Keysight 34420A Nano Volt/ Micro Ohm Meter

KEYSIGHT TECHNOLOGIES

User's Guide

Notices

Copyright Notice

© Keysight Technologies 1994 - 2021

No part of this manual may be reproduced in any form or by any means (including electronic storage and retrieval or translation into a foreign language) without prior agreement and written consent from Keysight Technologies as governed by United States and international copyright laws.

Trademarks

Microsoft® and WIndows® are U.S. registered trademarks of Microsoft Corporation. All other brand and product names are trademarks of their respective companies.

Manual Part Number

34420-90001

Edition

Edition 5, January 2021

Printed in:

Printed in Malaysia

Published by:

Keysight Technologies
Bayan Lepas Free Industrial Zone,
11900 Penang, Malaysia

Technology Licenses

The hardware and/or software described in this document are furnished under a license and may be used or copied only in accordance with the terms of such license.

Declaration of Conformity

Declarations of Conformity for this product and for other Keysight products may be downloaded from the Web. Go to http://www.keysight.com/go/conformity. You can then search by product number to find the latest Declaration of Conformity.

U.S. Government Rights

The Software is "commercial computer software," as defined by Federal Acquisition Regulation ("FAR") 2.101. Pursuant to FAR 12.212 and 27.405-3 and Department of Defense FAR Supplement ("DFARS") 227.7202, the U.S. government acquires commercial computer software under the same terms by which the software is customarily provided to the public. Accordingly, Keysight provides the Software to U.S. government customers under its standard commercial license, which is embodied in its End User License Agreement (EULA), a copy of which can be found at http://www.keysight.com/find/sweula. The license set forth in the EULA represents the exclusive authority by which the U.S. government may use, modify, distribute, or disclose the Software. The EULA and the license set forth therein, does not require or permit, among other things, that Keysight: (1) Furnish technical information related to commercial computer software or commercial computer software documentation that is not customarily provided to the public; or (2) Relinquish to, or otherwise provide, the government rights in excess of these rights customarily provided to the public to use, modify, reproduce, release, perform, display, or disclose commercial computer software or commercial computer software documentation. No additional government requirements beyond those set forth in the EULA shall apply, except to the extent that those terms, rights, or licenses are explicitly required from all providers of commercial computer software pursuant to the FAR and the DFARS and are set forth specifically in writing elsewhere in the EULA. Keysight shall be under no obligation to update, revise or otherwise modify the Software. With respect to any technical data as defined by FAR 2.101, pursuant to FAR 12.211 and 27.404.2 and DFARS 227.7102, the U.S. government acquires no greater than Limited Rights as defined in FAR 27.401 or DFAR 227.7103-5 (c), as applicable in any technical data.

Warranty

THE MATERIAL CONTAINED IN THIS DOCUMENT IS PROVIDED "AS IS," AND IS SUBJECT TO BEING CHANGED, WITHOUT NOTICE, IN FUTURE EDITIONS. FURTHER, TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, KEYSIGHT DISCLAIMS ALL WARRANTIES, EITHER EXPRESS OR IMPLIED, WITH REGARD TO THIS MANUAL AND ANY INFORMATION CONTAINED HEREIN, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. KEYSIGHT SHALL NOT BE LIABLE FOR ERRORS OR FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES IN CONNECTION WITH THE FURNISHING, USE, OR PERFORMANCE OF THIS DOCUMENT OR OF ANY INFORMATION CONTAINED HEREIN. SHOULD KEYSIGHT AND THE USER HAVE A SEPARATE WRITTEN AGREEMENT WITH WARRANTY TERMS COVERING THE MATERIAL IN THIS DOCUMENT THAT CONFLICT WITH THESE TERMS, THE WARRANTY TERMS IN THE SEPARATE AGREEMENT SHALL CONTROL.

Safety Information

CAUTION

A CAUTION notice denotes a hazard. It calls attention to an operating procedure, practice, or the like that, if not correctly performed or adhered to, could result in damage to the product or loss of important data. Do not proceed beyond a CAUTION notice until the indicated conditions are fully understood and met.

WARNING

A WARNING notice denotes a hazard. It calls attention to an operating procedure, practice, or the like that, if not correctly performed or adhered to, could result in personal injury or death. Do not proceed beyond a WARNING notice until the indicated conditions are fully understood and met.

Certification

Keysight Technologies certifies that this product met its published specifications at the time of shipment from factory. Keysight Technologies further certifies that its calibration measurements are traceable to the United States National Institute of Standards and Technology, to the extend allowed by that organization's calibration facility, and to the calibration facilities of other INternational Standards Organization members.

Safety Symbols

The following symbols on the instrument and in the documentation indicate precautions which must be taken to maintain safe operation of the instrument.

	Earth (ground) terminal		Frame or chassis (ground) terminal

Safety Considerations

Read the information below before using this instrument.

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

WARNING

- Only qualified, service-trained personnel who are aware of the hazards involved should remove the cover from the instrument.
- For continued protection against fire, replace the line fuse only with a fuse of the specified type and rating.
- Do not operate the instrument around flammable gases or fumes, vapor, or wet environments.
- The input voltage range for the instrument is 100 / 120 (127) / 220 (230) / 240 Vac, 45-440 Hz. Mains supply voltage fluctuations are not to exceed ±10% of the nominal supply voltage.

NOTE

Do not install substitute parts or perform any unauthorized modification to the product. Return the product to an Keysight Technologies Sales and Service Office for service and repair to ensure that safety features are maintained.

Environmental Conditions

The 34420A is designed for indoor use and in an area with low condensation. The table below shows the general environmental requirements for this instrument.

Environmental condition	Requirement
Operating temperature	Full accuracy at 0 °C to 55 °C
Operating humidity	Full accuracy to 80% RH at 40°C (non-condensing) Full accuracy to 40% RH for 41°C to 55°C (non-condensing)
Storage temperature	–40 °C to 75 °C
Altitude	Up to 2000 m
Pollution degree	Pollution Degree 2
Overvoltage category	II

Waste Electrical and Electronic Equipment (WEEE) Directive

This instrument complies with the WEEE Directive marking requirement. This affixed product label indicates that you must not discard this electrical or electronic product in domestic household waste.

Product category:

With reference to the equipment types in the WEEE directive Annex 1, this instrument is classified as a “Monitoring and Control Instrument” product.

The affixed product label is as shown below.

Do not dispose in domestic household waste.

To return this unwanted instrument, contact your nearest Keysight Service Center, or visit http://about.keysight.com/en/companyinfo/environment/takeback.shtml for more information.

Sales and Technical Support

To contact Keysight for sales and technical support, refer to the support links on the following Keysight websites:

- www.keysight.com/find/34420A
  (product-specific information and support, software and documentation updates)
- www.keysight.com/find/assist
  (worldwide contact information for repair and service)

The 34420A is a 71/2 digit, high performance nanovolt, micro-ohm meter. Its combination of bench-top and system features makes this meter a versatile solution for your testing requirements now and in the future.

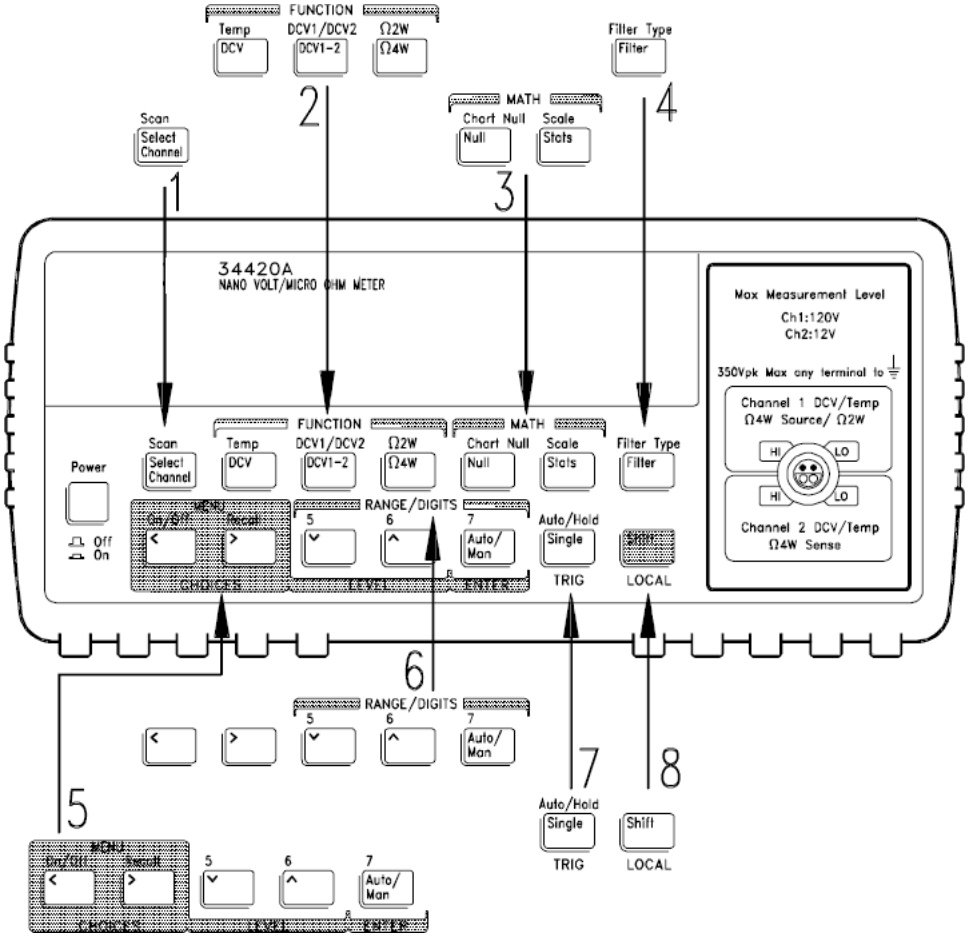
Convenient bench-top features

- Built-in math operations including thermistor, thermocouple and RTD temperature measurements
- Two channel input allows ratio and difference functions for voltage measurements
- Highly visible vacuum-fluorescent display
- Portable, ruggedized case with non-skid feet

Flexible system features

- GPIB (IEEE-488) interface and RS-232 interface are standard
- SCPI (Standard Commands for Programmable Instruments) and Keithley 181 compatibility
- Readings at up to 250 per second

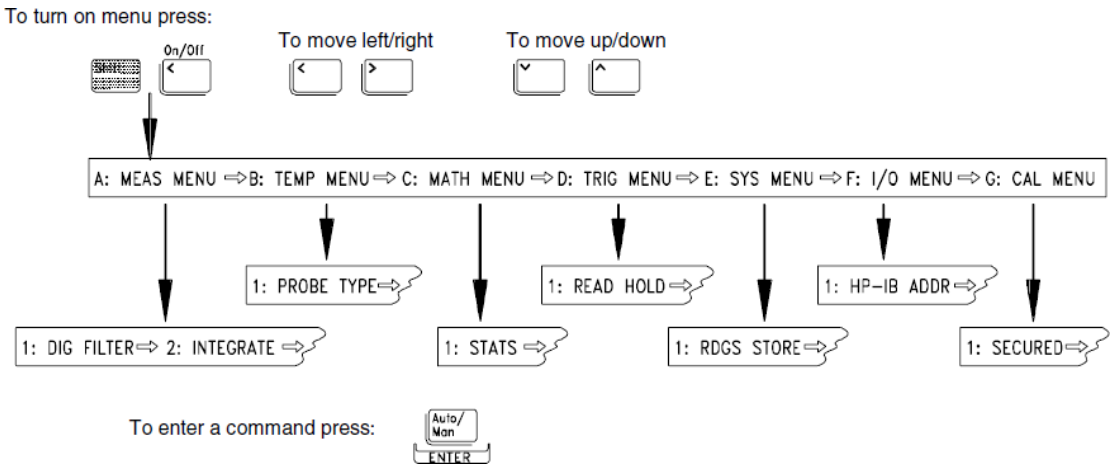
The Front-Panel at a Glance



1	Input channel select	**5**	Menu operation keys
2	Measurement function keys	**6**	Range/number of digits displayed keys
3	Math operation keys	**7**	Single trigger/autotrigger/auto hold key
4	Filter select key	**8**	Shift / local key

The Front-Panel Menu at a Glance

The menu is organized in a top-down tree structure with three levels.



A: MEASurement MENU

1: DIG FILTER ⇒ 2: INTEGRATE ⇒ 3: OCOMP Ω ⇒ 4: LOW POWER Ω ⇒ 5: LOW VOLT Ω ⇒ 6: LoV LIMIT Ω

B: TEMPerature MENU

1:PROBE TYPE ⇒ 2: UNITS ⇒ 3: RTD TYPE ⇒ 4:RTD Ro ⇒ 5:T/C TYPE ⇒ 6: COLD JUNCT ⇒ 7:JUNCT

C: MATH MENU

1: STATS ⇒ 2: NULL VALUE ⇒ 3: SCALE GAIN ⇒ 4 : SCALE OFST

D: TRIGger MENU

1: READ HOLD ⇒ 2: TRIG DELAY ⇒ 3: N SAMPLES

E: SYStem MENU

1: RDGS STORE ⇒ 2: SAVED RDGS ⇒ 3: ERROR ⇒ 4: TEST ⇒ 5: CHART OUT ⇒ 6: CHART SPAN ⇒
7: CHART NULL ⇒ 8: DISPLAY ⇒ 9: COMMA ⇒ 10: PRESET ⇒ 11: REVISION

F: Input/Output MENU

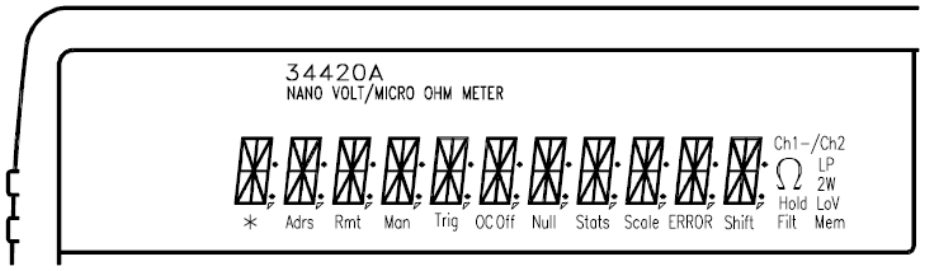
1: GPIB ADDR ⇒ 2: INTERFACE ⇒ 3: BAUD RATE ⇒ 4: PARITY ⇒ 5: LANGUAGE

G: CALibration MENU

1: SECURED ⇒ [1: UNSECURED] ⇒ [2:CALIBRATE] ⇒ [3: CHART ZERO] ⇒ [4 : CHART GAIN] ⇒
[5: INJECTED I] ⇒ 6:CAL COUNT ⇒ 7:MESSAGE

NOTE The commands enclosed in square brackets ([ ]) in the CAL MENU are "hidden" unless the meter is UNSECURED for calibration.

