summarized in the following table.

IEEE-488 bus command summary

Command type	Command	State of ATN line	Comments
Uniline	`REN` (remote enable)	X	Set up devices for remote operation
	`EOI`	X	Marks end of transmission
	`IFC` (interface clear)	X	Clears interface
	`ATN` (Attention)	Low	Defines data bus contents
	`SRQ`	X	Controlled by external device
Multiline Universal	`LLO` (local lockout)	Low	Locks out local operation
	`DCL` (device clear)	Low	Returns device to default conditions
	`SPE` (serial enable)	Low	Enables serial polling
	`SPD` (serial poll disable)	Low	Disables serial polling
Addressed	`SDC` (selective device clear)	Low	Returns instrument to default conditions
	`GTL` (go to local)	Low	Returns device to local

Command type	Command	State of ATN line	Comments
Unaddressed	UNL (unlisten)	Low	Removes all listeners from the bus
	UNT (untalk)	Low	Removes any talkers from the bus
Common	—	High	Programs IEEE-488.2 compatible instruments for common operations
SCPI	—	High	Programs SCPI compatible instruments for specific operations

Uniline commands

ATN, IFC, and REN are asserted only by the controller. SRQ is asserted by an external device. EOI may be asserted either by the controller or other devices, depending on the direction of data transfer. The following is a description of each command. Each command is sent by setting the corresponding bus line true.

REN (remote enable): REN is sent to set up instruments on the bus for remote operation. When REN is true, devices are removed from the local mode. Depending on device configuration, all front-panel controls except the LOCAL key may be locked out when REN is true. Generally, send REN before attempting to program instruments over the bus.

EOI (end or identify): EOI positively identifies the last byte in a multibyte transfer sequence, allowing data words of various lengths to be transmitted easily.

IFC (interface clear): IFC clears the interface and returns all devices to the talker and listener idle states.

ATN (attention): The controller sends ATN when transmitting addresses or multiline commands.

SRQ (service request): SRQ is asserted by a device when it requires service from a controller.

Universal multiline commands

Universal commands are multiline commands that require no addressing. All devices equipped to implement such commands do so simultaneously when the commands are transmitted. All multiline commands are transmitted with ATN true.

LLO (local lockout): LLO is sent to the instrument to lock out the LOCAL key and all front-panel controls of the instrument.

DCL (device clear): DCL returns instruments to a default state. Usually, instruments return to their power-up conditions.

SPE (serial poll enable): SPE is the first step in the serial polling sequence that determines which device has requested service.

SPD (serial poll disable): SPD is used by the controller to remove all devices on the bus from the serial poll mode and is generally the last command in the serial polling sequence.

Addressed multiline commands

Addressed commands are multiline commands that must be preceded by the device listen address before that instrument responds to the command. Only the addressed device responds to these commands. Both the commands and the address preceding it are sent with ATN true.

SDC (selective device clear): The SDC command performs essentially the same function as the DCL command, except that only the addressed device responds. Generally, instruments return to their power-up default conditions when responding to the SDC command.

GTL (go to local): The GTL command removes instruments from the remote mode. With some instruments, GTL also unlocks front-panel controls if they were previously locked out with the LLO command.

GET (group execute trigger): The GET command triggers devices to perform a specific action that depends on device configuration (for example, make a reading). Although GET is an addressed command, many devices respond to GET without addressing.

Addressed commands

Addressed commands include two primary command groups and a secondary address group. ATN is true when these commands are asserted. The commands include:

- LAG (listen address group): These listen commands are derived from the primary address of the instrument and address devices to listen. The actual command byte is obtained by ORing the primary address with $20.
- TAG (talk address group): The talk commands are derived from the primary address by ORing the address with $40. Talk commands address devices to talk.
- SCG (secondary command group): Commands in this group provide additional addressing capabilities. Many devices (including the 2182A) do not use these commands.

Unaddressed commands

The controller uses the unaddressed commands to remove any talkers or listeners from the bus. ATN is true when these commands are asserted.

UNL (unlisten): Listeners are placed in the listener idle state by the UNL command.

UNT (untalk): Any previously commanded talkers are placed in the talker idle state by the UNT command.

Common commands

Common commands are commands that are common to all devices on the bus. These commands are designated and defined by the IEEE-488.2 standard.

Generally, these commands are sent as one or more ASCII characters that tell the device to perform a common operation, such as reset. The IEEE-488 bus treats these commands as data because ATN is false when the commands are transmitted.