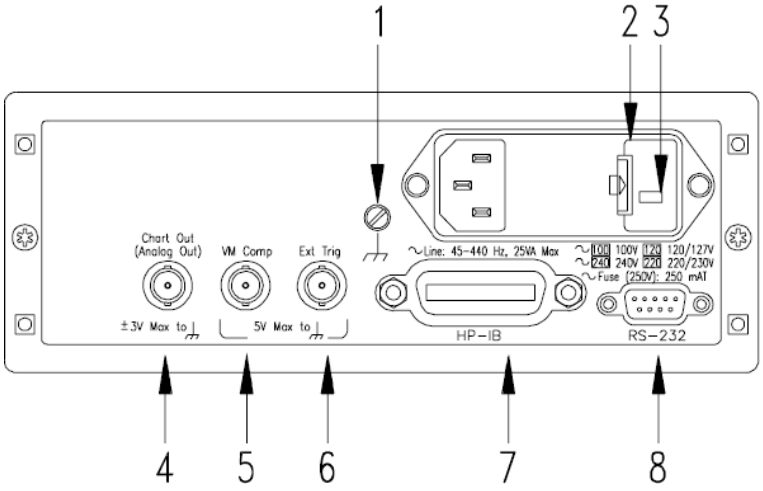
Display Annunciators



*****	Turns on during a measurement.
Adrs	Meter is addressed to listen or talk over the GPIB interface.
Rmt	Meter is in remote mode (using remote interface).
Man	Meter is using manual ranging (autorange is disabled).
Trig	Meter is waiting for a single trigger or external trigger.
OC Off	Offset compensation is turned off.
Null	A null value is being used.
Stats	Math statistics operations are being used.
Scale	Math scaling operations are being used.
ERROR	Hardware or remote interface command error(s) detected.
Shift	"Shift" key has been pressed.
Ch1	Meter input is on Channel 1.
Ch2	Meter input is on Channel 2.
Ch1 - Ch2	Meter is indicating the difference between inputs on channel 1 and channel 2.
Ch1 / Ch2	Meter is indicating the ratio of the inputs on channel 1 and channel 2.
W	Meter is measuring resistance (Ohms).
LP	Meter is using low power Ohms.
2W	Meter is using 2-wire Ohms (annunciator off indicates a 4-wire measurement).
LoV	Meter is using voltage limited Ohms.
Hold	Automatic reading hold is enabled.
Filt	The analog and/or the digital filter is enabled.
Mem	Turns on when reading memory is enabled.

To review the display annunciators, hold down the Shift key as you turn on the meter.

The Rear Panel at a Glance



1	Chassis ground	**5**	Voltmeter complete output terminal
2	Power-line fuse-holder assembly	**6**	External trigger input terminal
3	Power-line voltage setting	**7**	GPIB (IEEE-488) interface connector
4	Chart recorder output terminal (Analog out)	**8**	RS-232 interface connector

Use the front-panel Input / Output Menu to:

- Select the GPIB or RS-232 interface
- Set the GPIB bus address
- Set the RS-232 baud rate and parity

In This Book

Quick Start Chapter 1 prepares the meter for use and helps you get familiar with a few of its front-panel features.

Front-Panel Operation Chapter 2 introduces you to the front-panel menu and describes some of the meter's menu features.

Features and Functions Chapter 3 gives a detailed description of the meter's capabilities and operation. You will find this chapter useful whether you are operating the meter from the front panel or from a remote interface.

Remote Interface Reference Chapter 4 contains reference information to help you program the meter using a remote interface.

Error Messages Chapter 5 lists the error messages that may appear as you are working with the meter. Each listing contains enough information to help you diagnose and solve the problem.

Application Programs Chapter 6 contains several remote interface application programs to help you develop programs for your measurement application.

Measurement Tutorial Chapter 7 discusses measurement considerations and techniques to help you obtain the best accuracy and reduce sources of measurement error.

Specifications Chapter 8 lists the meter's specifications and describes how to interpret these specifications.

NOTE If you have questions relating to the operation of the meter, call **1-800-452-4844** in the United States, or contact your nearest Keysight Technologies Sales Office.

THIS PAGE HAS BEEN INTENTIONALLY LEFT BLANK.

Table of Contents

Safety Symbols 3
Safety Considerations 4
Measurement Category <valid for HH UG only - remove this> 5
Environmental Conditions 6
Regulatory Information 7
Safety compliance 7
EMC compliance 7
Regulatory Markings 8
Waste Electrical and Electronic Equipment (WEEE) Directive 9
Product category: 9
Sales and Technical Support 9

1 Quick Start

Quick Start 20
To Prepare the Meter For Use 21
If the Meter Does Not Turn On 22
To Adjust the Carrying Handle 24
To Make Input Connections 25
To Measure Voltage 27
To Measure Temperature With Thermistors 28
To Measure Temperature With RTDs 29
To Measure Temperature With Thermocouples 30
To Select a Range 31
To Set the Number of Digits 32
To Set the Integration Time 33
Front-Panel Display Formats 34
To Rack Mount the Meter 35

2 Front-Panel Operation

Front-Panel Operation 38
Front-Panel Menu Reference 39
A Front-Panel Menu Tutorial 43
To Select an Input Channel 52
To Select a Range 53
To Set or Change the Number of Digits 54
To Set the Integration Time 55
To Make Null (Relative) Measurements 56
To Store Minimum and Maximum Readings (Stats) 57
To Trigger the Meter 59
To Use Reading Hold 60
To Make Voltage Ratio and Difference Measurements 61
To Use Reading Memory 62

3 Features and Functions

Features and Functions 66
General Measurement Configuration 68
Input Filters 68
Integration Time 71
Reducing Measurement Noise 72
Number of Digits Displayed 74
Ranging 76
Overload Detection 77
Voltage Measurement Configuration 78
Input Channels 78
Resistance Measurement Configuration 81
Temperature Measurement Configuration 84
Measurement Units 84
RTD Measurements 85
Thermistor Measurements 85
Thermocouple Measurements 86
Math Operations 87

Null (Relative) Operation 90
Reading Hold 92
Triggering 93
Trigger Source Choices 95
The Wait-for-Trigger State 98
Halting a Measurement in Progress 98
Number of Samples 99
Number of Triggers 99
Trigger Delay 100
Automatic Trigger Delays 102
System-Related Operations 103
Reading Memory 103
Error Conditions 104
Self-Test 105
Display Control 106
Comma Separators 107
Firmware Revision Query 107
SCPI Language Version 108
Voltmeter Complete Terminal 109
External Trigger Terminal 110
Chart Output (Analog Output) 111
Chart Rollover 113
Remote Interface Configuration 115
Remote Interface Selection 115
GPIB Address 116
Baud Rate Selection (RS-232) 117
Parity Selection (RS-232) 118
Programming Language Selection 119
Connection to a Terminal or Printer (RS-232) 120
Calibration 121
Calibration Security 121
Calibration Count 124
Calibration Message 125

Defaults, Power-On and Reset States 126

4 Remote Interface Reference

Remote Interface Reference 131
Command Summary 133
Simplified Programming Sequence 142
Using the MEASure? Command 144
Using the CONFigure Command 144
Using the range and resolution Parameters 144
Using the READ? Command 145
Using the INITiate and FETCh? Commands 146
The MEASure? and CONFigure Commands 149
Setting the Function, Range, and Resolution 152
Selecting the Input Channel 155
Special Resistance Measurement Commands 156
Temperature Measurement Commands 157
Null (Relative) Commands 160
Input Filter Commands 162
Math Operation Commands 164
Statistics operation commands (AVERage) 165
Scale operation commands 166
Triggering 167
The Wait-for-Trigger State 169
Triggering Commands 170
Chart Output (Analog Out) Commands 172
System-Related Commands 173
The SCPI Status Model 175
What is an Event Register? 175
What is an Enable Register? 175
What is a Condition Register? 176
SCPI Status System 177
The Status Byte 178

Using Service Request (SRQ) and Serial POLL 179
Using *STB? to Read the Status Byte 180
To Interrupt Your Bus Controller Using SRQ 180
To Determine When a Command Sequence is Completed 180
How to Use the Message Available Bit (MAV) 181
Using *OPC to Signal When Data is in the Output Buffer 181
The Standard Event Register 182
The Questionable Data Register 183
The Operational Status and Condition Registers 184
Status Reporting Commands 186
Calibration Commands 189
RS-232 Interface Configuration 191
RS-232 Configuration Overview 191
RS-232 Data Frame Format 192
Connection to a Computer or Terminal 192
Connection to a Printer 194
DTR / DSR Handshake Protocol 194
RS-232 Troubleshooting 195
RS-232 Interface Commands 197
An Introduction to the SCPI Language 198
SCPI Data Types 200
Input Message Terminators 201
Output Data Formats 202
Using Device Clear to Halt Measurements 203
TALK ONLY for Printers 204
To Set the GPIB Address 205
To Select the Remote Interface 207
To Set the Baud Rate 209
To Set the Parity 211
To Select the Programming Language 213
Alternate Programming Language Compatibility 215
Specific Issues 215

SCPI Compliance Information 216
IEEE-488 Compliance Information 219

5 Error Messages

Error Messages 222
Execution Errors 223
Self-Test Errors 230
Calibration Errors 232

6 Application Programs

Application Programs 236
HP BASIC Language Programs 237
HP BASIC/GPIB Program Example 1 237
HP BASIC/GPIB Program Example 2 238
HP BASIC/GPIB Program Example 3 240
HP BASIC/GPIB Program Example 4 243
RS-232 Operations with QuickBASIC 245
Microsoft Visual Basic Language Program 247
C Language Programs 254
Compiling and Linking a C Program 254
C/GPIB Example 1 256
C/GPIB Example 2 258
C/GPIB Program Example 3 262
Microsoft Excel Macro Example 266
Macro Comments 267

7 Measurement Tutorial

Measurement Tutorial 272
Measurement Techniques and Sources of Error 273
Voltage Measurements 273
Resistance Measurements 283
Temperature Measurements 288
Cleaning the Connector Contacts 297

Input Connector 299
Obtaining Parts For the Input Connector 300

8 Specifications

THIS PAGE HAS BEEN INTENTIONALLY LEFT BLANK.

1 Quick Start

Quick Start 20
To Prepare the Meter For Use 21
If the Meter Does Not Turn On 22
To Adjust the Carrying Handle 24
To Make Input Connections 25
To Measure Voltage 27
To Measure Temperature With Thermistors 28
To Measure Temperature With RTDs 29
To Measure Temperature With Thermocouples 30
To Select a Range 31
To Set the Number of Digits 32
To Set the Integration Time 33
Front-Panel Display Formats 34
To Rack Mount the Meter 35

Quick Start

This chapter helps you prepare the meter for use and contains exercises designed to get you started with the meter, its menus, and the front panel.

The front panel has two rows of keys to select various functions and operations. Most keys have a shifted function printed in blue above the key. To perform a shifted function, press Shift (the **Shift** display annunciator will turn on). Then, press the key that has the desired label above it. For example, to select the temperature measurement function, press Shift DCV .

If you accidentally press Shift , just press it again to turn off the **Shift** annunciator.

NOTE The rear cover of this book is a fold-out Quick Reference Guide. On this cover you will find a quick summary of various meter features. Inside the rear cover is a diagram of the front panel menu options.

To Prepare the Meter For Use

The following steps help you verify that the meter is ready for use.

1 Check the list of supplied items.

Verify that you have received the following items with your meter. If any item is missing, contact your nearestKeysight Technologies Sales Office.

- One low thermal input cable.
- One low thermal four-wire shorting plug.
- One power cord.
- This *User's Guide*.
- One *Service Guide*.
- One folded *Quick Reference* card.
- Certificate of Calibration with removeable calibration label.
- One bottle of DeoxIT™ contact cleaner.[1]

2 Connect the power cord and turn the meter on.

The front panel display will light up while the meter performs its power-on self-test. The GPIB bus address is displayed. The meter is left in 61/2 digit resolution, channel 1 input, and digital filter on.

To view the display with all annunciators turned on, hold down Shift *as you turn on the meter.*

3 Perform a complete self-test.

The complete self-test performs a more extensive series of tests than those performed at power-on. Hold down Shift as you press the power switch to turn on the meter; *continue to hold* Shift *for more than 5 seconds*. The self-test will begin when you release the key.

If the self-test is successful, "PASS" is displayed on the front panel. If the test is not successful, "FAIL" is displayed and the **ERROR** annunciator turns on. See the *Service Guide* for instructions on returning the meter to Keysight Technologies for service.

[1] DeoxIT™ is a trademark of CAIG Laboratories, Inc., San Diego, California.

If the Meter Does Not Turn On

Use the following steps to help solve problems you might experience when turning on the meter. If you need more help, see the *Service Guide* for instructions on returning the meter to Keysight Technologies for service.

1 **Verify there is ac power to the meter.**

First verify that the meter's power switch is in the "On" position. Make sure the power cord is firmly plugged into the power module on the meter's rear panel. Verify that the power source is energized.

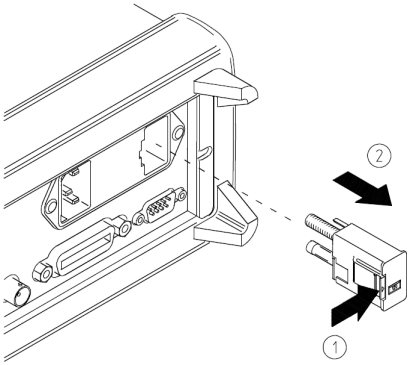
2 **Verify the power-line voltage setting.**

The line voltage is set to the proper value for your country when the meter is shipped from the factory. Change the voltage setting if it is not correct. The settings are: 100, 120, 220, or 240 Vac (for 230 Vac operation, use the 220 Vac setting).

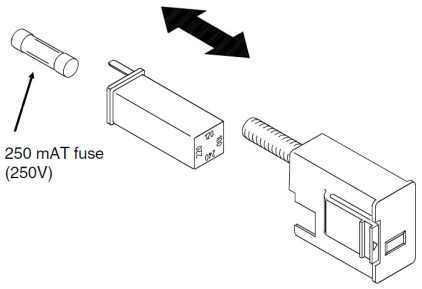
See the diagram on the next page to verify or change the setting.

NOTE To replace the 250 mAT 250 V fuse, order part number 2110-0817. This fuse is used for all power line voltage settings.

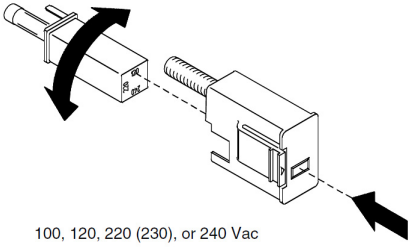
1 **Remove the power cord.** Remove the fuse holder assembly from the rear



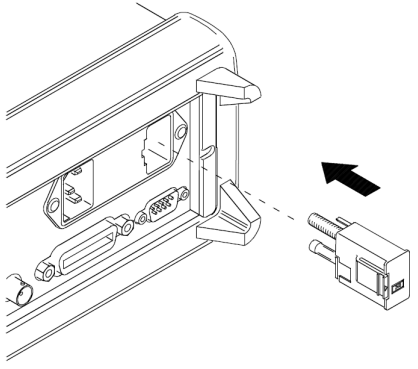
2 *Install the fuse.* Remove the line-voltage selector from the assembly.



3 Rotate the line-voltage selector until the correct voltage appears in the window.



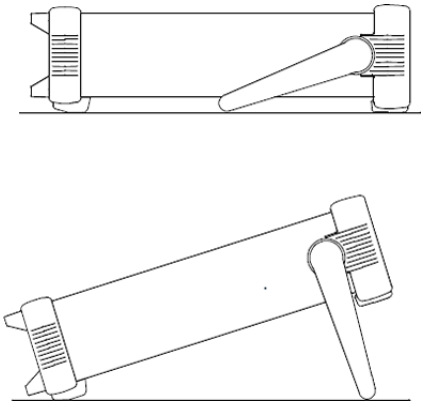
4 Replace the fuse-holder assembly in the rear panel.



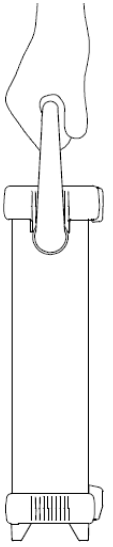
Install the correct fuse and verify that the correct line voltage appears in the window.

To Adjust the Carrying Handle

To adjust the position, grasp the handle by the sides and *pull outward*. Then, rotate the handle to the desired position.



Bench-top viewing positions

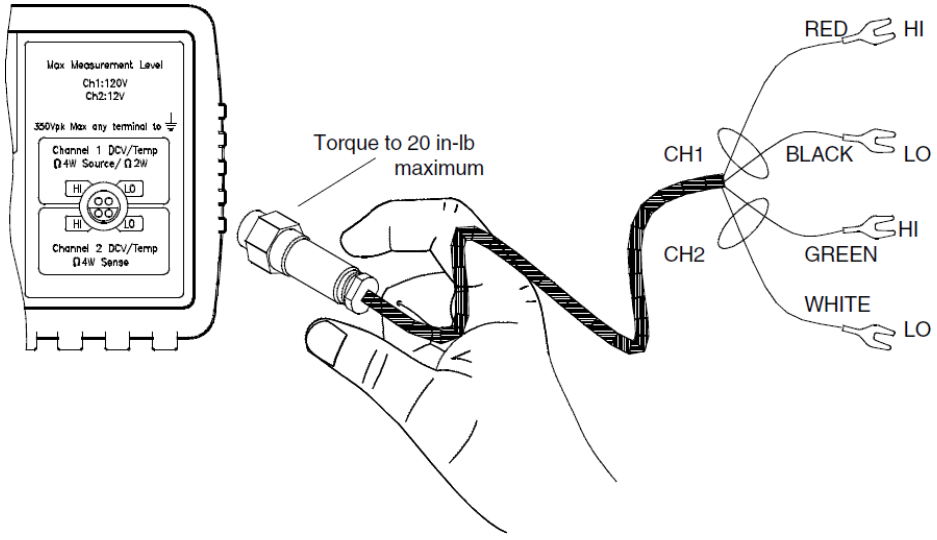


Carrying Position

To Make Input Connections

Using the Cable Provided

Align the front panel connector and cable conductors, press in.Tighten coupling nut.

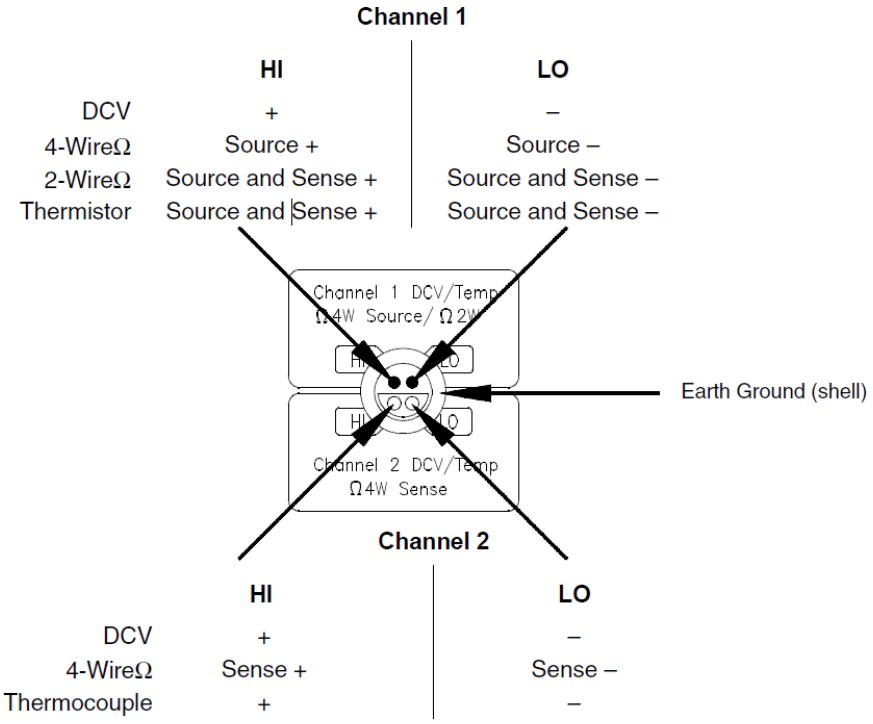


NOTE The connector and cable are an integral part of the measurement system. For the highest accuracy, use the copper cable and connectors supplied by Keysight Technologies.

The conductors may require occasional cleaning to remove oxides. Cleaning the conductors is described on page 297.

WARNING The connector body and cable shield are connected to earth ground.

Using a Custom Cable



Refer to page 299 for additional information about building custom input cables.

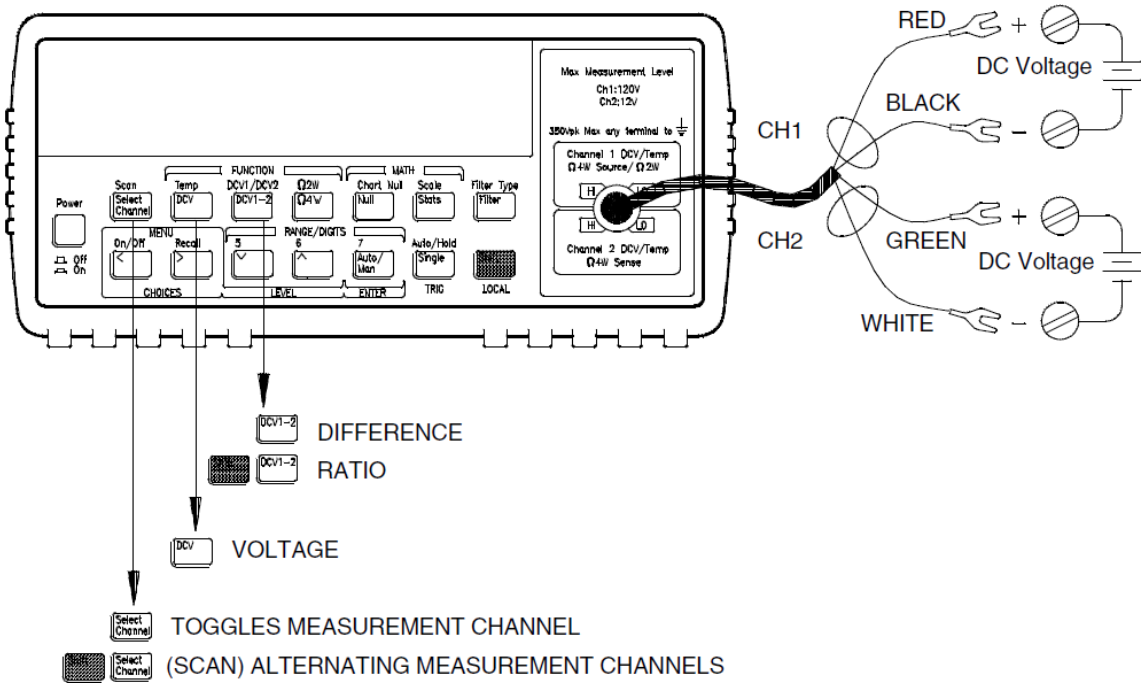
To Measure Voltage

Channel 1 Ranges: 1 mV, 10 mV, 100 mV, 1 V, 10 V, 100 V
Maximum resolution: 0.1 nV (*on 1 mV range*)

Channel 2 Ranges: 1 mV, 10 mV, 100 mV, 1 V, 10 V
Maximum resolution: 0.1 nV (*on 1 mV range*)

Independent NULL for each channel

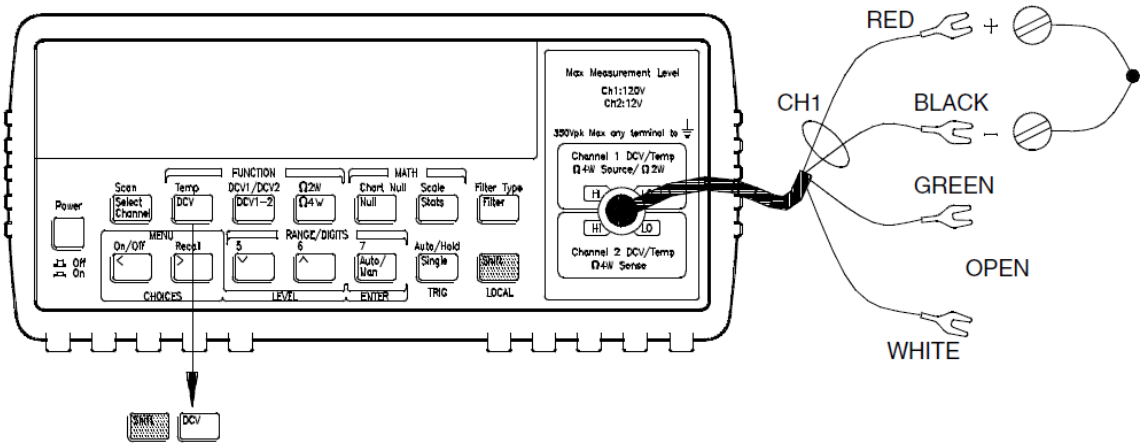
Channel 1 LO to Channel 2 LO isolated to 150 Vpeak.



NOTE Scan mode alternates between input channels every 3 seconds or every measurement, whichever is longer.