SCPI commands

SCPI commands are commands that are particular to each device on the bus. These commands are designated by the instrument manufacturer and are based on the instrument model defined by the Standard Commands for Programmable Instruments (SCPI) Consortium's SCPI standard.

Generally, these commands are sent as one or more ASCII characters that tell the device to perform a particular operation, such as setting a range or closing a relay. The IEEE-488 bus treats these commands as data because ATN is false when the commands are transmitted.

Command codes

Command codes for the various commands that use the data lines are summarized in the following figure. Hexadecimal and the decimal values for the various commands are listed in Hexadecimal and decimal values for the common commands (on page 18-9).

Figure 97: Command codes

D_7 D_6 D_5 D_4					X 0 0 0	Command	X 0 0 1	Command	X 0 1 0	Primary Address	X 0 1 1	Primary Address	X 1 0 0	Primary Address	X 1 0 1	Primary Address	X 1 1 0		X 1 1 1	
Bits D_3 ↓	D_2 ↓	D_1 ↓	D_0 ↓	Column → Row ↓	0 (A)	0 (B)	1 (A)	1 (B)	2 (A)	2 (B)	3 (A)	3 (B)	4 (A)	4 (B)	5 (A)	5 (B)	6 (A)	6 (B)	7 (A)	7 (B)
0	0	0	0	0	NUL		DLE		SP	0	0	16	@	0	P	16			p	
0	0	0	1	1	SOH	GTL	DC1	LLO	!	1	1	17	A	1	Q	17	a		q	
0	0	1	0	2	STX		DC2		"	2	2	18	B	2	R	18	b		r	
0	0	1	1	3	ETX		DC3		#	3	3	19	C	3	S	19	c		s	
0	1	0	0	4	EOT	SDC	DC4	DCL	$	4	4	20	D	4	T	20	d		t	
0	1	0	1	5	ENQ	PPC*	NAK	PPU*	%	5	5	21	E	5	U	21	e		u	
0	1	1	0	6	ACK		SYN		&	6	6	22	F	6	V	22	f		v	
0	1	1	1	7	BEL		ETB		'	7	7	23	G	7	W	23	g		w	
1	0	0	0	8	BS	GET	CAN	SPE	(	8	8	24	H	8	X	24	h		x	
1	0	0	1	9	HT	TCT*	EM	SPD	)	9	9	25	I	9	Y	25	i		y	
1	0	1	0	10	LF		SUB		•	10	:	26	J	10	Z	26	j		z	
1	0	1	1	11	VT		ESC		+	11	;	27	K	11	[	27	k		{	
1	1	0	0	12	FF		FS		,	12	<	28	L	12	\	28	l		:	
1	1	0	1	13	CR		GS		_	13	=	29	M	13	]	29	m		}	
1	1	1	0	14	SO		RS		.	14	>	30	N	14	^	30	n		≅	
1	1	1	1	15	SI		US		/	15	?	UNL	O	15	_	UNT	o		DEL	

ADDRESSED COMMAND GROUP (ACG): columns 0 (A)–0 (B)
UNIVERSAL COMMAND GROUP (UCG): columns 1 (A)–1 (B)
LISTEN ADDRESS GROUP (LAG): columns 2 (A)–3 (B)
TALK ADDRESS GROUP (TAG): columns 4 (A)–5 (B)
PRIMARY COMMAND GROUP (PCG): columns 0 (A)–5 (B)
SECONDARY COMMAND GROUP (SDC): columns 6 (A)–7 (B)

*PPC (PARALLEL POLL CONFIGURE) PPU (PARALLEL POLL UNCONFIGURE), and TCT (TAKE CONTROL) not implemented by Model 6517B.
Note: D_0 = D101 ...D_7 = D108; X = Don't Care.

Hexadecimal and decimal values for the common commands

Hexadecimal and the decimal values for the common commands are listed in the following tables.

Command	Hex value	Decimal value
GTL	01	1
SDC	04	4
GET	08	8
LLO	11	17
DCL	14	20
SPE	18	24
SPD	19	25
LAG	20-3F	32-63
TAG	40-5F	64-95
SCG	60-7F	96-127
UNL	3F	63
UNT	5F	95

Typical command sequences

For the multiline commands, a specific bus sequence must take place to properly send the command. In particular, the correct listen address must be sent to the instrument before it responds to addressed commands. The following table lists a typical bus sequence for sending the addressed multiline commands. In this instance, the `SDC` command is sent to the instrument. `UNL` is generally sent as part of the sequence to ensure that no other active listeners are present. `ATN` is true for both the listen command and the `SDC` command byte.