To Measure Temperature With Thermistors

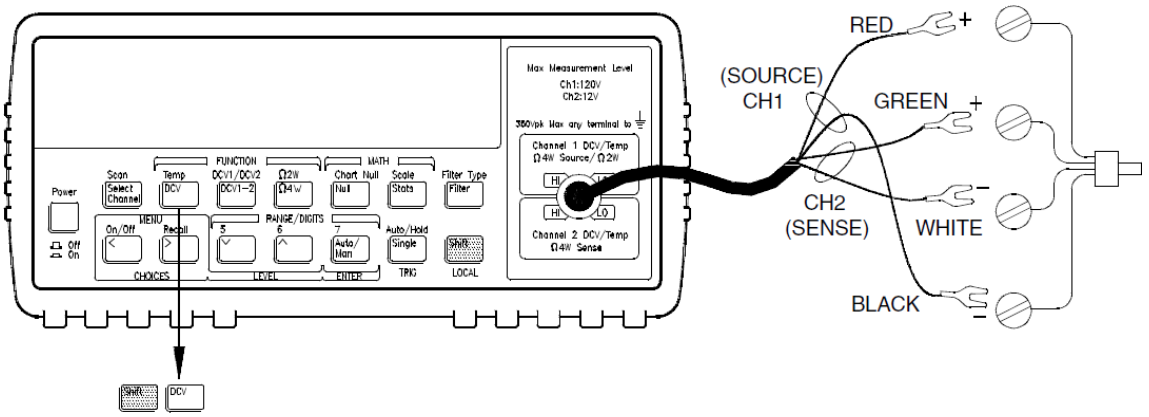
Thermistor type: 5 kΩ



To Measure Temperature With RTDs

4-wire, type: α = .00385 (DIN/IEC 751) or α = .00391

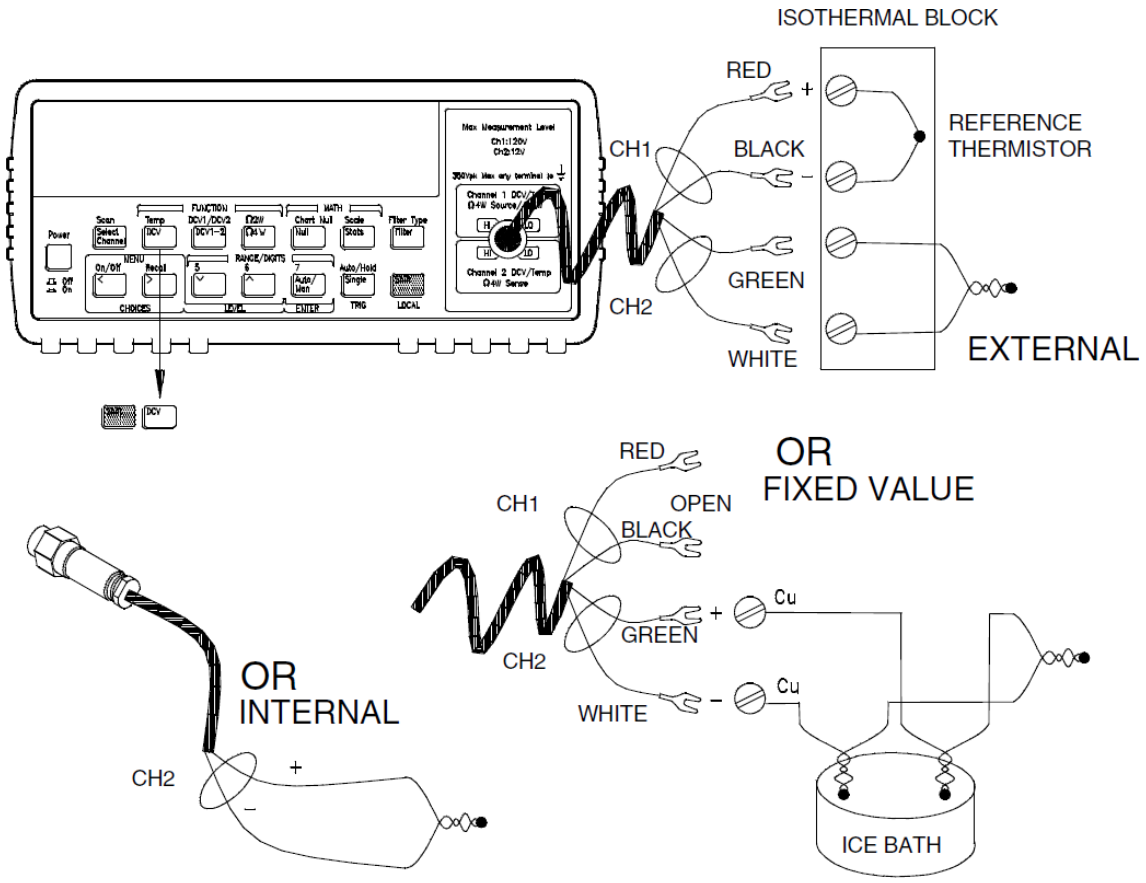
R_0 = 4.9 Ω to 2.1 kΩ



To Measure Temperature With Thermocouples

Thermocouple types: B, E, J, K, N, R, S, T

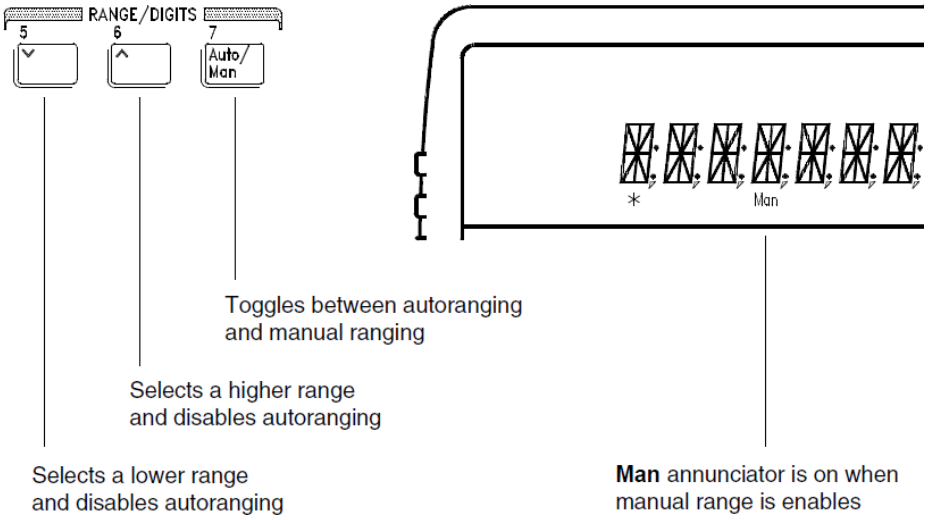
Reference: external thermistor, external fixed value, or internal thermistor. Channel 2 only.



NOTE When using the internal thermistor as the temperature reference, you should make a custom cable to connect the thermocouple wire directly to the input terminals (*see page 299*).

To Select a Range

You can let the meter automatically select the range using *autoranging* or you can select a fixed range using *manual ranging*.



- Autoranging is selected at power-on and after a remote interface reset.
- Autorange thresholds:

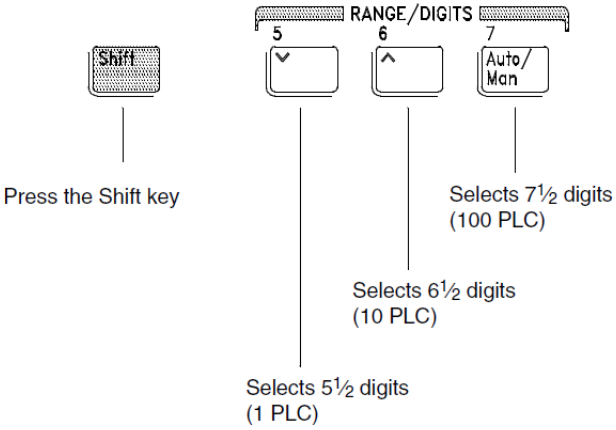
  Down range at <10% of range
  Up range at >120% of range

- If the input signal is greater than the present range can measure, the meter will give an *overload* indication ("OVLD").
- When making difference or ratio measurements, an overload on either channel will give an overload indication ("OVLD").

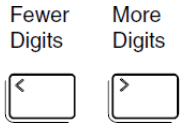
NOTE For voltage measurements, ranging is local to the selected channel. This means that you can select the ranging method (auto or manual) for each channel independently. When manually ranging, the selected range is local to the active channel; the meter remembers the range when you switch between channels.

To Set the Number of Digits

You can set the display to show 4½, 5½, 6½, or 7½ digits. In this book, the most significant digit (leftmost on the display) is referred to as the "½" digit, since it can only be a "0" or "1". *The number of digits displayed also depends upon the integration time set, see* *page 33*.



- The number of digits is set to 6½ digits at power-on and after a remote interface reset.
- The number of digits shown is dependent upon integration time and filter settings. The meter will not allow you to show more digits than the practical measurement capability of the meter. You can, however, reduce the number of digits shown.



To Set the Integration Time

Integration time is specified in Number of Power Line Cycles (NPLC). You can set the NPLC to 0.02, 0.2, 1, 2, 10, 20, 100, or 200.

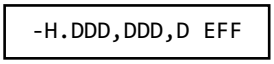
- You can set the integration time to one of three fixed values by choosing the number of digits displayed, see page 32.
- You can set the integration time in the MEASure menu using the INTEGRATE command.
- The Integration Time is directly related to the maximum number of digits the meter will display.

NPLC	Filter Off Max Digits Shown	Filter On Max Digits Shown
0.02	4½	5½
0.2	5½	6½
1	6½	7½
2	6½	7½
10	6½	7½
20	7½	7½
100	7½	7½
200	7½	7½

- You can always show fewer digits than the maximum allowed (the minimum number of digits shown is 4½).

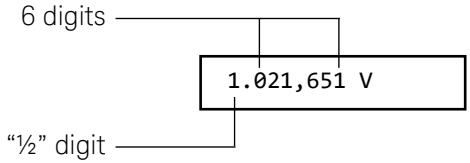
NOTE Integration Time is local to the selected function. This means that you can select the integration time for each function independently. The meter remembers integration time when you switch between functions.

Front-Panel Display Formats

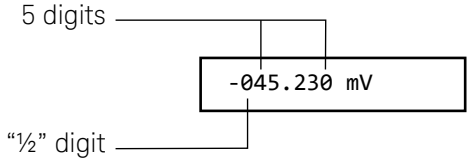


Front-panel display format

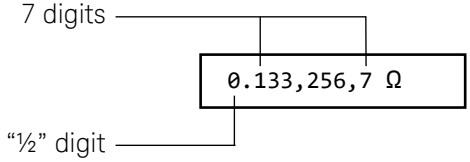
- **–** Negative sign or blank (positive)
- **H** "½" digit (0 or 1)
- **D** Numeric Digits
- **E** Exponent (m, k, M)
- **F** Measurement units



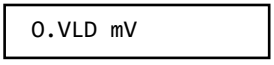
This is the 1 V range, 6½ digits are displayed.



This is the 100 mV range, 5½ digits are displayed.



This is the 1 ohm range, 7½ digits are displayed.

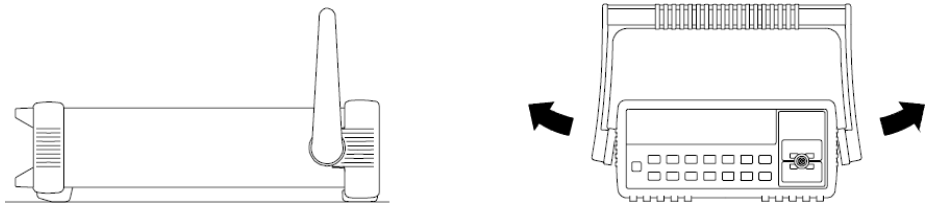


This is an overload indication on the 1 mV range.

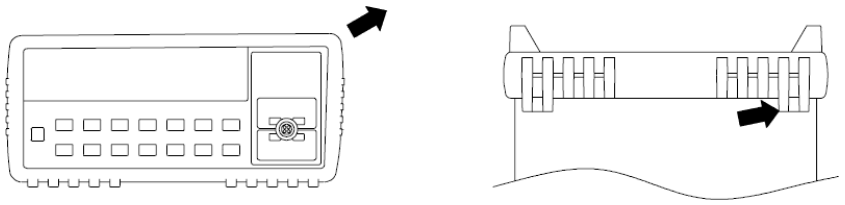
To Rack Mount the Meter

You can mount the meter in a standard 19-inch rack cabinet using one of three optional kits available. Instructions and mounting hardware are included with each rack-mounting kit. Any *System II* instrument of the same size can be rack-mounted beside the 34420A meter.

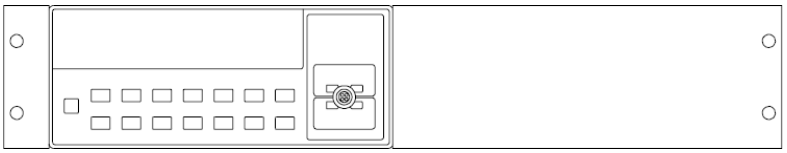
Remove the carrying handle, and the front and rear rubber bumpers, before rack-mounting the meter.



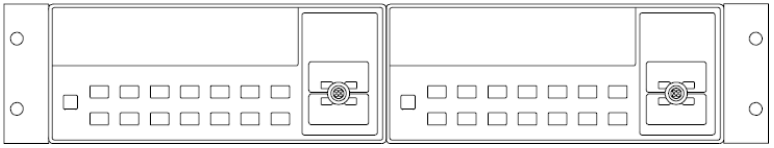
To remove the handle, rotate it to the vertical position and pull the ends outward.



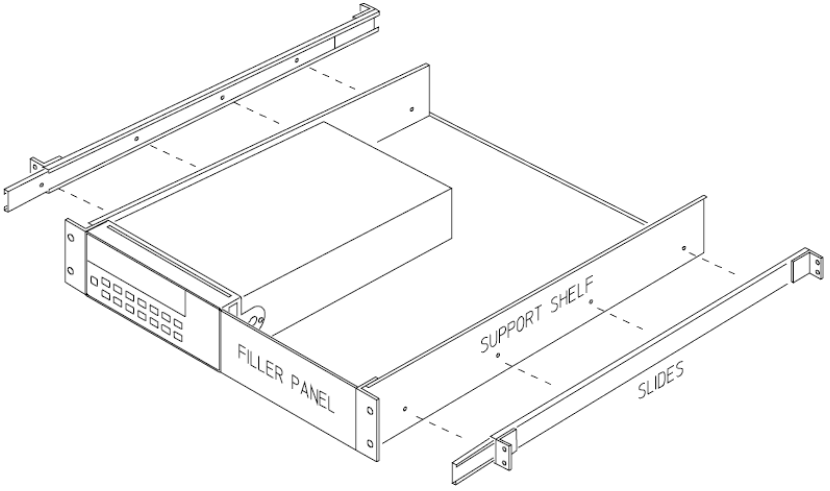
To remove the rubber bumper, stretch a corner and then slide it off.



To rack mount a single instrument, order adapter kit 5063-9240.



To rack mount two instruments side-by-side, order lock-link kit 5061-9694 and flange kit 5063-9212.



To install one or two instruments in a sliding support shelf, order shelf 5063-9255, and slide kit 1494-0015 (for a single instrument, also order filler panel 5002-3999).

2 Front-Panel Operation

Front-Panel Operation 38
Front-Panel Menu Reference 39
A Front-Panel Menu Tutorial 43
To Select an Input Channel 52
To Select a Range 53
To Set or Change the Number of Digits 54
To Set the Integration Time 55
To Make Null (Relative) Measurements 56
To Store Minimum and Maximum Readings (Stats) 57
To Trigger the Meter 59
To Use Reading Hold 60
To Make Voltage Ratio and Difference Measurements 61
To Use Reading Memory 62

Front-Panel Operation

This chapter assumes you are familiar with the meter and menu operations. You should also understand how to make connections for the various types of measurements. If you are not familiar with this information, please read Chapter 1, "Quick Start" starting on page 19.

This chapter does not give a detailed description of every front-panel key or menu operation. It does, however, give you a good overview of the front-panel menu and the most common front-panel operations. See Chapter 3, "Features and Functions" starting on page 65, for a complete discussion of the meter's capabilities and operation.

Front-Panel Menu Reference

A: MEASurement MENU

1: DIG FILTER ⇒ 2: INTEGRATE ⇒ 3: OCOMP Ω ⇒ 4: LOW POWER Ω ⇒ 5: LOW VOLT Ω ⇒ 6: LoV LIMIT Ω

1: DIG FILTER	Selects the digital filter speed. Can be set to FAST, MEDIUM, or SLOW.
2: INTEGRATE	Sets the measurement integration time.
3: OCOMP Ω	Enables or disables offset compensation for resistance measurements.
4: LOW POWER Ω	Enables or disables low power ohms measurements.
5: LOW VOLT Ω	Enables or disables voltage limited resistance measurements.
6: LoV LIMIT Ω	Sets the voltage limit for voltage limited resistance measurements.

B: TEMPerature MENU

1:PROBE TYPE ⇒ 2: UNITS ⇒ 3: RTD TYPE ⇒ 4:RTD Ro ⇒ 5:T/C TYPE ⇒ 6: COLD JUNCT ⇒ 7:JUNCT

1: PROBE TYPE	Selects thermocouples, thermistors, or RTDs as the temperature probe.
2: UNITS	Sets the measurement units to either °C, °F, or Kelvins.
3: RTD TYPE	Sets the type of RTD in use.
4: RTD Ro	Sets the nominal (0°C) value for the RTD in use.
5: T/C TYPE	Selects the type of thermocouple in use.
6: COLD JUNCT	Sets the source of the temperature reference junction.
7: JUNCT TEMP	Allows direct entry of reference junction temperature.

C: MATH MENU

1: STATS ⇒ 2: NULL VALUE ⇒ 3: SCALE GAIN ⇒ 4 : SCALE OFST

1:STATS	Recalls the values of minimum, maximum, standard deviation, average, peak-to-peak, and number of readings.
2: NULL VALUE	Selects independent null settings for voltage on channels 1 and 2, resistance, and temperature functions.
3: SCALE GAIN	Sets the gain for linear scaling of the data.
4: SCALE OFST	Sets the offset for linear scaling of data.

D: TRIGger MENU

1: READ HOLD ⇒ 2: TRIG DELAY ⇒ 3: N SAMPLES

1: READ HOLD	Selects the reading hold sensitivity band.
2: TRIG DELAY	Specifies a time delay between the trigger and the start of the measurement.
3: N SAMPLES	Sets the number of samples taken per trigger.

E: SYStem MENU

1: RDGS STORE ⇒ 2: SAVED RDGS ⇒ 3: ERROR ⇒ 4: TEST ⇒ 5: CHART OUT ⇒ 6: CHART SPAN ⇒
7: CHART NULL ⇒ 8: DISPLAY ⇒ 9: COMMA ⇒ 10: PRESET ⇒ 11: REVISION

1: RDGS STORE	Enables or disables reading memory.
2: SAVED RDGS	Recalls readings stored in memory (up to 1024 readings).
3: ERROR	Retrieves errors from the error queue (up to 20 errors).
4: TEST	Performs a complete self-test or individual self-tests.
5: CHART OUT	Enables or disables the chart recorder output (Analog Out).
6: CHART SPAN	Sets the chart recorder output range (Analog Out).
7: CHART NULL S	Sets the chart recorder output offset (Analog Out).
8: DISPLAY	Enables or disables the front panel display.
9: COMMA	Enables or disables a comma separator between digits in the display.
10: PRESET	Returns the meter to factory default settings.
11: REVISION	Displays the meter's firmware revision code.

F: Input / Output MENU

1: GPIB ADDR ⇒ 2: INTERFACE ⇒ 3: BAUD RATE ⇒ 4: PARITY ⇒ 5: LANGUAGE

1: GPIBADDR	Sets the GPIB bus address (0 to 30, default 22).
2: INTERFACE	Selects either the GPIB or RS-232 remote interface.
3: BAUD RATE	Sets the baud rate for RS-232 operation.
4: PARITY	Selects even, odd, or no parity for RS-232 operation.
5: LANGUAGE	Selects interface language, either SCPI or Keithley 181.

G: CALibration MENU[1]

1: SECURED ⇒ [1: UNSECURED] ⇒ [2:CALIBRATE] ⇒ [3: CHART ZERO] ⇒ [4 : CHART GAIN] ⇒ [5: INJECTED I] ⇒ 6:CAL COUNT ⇒ 7:MESSAGE

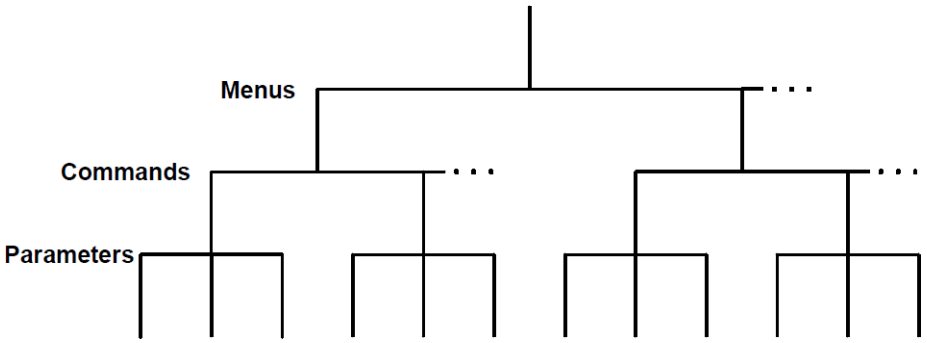
1: SECURED	The meter is secured against calibration; enter code to unsecure.
1: UNSECURED	The meter is unsecured for calibration; enter code to secure.
2: CALIBRATE	Performs a calibration of meter; must be UNSECURED.
3: CHART ZERO	Performs a calibration of the chart recorder output zero level.
4: CHART GAIN	Performs a calibration of the chart recorder output gain.
5: INJECTED I	Performs a calibration to minimize the injected current.
6: CAL COUNT	Reads the total number of times the meter has been calibrated.
7: MESSAGE	Reads the first 11 characters of a calibration string, if any, entered from the remote interface.

[1] The commands enclosed in square brackets ([ ]) are “hidden” unless the meter is UNSECURED for calibration.

A Front-Panel Menu Tutorial

This section is a step-by-step tutorial which shows how to use the front-panel menu. We recommend that you spend a few minutes with this tutorial to get comfortable with the structure and operation of the menu.

The menu is organized in a top-down tree structure with three levels (menus, commands, and parameters). You move down [^] or up [v] the menu tree to get from one *level* to the next. Each of the three levels has several horizontal *choices* which you can view by moving left [<] or right [>].



- To *turn on* the menu, press [Shift] [<] (Menu On/Off).
- To *turn off* the menu, press [Shift] [<] (Menu On/Off), or press any of the function or math keys on the top row of front-panel keys.
- To *execute* a menu command, press [Auto/Man].
- To *recall* the last menu command that was executed, press [Shift] [>] (Menu Recall).
- To *turn off* the menu at any time without saving changes, press any function key

NOTE **If you become confused or get lost at any point during the tutorial, simply turn off the menu and start over again with step 1 for that example.**

Messages Displayed During Menu Use

TOP OF MENU -You pressed ∧ while on the menus level; this is the top level of the menu and you cannot go any higher.

To turn off the menu, press Shift < (Menu On/Off). To move across the choices on a level, press < or > . To move down a level, press ∨ .

MENUS - You are on the menus level. Press < or > view the choices.

COMMANDS - You are on the commands level. Press < or > to view the command choices within the selected menu group.

PARAMETER - You are on the parameter level. Press < or > to view and edit the parameter for the selected command.

MENU BOTTOM - You pressed ∨ while on the parameter level; this is the bottom level of the menu and you cannot go any lower.

To turn off the menu, press Shift < *(Menu On/Off). To move up a level, press ∧ .*

ENTERED - The change made on the parameter level is saved. This is displayed after you press Auto/Man (Menu Enter) to execute the command.

MIN VALUE - The value you specified on the parameter level is too small for the selected command. The minimum value allowed is displayed for you to edit.

MAX VALUE - The value you specified on the parameter level is too large for the selected command. The maximum value allowed is displayed for you to edit.

EXITING - You will see this message if you turn off the menu by pressing Auto/Man (Menu On/Off) or a front-panel function/math key. You did not edit any values on the parameter level and changes were NOT saved.

NOT ENTERED - You will see this message if you turn off the menu by pressing Shift < (Menu On/Off) or a front-panel function/math key. You did some editing of parameters but the changes were NOT saved. Press Auto/Man (Menu Enter) to save changes made on the parameter level.

VOLTS ONLY - The requested function is only available for voltage measurements.

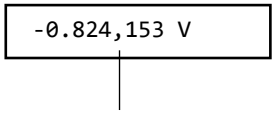
UNAVAILABLE - The action requested is NOT allowed in the present configuration.

Menu Example 1

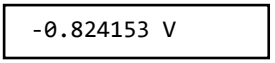
The following steps show you how to turn on the menu, move up or down between levels, move across the choices on each level, and turn off the menu.

In this example, you will turn off the display comma separator.

The meter can display readings on the front panel with or without a comma separator. The following steps show how to disable the comma.



With comma separator (factory setting)



Without comma separator

1 Turn on the menu.

Shift On/Off <

You enter the menu on the *menus* level. The MEAS MENU is your first choice on this level.

2 Move across to the SYS MENU choice on this level.

> > > >

There are six menu group choices available on the *menus* level. Each choice has a letter prefix for easy identification (**A:** , **B:** , etc.).

3 Move down to the *commands* level within the SYS MENU.

[V]

The RDGS STORE command is your first choice on this level.

```
1: RDGS STORE
```

4 Move across to the COMMA command on the *commands* level.

[>] [>] [>] [>]

[>] [>] [>] [>]

There are eleven command choices available in the SYS MENU. Each choice on this level has a number prefix for easy identification (**1:** , **2:** , etc.).

```
9: COMMA
```

5 Move down a level to the COMMA parameter choices.

[V]

The first parameter choice is ON for the COMMA command (the comma setting is stored in non-volatile memory and ON is the factory setting).

```
ON
```

6 Move across to the *OFF* choice.

[>]

There are two parameter choices for COMMA.

```
OFF
```

7 Save the change and turn off the menu.

Auto/Man
ENTER

The meter beeps and displays a message to show that the change is now in effect. You are then exited from the menu. The meter will no longer show commas in the display.

```
ENTERED
```

Menu Example 2

The following exercise demonstrates how to use the *menu recall* feature as a shortcut to set the COMMA command back to its original setting. *You must perform the steps in Menu Example 1 before you start this example.*

1 Use menu recall to return to the COMMA command.

Shift Recall >

This returns you to the COMMA command, which was the last command used before you exited the menu in the Example 1.

```
9: COMMA
```

2 Move down to the COMMA parameter choices.

V

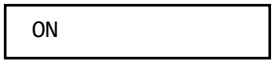
The first parameter choice is OFF (the current setting from Example 1).

```
OFF
```

3 Move across to the *ON* choice.

[>]

Set the parameter back to its original value.



4 Save the change and turn off the menu.

[Auto/Man]
ENTER

The meter beeps and displays a message to show that the change is now in effect. You are then exited from the menu.

Menu Example 3

Some commands in the menu require that you enter a numeric parameter value. The following steps show you how to enter a number in the menu. For this example, you will set the null value for channel 1 to –30 millivolts.

Make sure the meter has channel 1 selected, is in the dc voltage function, and has 6½ digits displayed. Connect the four-wire short to the meter input.

1 Turn on the menu.

[Shift] [On/Off <]

You enter the menu on the menus level. The MEAS MENU is your first choice on this level.

2 Move across to the MATH MENU choice on this level.

[>] [>]

There are six menu group choices available on this level.

```
C: MATH MENU
```

3 Move down to the *commands* level within the MATH MENU.

[V]

The STATS command is your first choice on this level.

```
1: STATS
```

4 Move across to the NULL VALUE command.

[>]

```
2: NULL VALUE
```

5 Move down to edit the NULL VALUE parameter.

[V]

The null value should be 0.000000 V when you come to this point in the menu for the first time. For this example, you will first set the null value to –0.300000 volts.

```
^ 0.000,000,0 V
```

When you see the flashing "^" on the left side of the display, you can abort the edit and return to the "commands" level by pressing [^].

6 Make the number negative.

[V] [V]

The left most character on the display toggles between +, –, and ^.

```
– 0.000,000,0 V
```

7 Move the flashing cursor over to edit the second digit.

[>] [>]

Notice that the digit to the right of the decimal point is flashing.

```
– 0.000,000,0 V
```

8 Increment the digit until "3" is displayed.

[^] [^] [^]

You decrement or increment each digit independently. Neighboring digits are not affected.

```
– 0.300,000,0 V
```

9 Move the flashing cursor over to the *units* location.

[>] [>] [>] [>]

[>] [>] [>]

Notice that the units are flashing on the right side of the display.

```
– 0.300,000,0 V
```

10 Decrease the displayed number by a factor of 10.

[v]

Notice that the position of the decimal point changes and the displayed number increases by a factor of 10 and the **mV** annunciator is on.

```
- 030.000,00 mV
```

11 Save the change and turn off the menu.

[Auto/Man]
ENTER

The meter beeps and displays a message to show that the change is now in effect. You are then exited from the menu.

```
ENTERED
```

You will be returned to the normal measurement display, and, if you have installed the four-wire short, the display will show a positive measurement in the approximate amount of the null value just entered.

The **Null** annunciator is on in the display.

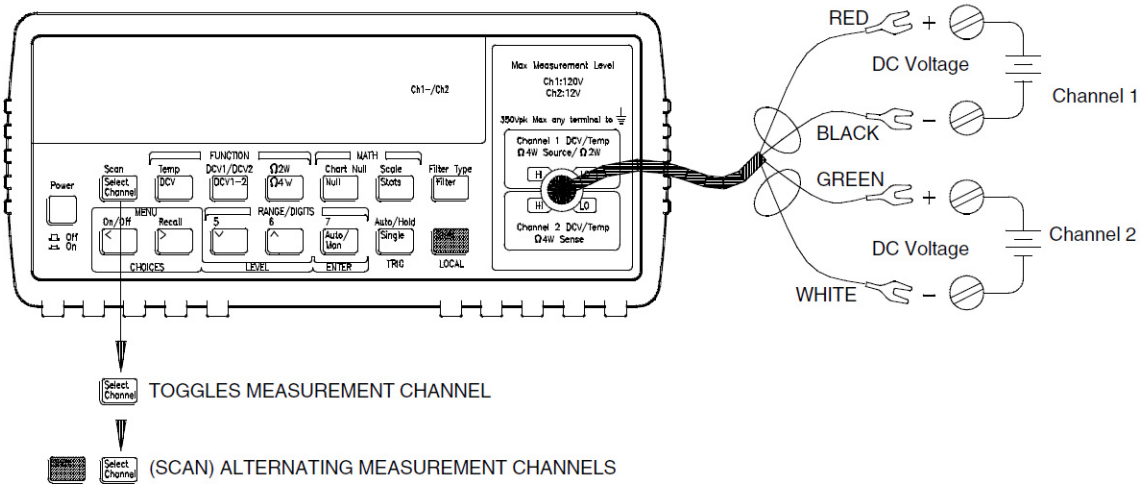
Keep in mind that null is turned on and - 0.03 volts is used as the null value for measurements. To clear the null value, press [Null].

Hint You can use the menu recall feature to save steps when entering the null value. First, press [Null] on the front panel and then press [Shift] [>] (Menu Recall). The meter will jump to the **2:** NULL VALUE command. Press [v]. The null value in the display is the first reading taken after you turned NULL on. You can then edit this number as described.

NOTE **This is the end of the front-panel menu tutorial. The remainder of the chapter discusses several of the most common front-panel operations.**

To Select an Input Channel

For *voltage measurements*, you have two independent input channels.