Typical addressed command sequence

			Data bus		
Step	**Command**	**ATN state**	**ASCII**	**Hex**	**Decimal**
1	`UNL`	Set low	?	3F	63
2	`LAG*`	Stays low	;	3B	59
3	`SDC`	Stays low	EOT	04	4
4		Returns high			
*Assumes primary address = 27					

The following table gives a typical common command sequence. In this instance, `ATN` is true while the instrument is being addressed, but it is set high while sending the common command string.

Typical common command sequence					
			Data bus		
Step	**Command**	**ATN state**	**ASCII**	**Hex**	**Decimal**
1	`UNL`	Set low	?	3F	63
2	`LAG*`	Stays low	;	3B	59
3	`Data`	Set high	*	2A	42
4	`Data`	Stays high	R	52	82
5	`Data`	Stays high	S	53	83
6	`Data`	Stays high	T	54	84
*Assumes primary address = 27					

IEEE command groups

Command groups supported by the 2182A are listed in the following table. Common commands and SCPI commands are not included in this list.

IEEE command groups

Handshake command group	
	NDAC = Not data accepted NRFD = Not ready for data DAV = Data valid
Universal command group	
	ATN = Attention DCL = Device clear IFC = Interface clear REN = Remote enable SPD = Serial poll disable SPE = Serial poll enable
Address command group	
LISTEN	LAG = Listen address group MLA = My listen address UNL = Unlisten
TALK	TAG = Talk address group MTA = My talk address UNT = Untalk OTA = Other talk address
Addressed command group	
	ACG = Addressed command group GTL = Go to local SDC = Selective device clear
Status command group	
	RQS = Request service SRQ = Serial poll request STB = Status byte EOI = End

Interface function codes

The interface function codes, which are part of the IEEE-488 standards, define the ability of an instrument to support interface functions and should not be confused with programming commands found elsewhere in this manual. The interface function codes for the 2182A are listed in the following table.

2182A interface function codes

Code	Interface function
SH1	Source Handshake capability
AH1	Acceptor Handshake capability
T5	Talker (basic talker, talk-only, serial poll, unaddressed to talk on LAG)
L4	Listener (basic listener, unaddressed to listen on TAG)
SR1	Service Request capability
RL1	Remote/Local capability
PP0	No Parallel Poll capability
DC1	Device Clear capability
DT1	Device Trigger capability
C0	No Controller capability
E1	Open collector bus drivers
TE0	No Extended Talker capability
LE0	No Extended Listener capability

The codes define 2182A capabilities as follows:

- **SH (Source Handshake Function):** SH1 defines the ability of the instrument to initiate the transfer of message/data over the data bus.
- **AH (Acceptor Handshake Function):** AH1 defines the ability of the instrument to guarantee proper reception of message/data transmitted over the data bus.
- **T (Talker Function):** The ability of the instrument to send data over the bus to other devices is provided by the T function. Instrument talker capabilities (T5) exist only after the instrument is addressed to talk.
- **L (Listener Function):** The ability for the instrument to receive device-dependent data over the bus from other devices is provided by the L function. Listener capabilities (L4) of the instrument exist only after it is addressed to listen.
- **SR (Service Request Function):** SR1 defines the ability of the instrument to request service from the controller.
- **RL (Remote-Local Function):** RL1 defines the ability of the instrument to be placed in the remote or local modes.
- **PP (Parallel Poll Function):** The instrument does not have parallel polling capabilities (PP0).
- **DC (Device Clear Function):** DC1 defines the ability of the instrument to be cleared (initialized).
- **DT (Device Trigger Function):** DTI defines the ability of the 2182A to have readings triggered.
- **C (Controller Function):** The instrument does not have controller capabilities (C0).
- **E (Bus Driver Type):** The instrument has open-collector bus drivers (E1).
- **TE (Extended Talker Function):** The instrument does not have extended talker capabilities (TE0).
- **LE (Extended Listener Function):** The instrument does not have extended listener capabilities (LE0).

Keithley Instruments • 28775 Aurora Road • Cleveland, Ohio 44139 • 1-800-833-9200 • tek.com/keithley

KEITHLEY
A Tektronix Company

04/2022