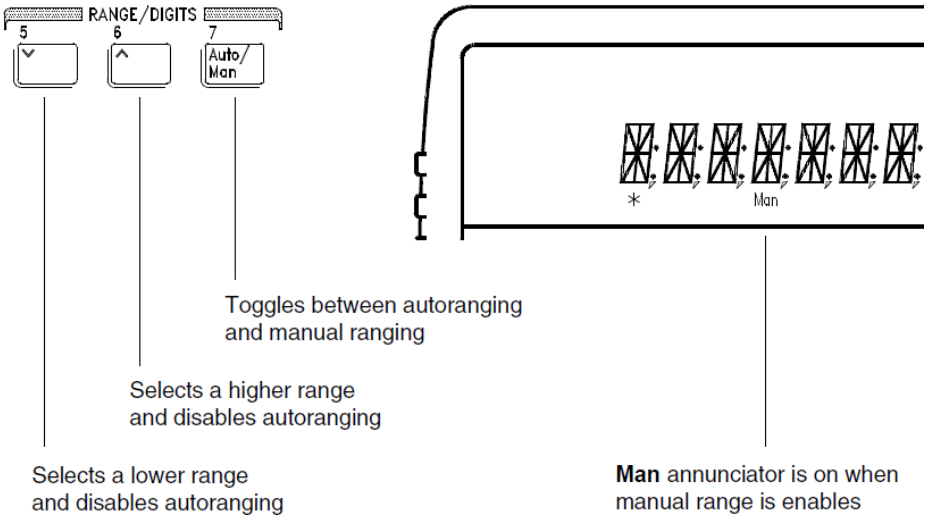


- Pressing Channel toggles the display between Channel 1 and Channel 2.
- Pressing Shift Channel (Scan) creates an alternating Channel 1 and Channel 2 display. Press Channel again to cancel.

NOTE Scan mode alternates between input channels every 3 seconds or every measurement, whichever is longer. If you are using the scan mode, turn off all the input filters (both digital and analog), see page 68.

To Select a Range

You can let the meter automatically select the range using *autoranging* or you can select a fixed range using *manual ranging*.



- Autoranging is selected at power-on and after a remote interface reset.
- Autorange thresholds:

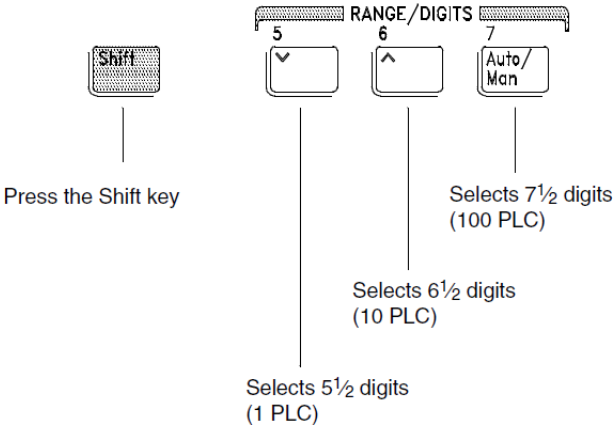
  Down range at <10% of range
  Up range at >120% of range

- If the input signal is greater than the present range can measure, the meter will give an *overload* indication ("OVLD").
- When making difference or ratio measurements, an overload on either channel will give an overload indication ("OVLD").

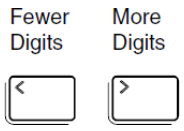
NOTE For voltage measurements, ranging is local to the selected channel. This means that you can select the ranging method (auto or manual) for each channel independently. When manually ranging, the selected range is local to the active channel; the meter remembers the range when you switch between channels.

To Set or Change the Number of Digits

You can set the display to show 4½, 5½, 6½, or 7½ digits. In this book, the most significant digit (leftmost on the display) is referred to as the "½" digit, since it can only be a "0" or "1". *The number of digits displayed also depends upon the integration time set, see* *page 55*.



- The number of digits is set to 6½ digits at power-on and after a remote interface reset.
- The number of digits shown is dependent upon integration time and filter settings. The meter will not allow you to show more digits than the practical measurement capability of the meter. You can, however, reduce the number of digits shown.



To Set the Integration Time

Integration time is specified in Number of Power Line Cycles (NPLC). You can set the NPLC to 0.02, 0.2, 1, 2, 10, 20, 100, or 200.

- You can set the integration time to one of three fixed values by choosing the number of digits displayed, see page 54.
- You can set the integration time in the MEASure menu using the INTEGRATE command.
- The Integration Time is directly related to the maximum number of digits the meter will display.

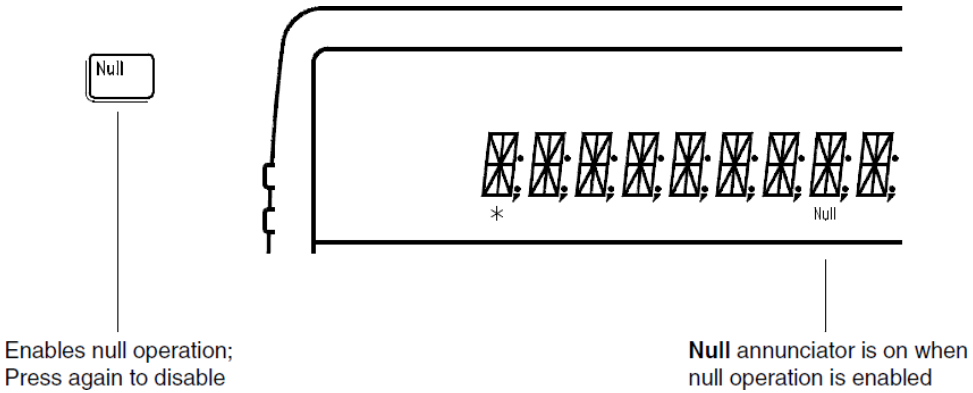
NPLC	Filter Off Max Digits Shown	Filter On Max Digits Shown
0.02	4½	5½
0.2	5½	6½
1	6½	7½
2	6½	7½
10	6½	7½
20	7½	7½
100	7½	7½
200	7½	7½

- You can always show fewer digits than the maximum allowed (the minimum number of digits shown is 4½).

NOTE Integration Time is local to the selected function. This means that you can select the integration time for each function independently. The meter remembers integration time when you switch between functions.

To Make Null (Relative) Measurements

Each null measurement, also called *relative*, is the difference between a stored null value and the input signal. *See page 90*.

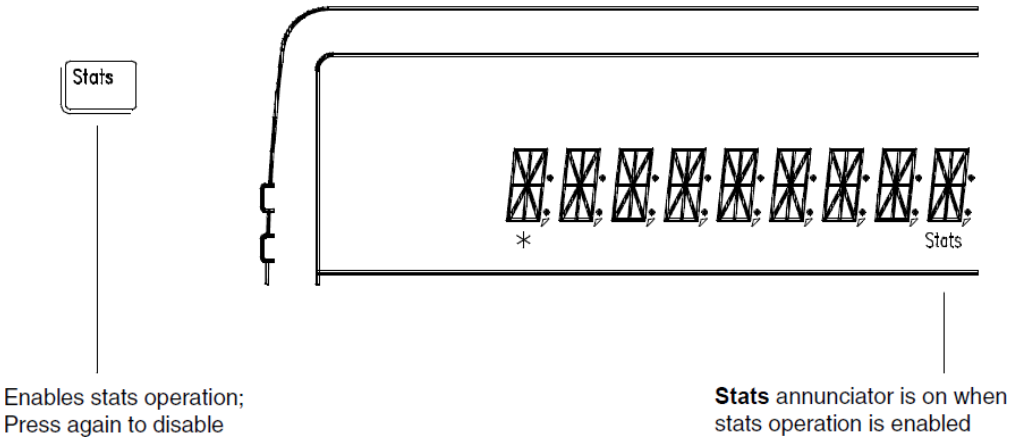


Reading = *measurement – null value*

- You can make null measurements on any function or channel. You can set a null for each channel and function independently. The meter remembers the null value when you switch between input channels or functions.
- To null the test lead resistance for more accurate two-wire ohms measurements, short the ends of the test leads together and then press Null.
- The first reading taken after you press Null is stored as the null value in the Null Register. Any previously stored value is replaced with the new value.
- After enabling null, you can edit the stored null value by pressing Shift > (Menu Recall). This takes you to the "NULL VALUE" command in the MATH MENU (*only if null is enabled*). Go down to the "parameter" level, and then edit the displayed value.

To Store Minimum and Maximum Readings (Stats)

You can store the minimum and maximum readings during a series of measurements. The following discussion shows how to read the minimum, maximum, average, peak-to-peak, standard deviation, and reading count.

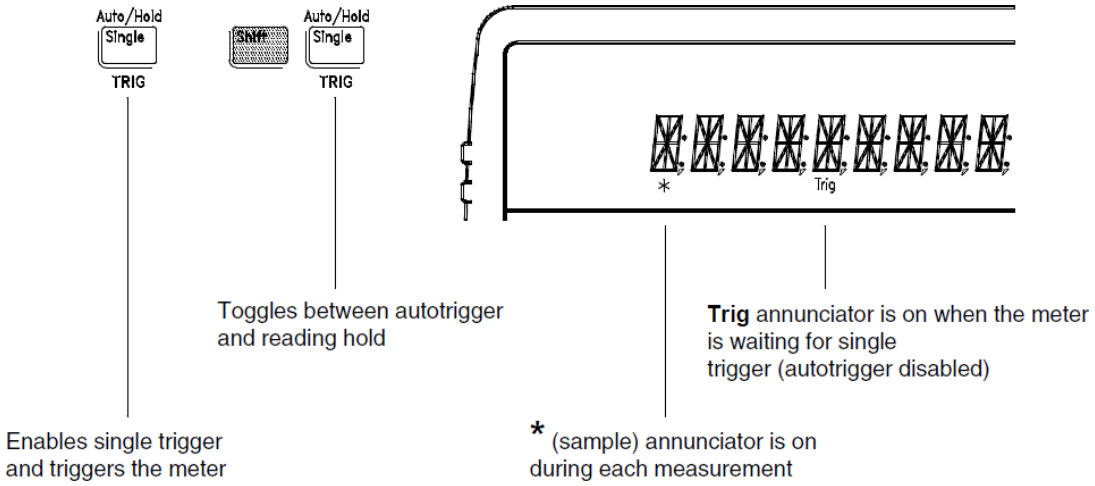


- You can use stats with any function. When you change functions, stats operation is disabled.
- Changing input channels or selecting Scan does not disable stats. The stored readings, however, are not marked as to the input channel.
- After enabling stats, you can read the stored minimum, maximum, average, standard deviation, peak-to-peak, and count by pressing Shift > (Menu Recall). This takes you to the "STATS" command in the MATH MENU (*only if stats is enabled*). Go down to the "parameter" level v, and then read the values by pressing < or >.
- After reading the stored value, pressing Shift < (Menu On/Off) returns to the measurement display and continues stats operations.
- The stored values are cleared when you turn stats on, turn off the power, or perform either a remote interface reset or preset.

– The average and standard deviation are of all readings taken since stats was enabled (not just the average of the stored minimum and maximum). The count is the total number of readings taken since stats was enabled.

To Trigger the Meter

You can trigger the meter from the front panel using *single trigger* or *autotrigger*.



- Autotriggering is enabled when you turn on the meter. Notice that the * (sample) annunciator turns on during each measurement.
- Single triggering takes one reading each time you press Single and then waits for the next trigger. Continue pressing this key to trigger the meter.

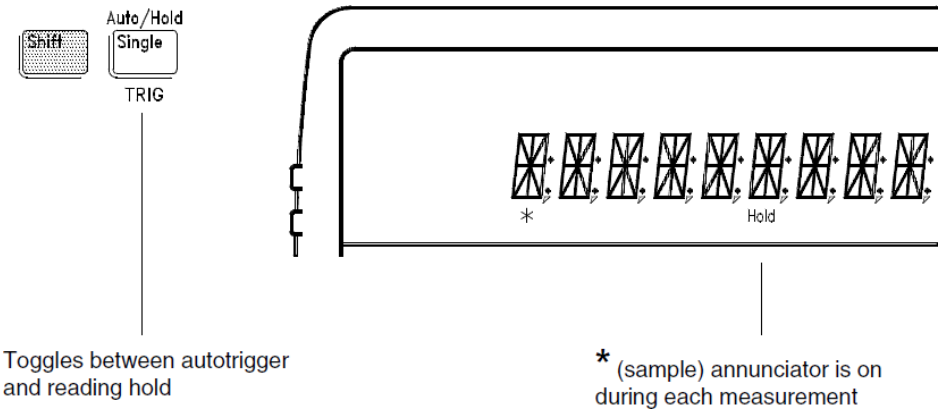
Using an External Trigger

The external trigger mode is also enabled by pressing Single . It is like the single trigger mode except that you apply a trigger pulse to the rear-panel *Ext Trig* terminal. The meter is triggered on the negative edge of a TTL pulse.

The front-panel Single key is disabled when in remote.

To Use Reading Hold

The reading hold feature allows you to capture and hold a stable reading on the display. When a stable reading is detected, the meter emits a beep and holds the value on the display.



- Reading hold has an adjustable *sensitivity band* to allow you to select which readings are considered stable enough to be displayed. The band is expressed as a percent of reading on the selected range. The meter will capture and display a new value only after *three* consecutive readings are within the band.
- The default band is 0.10% of reading. After enabling reading hold, you can choose a different band by pressing Shift > (Menu Recall). This takes you to the "READ HOLD" command in the TRIG MENU (*only if reading hold is enabled*).

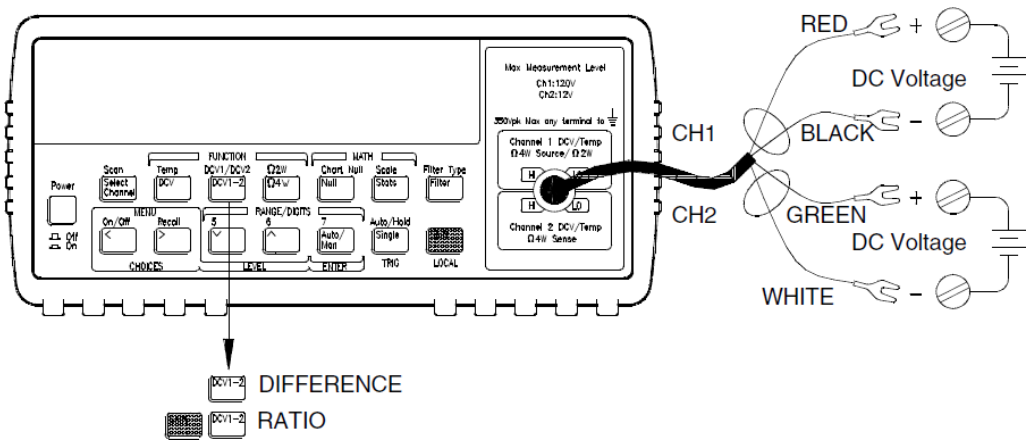
  Go down to the "parameter" level, and then select a value: *0.01%, **0.10%**, 1.00%, or 10.00% of reading*.
- The sensitivity band is stored in volatile memory; the meter sets the band to 0.10% of reading when power has been off or after a remote interface reset.

To Make Voltage Ratio and Difference Measurements

To calculate a voltage ratio, the meter measures the voltages applied to the **Channel 1** input terminals and **Channel 2** input terminals. The calculation is as follows:

$$Difference = ((Ch\ 1\ voltage - Ch\ 1\ Null) - (Ch\ 2\ voltage - Ch\ 2\ Null)) - Null$$ [1]

$$Ratio = \frac{(Ch\ 1\ voltage - Ch\ 1\ Null)}{(Ch\ 2\ voltage - Ch\ 2\ Null)}$$



- Maximum input signal on **Channel 1** is 120 Vdc, the meter will autorange on the actual input or can be manually set.
- Maximum input signal on **Channel 2** is 12 Vdc, the meter will autorange on the actual input or can be manually set.
- Maximum voltage difference between **Channel 1 LO** and **Channel 2 LO** is 150 Vdc.
- Null, if used, is independent for each channel. An additional null[1] of the difference can also be made. If null is used on either input channel, the 'Null' annunciator will flash when the difference function is selected. If you null the difference, the 'Null' annunciator will be steady.

[1] The difference null is only available from the front panel.

To Use Reading Memory

The meter can store up to 1024 readings in internal memory. The following steps demonstrate how to store readings and retrieve them.

1 Select the function.

Select any measurement function. You can change the function at any time during reading memory.

2 Select the single trigger mode.

Single

Notice that the **Trig** annunciator turns on. When reading memory is enabled, readings are stored when you trigger the meter.

For this example, single triggering is used to store readings. You can also use auto triggering or reading hold.

3 Turn on the menu.

Shift On/Off <

```
A: MEAS MENU
```

4 Move across to the SYS MENU choice on this level.

< < <

```
B: SYS MENU
```

5 Move down to a level to the RDGS STORE command.

V

```
1: RDGS STORE
```

6 Move down a level and then across to the "ON" choice.

[v] [>]

```
ON
```

7 Save the change and exit the menu.

[Auto/Man]
ENTER

Notice that the **Mem** (memory) annunciator turns on to indicate that the meter is ready to store readings. Up to 1024 readings can be stored in first-in-first-out (FIFO) order. When memory is full, the **Mem** annunciator will turn off.

Readings are preserved until you re-enable reading memory at another time, turn off the power, or perform a remote interface reset.

8 Trigger the meter three times.

[Single] [Single] [Single]

This stores three readings in memory.

9 Use menu recall to retrieve the stored readings.

Recall
[Shift] [>]

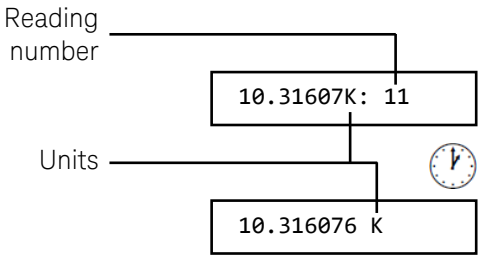
This takes you to the "SAVED RDGS" command in the SYS MENU.

```
2: SAVED RDGS
```

10 Move down a level to view the first stored reading.

Reading memory is automatically turned off when you go to the "parameter" level in the menu.

The first reading displayed is the first reading that was stored (FIFO). If no readings are stored in memory, "EMPTY" is displayed. The first five digits of the reading and the reading number are displayed. After a brief pause, the reading number is removed and all reading digits are shown. The stored readings are displayed with their units (μ, m, k, etc.) when appropriate. For example:



11 Move across to view the two remaining stored readings.

[>] [>]

The readings are stored *horizontally* on the "parameter" level.

If you press [<] *when you first get to the "parameter" level, you will see the last reading and you will know how many readings were stored.*

12 Turn off the menu.

Shift On/Off [<]

```
EXITING
```

3 Features and Functions

Features and Functions 66
General Measurement Configuration 68
Voltage Measurement Configuration 78
Resistance Measurement Configuration 81
Temperature Measurement Configuration 84
Math Operations 87
Null (Relative) Operation 90
Reading Hold 92
Triggering 93
System-Related Operations 103
Voltmeter Complete Terminal 109
External Trigger Terminal 110
Chart Output (Analog Output) 111
Remote Interface Configuration 115
Calibration 121
Defaults, Power-On and Reset States 126

Features and Functions

You will find that this chapter makes it easy to look up all the details about a particular feature of the meter. Whether you are operating the meter from the front panel or from the remote interface, this chapter will be useful.

This chapter is divided into the following sections:

- "General Measurement Configuration" on page 68
- "Voltage Measurement Configuration" on page 78
- "Resistance Measurement Configuration" on page 81
- "Temperature Measurement Configuration" on page 84
- "Math Operations" on page 87
- "Null (Relative) Operation" on page 90
- "Reading Hold" on page 92
- "Triggering" on page 93
- "System-Related Operations" on page 103
- "Voltmeter Complete Terminal" on page 109
- "External Trigger Terminal" on page 110
- "Chart Output (Analog Output)" on page 111
- "Remote Interface Configuration" on page 115
- "Calibration" on page 121
- "Defaults, Power-On and Reset States" on page 126

Some knowledge of the front-panel menu will be helpful before you read this chapter. If you have not already read Chapter 2, "Front-Panel Operation" starting on page 37, you may want to read it now. Chapter 4, "Remote Interface Reference" starting on page 129, lists the syntax for the SCPI (Standard Commands for Programmable Instruments) commands available to program the meter.

NOTE

Throughout this manual, the following conventions are used for SCPI command syntax. Square brackets ([]) indicate optional keywords or parameters. Braces ({}) enclose a list of parameters separated with a vertical bar (|), use only one of the values . Triangle brackets (<>) indicate that you must substitute a value for the enclosed parameter.

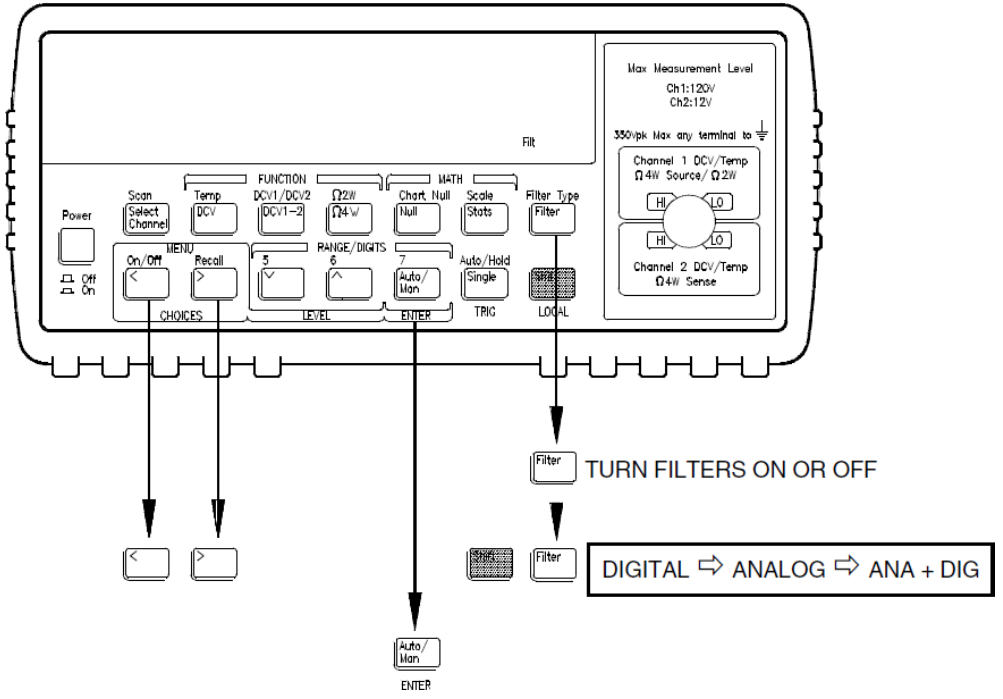
The actual command string you send to the meter should not include any of the conventions indicated above.

General Measurement Configuration

This section contains information to help you configure the meter for making measurements. You may never have to change any of the measurement parameters discussed here, yet they are provided to give you the flexibility you might need.

Input Filters

Two input filters are available; a digital filter and an analog filter. You can enable the digital filter, the analog filter, or both. The digital filter may be used for any measurement type. The analog filter is only available for thermocouple temperature measurements and voltage measurements on the 1 mV, 10 mV, or 100 mV ranges.



- **Front Panel operation**: Press [Shift] [Filter] and scroll to one of DIGITAL, ANALOG, or ANA + DIG. Press [Enter] to enable the filter type. Press the [Filter] key to turn the filter on or off.

  To set the digital filter averaging, use the MEASurement menu **1: DIG FILTER** command.

- **Remote operation**: the following commands turn the filters off, on, set the digital filter count:

```
INPut:FILTer
      :STATe {OFF|ON}
      :TYPE {ANAlog|DIGital|BOTH}
      :DIGital:RESPonse {SLOW|MEDium|FAST}
      :DIGital:PRECharge {ON|OFF}
```

  Using the filters in remote operation is **not recommended**. Use:

```
INPut:FILTer:STATe OFF
```

  to disable the filters. The digital filter is also disabled following a remote reset (*RST) or preset (SYStem:PREset).

  Use the math operation (AVERage) commands to obtain a single reading that is the average of a number of readings (*see page 164*). An example program is shown on page 243 in Chapter 6.

  If you attempt to use the digital filter in remote operations, be aware of the following special conditions:

  The digital filter requires a number of readings to settle (10, 50 or 100). The digital filter readings only occur when the meter is triggered to take a reading. In remote operation a reading is only taken when requested. For example, if you have set the digital filter to MEDIUM (50 readings), you will need to request (trigger) 50 readings from the remote interface before the filter can settle. It is also possible that after 50 readings the digital filter will still not be settled as described in the next paragraph.

  The STATus:OPERational:CONDition register contains a bit that indicates the settled/unsettled state of the digital filter on a per reading basis. *See page 184 in Chapter 4*. You should check the state of this bit for each reading to ensure the digital filter count has not reset (filter reset operation is described in the next section).

Digital Filter

- The digital filter is a moving average (boxcar) filter. Equal weighting is applied to all readings to calculate the displayed reading.
- Three digital filters are available:

  SLOW (average last 100 readings),

  MEDIUM (average last 50 readings), or

  FAST (average last 10 readings)
- When the digital filter is enabled, the 'Filt' annunciator flashes until the required number of readings are obtained. The measurement displayed before the 'Filt' annunciator is steady is the average of the number of readings taken so far. For example, with the filter set to MEDIUM (average of 50 readings), the measurement displayed after 20 readings will be the average those 20 readings. The 'Filt' annunciator will flash until 50 readings have been taken.
- The filter count resets (and the 'Filt' annunciator will flash) if the function, range, integration time, or channel is changed.
- The filter count resets if a reading is outside a set range of the average. The amount of input reading change required to reset the filter count is:

	1mV Range	All Other Ranges
FAST	± 400 ppm of range	± 100 ppm of range
MEDIUM	± 700 ppm of range	± 300 ppm of range
SLOW	± 2000 ppm of range	± 1000 ppm of range

Analog Filter

The analog filter is an 11 Hz, two-pole low pass filter. Use this filter to remove line frequency noise from the input measurement.

- The filter requires approximately 300 milliseconds after the input is present to settle. Using the filter, therefore, slows the measurement rate significantly.
- Use the analog filter only when line power noise (50/60 Hz) is causing an overload (OVLD) condition. *See page 77*.
- The analog filter is available for voltage and thermocouple temperature measurements. The analog filter is not available for resistance measurements.

– The analog filter is available for the 1 mV, 10 mV, and 100 mV ranges.

Integration Time

Integration time is the period during which the meter's analog-to-digital (A/D) converter samples the input signal for a measurement. Integration time affects the measurement resolution (for better resolution, use a longer integration time), and measurement speed (for faster measurements, use a shorter integration time).

– Integration time is specified in *number of power line cycles* (NPLCs). The choices are 0.02, 0.2, 1, 2, **10**, 20, 100, or 200 power line cycles. *The default is 10 NPLCs*.

– The integration time is stored in *volatile* memory; the meter selects 10 NPLC when power has been off or after an interface reset.

– Only the integer number of power line cycles (1, 2, 10, 20, 100 or 200 NPLC) provide normal mode (line frequency noise) rejection.

– Integration time is related to the maximum number of digits the meter will display. You can always choose to show fewer digits than the maximum allowed (the minimum number of digits shown is 4½).

NPLC	Digital Filter Off Max Digits Shown	Digital Filter On Max Digits Shown
0.02	4½	5½
0.2	5½	6½
1	6½	7½
2	6½	7½
10	6½	7½
20	7½	7½
100	7½	7½
200	7½	7½

– **Front Panel operation**: Integration time can be set indirectly when you select the number of digits (*see page 54*).

  You can also set the integration time in the MEASurement menu with the 2: INTEGRATE command.

– **Remote operation**: Refer to the table on page 144. Use one of the following commands:

```
[SENSe:]
     VOLT:DC:NPLC {0.02|0.2|1|2|10|20|100|200|MIN|MAX}
     FRES|RES
          :NPLC {0.02|0.2|1|2|10|20|100|200|MIN|MAX}
     TEMP:NPLC {0.02|0.2|1|2|10|20|100|200|MIN|MAX}
```

  Integration time and resolution are related. The CONFigure and MEASure? commands can also set the integration time, see page 144.

NOTE **Integration Time is local to the selected function. This means that you can select the integration time for each function independently. The meter remembers the integration time when you switch between functions.**

Reducing Measurement Noise

At high resolutions and low measurement levels, measurement noise can become an important factor in the accuracy of your measurements. One possible source of measurement noise is the wiring and cabling in the test setup. These noise sources and their effects are discussed in Chapter 7, "Measurement Tutorial". A smaller level of noise is also inherent in the meter itself.

To minimize the effects of noise on your measurements, use the highest possible integration time (NPLC) and average the readings. Faster reading rates are more susceptible to noise. Choosing the best NPLC and averaging method for your measurement requires a tradeoff between the reading rate and the level of acceptable noise in the resulting measurement.

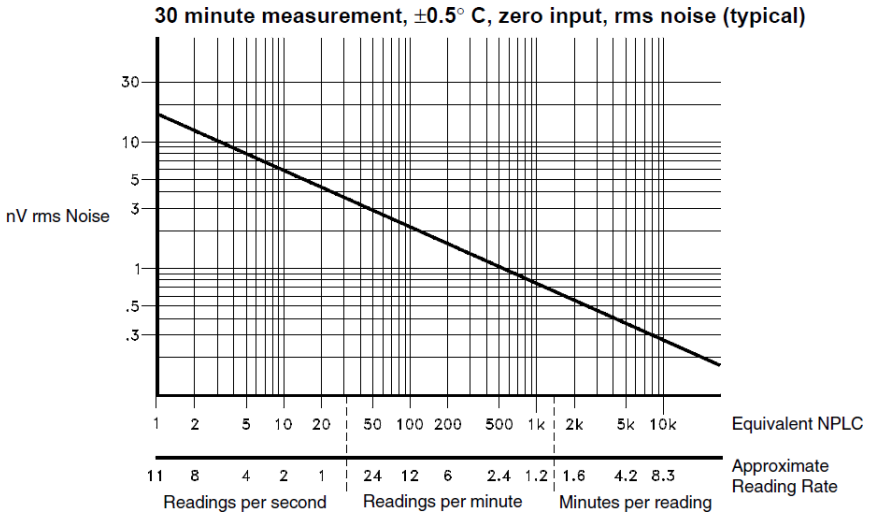
The Integration Time (NPLC) and averaging can be multiplied to create an *equivalent* NPLC. The equivalent NPLC defines the reading rate and characterizes the expected noise level. Averaging can be performed in the meter by using either the digital filter or by using the statistics (Stats) mode of operation.

Measurement noise characteristics of the meter closely match theoretical gaussian noise distributions. The graph on the next page shows *typical* input

noise measurements for the 1 mV range with the low thermal shorting plug applied to the input terminals. Other functions and ranges will exhibit similar noise reduction characteristics versus NPLC.

For example, if you set the NPLC to 20 and enable the digital filter set to FAST (10 reading average), the equivalent NPLC is 200. At an equivalent NPLC of 200, the reading rate is approximately 6 readings per minute (digital filter fully settled) and the typical noise level expected is 1.6 nVrms.

Alternately, if you take readings every 5 minutes, you can set an equivalent NPLC of 5000 to minimize the noise in the measurement. An equivalent NPLC of 5000 could be obtained by setting the integration time to 100 NPLC and enabling the digital filter set to MEDIUM (50 reading average). At this equivalent NPLC, a typical noise level of 0.35 nVrms can be expected.



– **Front Panel operation**: Integration time can be set indirectly when you select the number of digits (*see page 54*).

  You can also set the integration time in the MEASurement menu with the **2: INTEGRATE** command.

  Enable the digital filter and set the filter reading average as described on page 68.

– **Remote operation**: Set the integration time for the measurement function. If possible, use NPLC 200. If you use the CONFigure or MEASure commands, refer to the table on page 144.

  Use the math operations (AVERage) to average the readings or average the readings in your program. An example program that uses the math operations to average readings is shown on page 243 in Chapter 6. *Do not use the digital filter to average readings, see page 69.*

Number of Digits Displayed

Front Panel operation only. See also "Integration Time" on page 71.

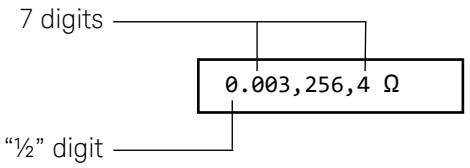
– You can set the number of digits shown in the display to 4, 5, 6, or 7 *full digits*, plus a "½" digit which can only be a "0" or "1".

– You can set both the number of digits displayed and the Integration Time in a single operation by pressing digits keys (*see page 54*).

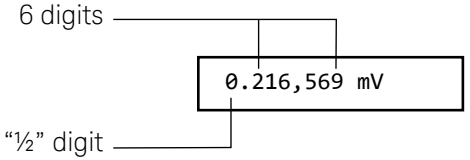
– You can change the number of digits displayed without disturbing the Integration Time settings by pressing the < or > keys (*see page 54*).

– The number of digits is set to 6½ digits at power-on and after either a remote interface reset or preset.

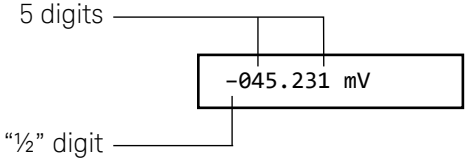
– The number of digits shown is dependent upon integration time and filter settings. The meter will not allow you to show more digits than the practical measurement capability of the meter. You can, however, reduce the number of digits shown. The minimum number of digits available is 4½. (*See the table on page 71*).



This is the 1 ohm range, 7½ digits are displayed.



This is the 1 mV range, 6½ digits are displayed.



This is the 100 mV range, 5½ digits are displayed.

Ranging

You can let the meter automatically select the range using *autoranging* or you can select a fixed range using *manual ranging*. Autoranging is convenient because the meter automatically selects the appropriate range for each measurement. However, you can use manual ranging for faster measurements since the meter does not have to determine which range to use for each measurement.

- The selected mode (auto or manual range) is stored in *volatile* memory; the meter returns to autoranging when power has been off or after a remote interface reset.
- Autorange thresholds:

  Down range at <10% of range

  Up range at >120% of range
- If the input signal is greater than the present range can measure, the meter gives an *overload indication*: "OVLD" from the front panel or "9.90000000E+37" from the remote interface.
- For ratio and difference measurements, the specified range is independent for both **Channel 1** and **Channel 2**.
- **Front Panel operation**: Press a front-panel RANGE key to set manual ranging and to up or down range. Press the Auto/Man key to return to autoranging. *See also "To Select a Range"* on page 53.

  [∧] [∨] [Auto/Man]
- **Remote operation**: Range is set with parameters in the MEASure and CONFigure subsystems. For example:

  ```
  CONF:VOLT:DC 10.0              Sets the 10 Volt DC range
  ```

  Range can be set directly with the SENSe subsystem. For example:

  ```
  SENS:VOLT:DC:RANGe 0.001       Sets the 1 mV DC range
  SENS:RES:RANGE:AUTO            Sets auto ranging on 2-wire
                                 resistance measurements
  ```

Overload Detection

The 34420A uses both an analog and digital method to generate an overload condition and display the OVLD message. An understanding of the cause of the overload can help you make more accurate measurements.

Digital Overload Detection

The digital overload detect occurs when the magnitude of the measurement is greater than 120% of the Range. For example, on the 1 volt Range, signal levels of greater than ± 1.2 volts will cause an OVLD in the display.

Analog Overload Detection

Another type of overload can result in measurement errors that are hard to identify. For example if an overload signal only occurs for a portion of the measurement period, the input amplifier could become non-linear resulting in an inaccurate reading. The displayed result may appear to be correct because the overload signal only occurred during a small portion of the measurement cycle. The 34420A detects this problem by providing comparators on the input amplifier. If signals large enough to overload the input amplifier are detected, the display will read OVLD. The comparators detect signals greater than 140% of the range setting.

If an overload occurs, two solutions are possible.

- Setting the meter to the next higher range will provide 10 times improvement in the overload signal handling capability. For example,if you are measuring on the 1 mV range and have 2 mV rms of power line (50 or 60 Hz) pickup, the noise overload comparators would sense the peaks of the signals and an OVLD message would be displayed. If you set the range to 10 mV, the resolution (with 7 digits) is still 1 nanovolt, and the reading rate is faster than if ANALOG FILTER had been turned on. Additionally, the 10 mV range noise is almost as low as the 1 mV range.
- Alternately, if using the 1 mV, 10 mV, or 100 mV ranges, turn on the ANALOG FILTER. The reading rate slows down with the filter on. In the case of a larger overloads (for example, 20 mV rms noise on the 1 mV range) using the ANALOG FILTER would give a lower noise result than changing to the 100 mV range.

Voltage Measurement Configuration

Input Channels

The meter has two independent input channels for measuring dc Volts. You can make measurements on either or both channels, measure the difference between the channels, or measure the ratio between the channels. Each channel has an independent settings for null and range.

The figure on page 27 shows connections for two input channels.

- **Channel 1** is the default input channel. Both channels use the same integration time setting.
- The maximum measurable input on **Channel 1** is 120V, on **Channel 2** is 12 V.
- For two channel operation, the maximum voltage difference between **Channel 1** LO and **Channel 2** LO is ±150 V.
- For either channel, the maximum non-destructive input is ±150 V.
- If you enable input filters, the filters are active for both input channels. Switching between the channels will reset the digital filter reading count (*see page 70*).
- **Front Panel operation**: Toggle the input channel by pressing the Channel key on the front panel. The **CH1** or **CH2** annunciator will light in the display. The meter keeps range and null settings independently for each channel.

  Channel

  Choose the scan mode to alternately show the **CH1** and **CH2** readings. Disable the digital filter when using the scan mode. In scan mode, the meter alternates between the **CH1** and **CH2** readings every 3 seconds or for each measurement (whichever is longer).

  Shift Scan Channel

– **Remote operation**: The CONFigure and MEASure subsystems use an optional parameter to specify the input channel as either 1 or 2:

```
CONF:VOLT:DC DEF, DEF,(@FRONT1)      Channel 1 operation
CONF:VOLT:DC DEF, DEF,(@FRONT2)      Channel 2 operation
```

The SENSe subsystem uses a keyword modification to indicate the channel:

```
SENS1:VOLT:DC:RANG:AUTO        Channel 1 operation
SENS2:VOLT:DC:RANG:AUTO        Channel 2 operation
```

Two Channel Measurements

– **Difference measurements**: First, set the range and null value for each input channel. Then, select difference measurements.

Ch1 - Ch2

$$Difference = ((Ch\ 1\ voltage - Ch\ 1\ Null) - (Ch\ 2\ voltage - Ch\ 2\ Null)) - Null$$ [1]

– **Ratio measurements**: First, set the range and null value for each input channel. Then, select ratio measurements.

Shift Ch1 - Ch2

$$Ratio = \frac{(Ch\ 1\ voltage - Ch\ 1\ Null)}{(Ch\ 2\ voltage - Ch\ 2\ Null)}$$

– **Remote operation**: use the following commands to measure or configure a difference or ratio measurement. Set each channel to the appropriate range and null conditions using:

```
[SENSe1:|SENSe2:]
     VOLTage:DC
          :RANGE <range>
          :NULL:STATe {OFF|ON}
          :NULL:VALue {<value>|MIN|MAX}
```

[1] The difference null is only available from the front panel.

Then use the following commands to measure the ratio or difference:

```
MEASure:VOLTage:DC
+     :RATIO?
     :DIFFerence?
CONFigure:VOLTage:DC
     :RATIO
     :DIFFerence
[SENSe:]FUNCtion "VOLTage:RATio"
[SENSe:]FUNCtion "VOLTage:DIFFerence"
```

Resistance Measurement Configuration

The meter can make 2-wire or 4-wire ohms measurements. The meter can compensate for voltages in the resistive circuit being measured using offset compensation. You can also choose to either limit the power applied or limit the open circuit voltage applied during a resistance measurement.

The test current sourced by the meter depends upon the range and conditions as shown below:

Range	Test current		
	2-wire or 4-wire measurements	4-wire measurements	
		Low Power	Voltage Limited
1 Ω	10 mA	10 mA	Not Available
10 Ω	10 mA	10 mA	1 mA
100 Ω	10 mA	1 mA	100 µA
1 kΩ	1 mA	100 µA	Not Available
10 kΩ	100 µA	10 µA	Not Available
100 kΩ[a]	10 µA	5 µA	Not Available
1 MΩ[a]	5 µA	5 µA	Not Available

[a] Offset Compensation is not applied to measurements on these ranges

Offset Compensated Measurements

Offset compensation removes the effect of any voltages in the circuit being measured. The technique involves taking two measurements, one with the current source turned on, and one with the current source turned off and computing the difference. *A description of this operation is given on page 284*.

- **Front Panel operation**: Offset compensation is on at power on. Turn offset compensation off under the **3: OCOMP Ω** command in the MEASure menu.
- **Remote operation**: Offset compensation is off following a remote reset or preset. Use the following command to change the state of offset compensation:

```
SENSe:
     FRESistance:OCOMpensated {OFF|ON}
     RESistance:OCOMpensated {OFF|ON}
```

Low Power Measurements

Low Power measurements use a reduced output measurement current to limit the power dissipation in the circuit being measured.

- Low Power measurements apply only to 4-wire ohms measurements.
- When the low power measurement mode is enabled, the 'LP' annunciator is on in the display.
- Setting the Voltage Limited measurement mode will cancel the Low Power mode.
- **Front Panel operation**: set low power mode under the **4: LOW POWER Ω** command in the MEASure menu.
- **Remote operation**: use the following command to set the low power ohms mode:

```
SENSe:FRESistance:POWer:LIMit {OFF|ON}
```

Voltage Limited Measurements

Voltage limited resistance measurements clamp the open circuit voltage to predefined limits.

- Voltage limits apply only to 4-wire ohms measurements.
- Voltage limited resistance measurements are only available on the 10 Ω and 100 Ω ranges. If a manual range is selected in either 2-wire or 4-wire ohms, and then Voltage Limited measurements are enabled, the range will be set to the highest valid range (either 10 Ω or 100 Ω).
- The open circuit voltage can be limited to 20 mV, 100 mV, or 500 mV.
- Setting the Low Power measurement mode will cancel the Voltage Limited mode.
- **Front panel operation**: Enable or disable the voltage limit under the **5: LOW VOLT Ω** command in the MEASure menu.

  Set the voltage limit under the **6: LoV LIMIT Ω** command in the MEASure menu.
- **Remote operation**: the following command enables or disable the low voltage limit mode:

  `SENSe:FRESistance:VOLTage:LIMit {OFF|ON}`

  Set the limit with the following command:

  `SENSe:FRESistance:VOLTage:LIMit:VALue {<value>|MIN|MAX}`

Temperature Measurement Configuration

This section contains information to help you configure the meter for making temperature measurements. To measure temperature, you will need a temperature transducer. Transducer descriptions and specific comments about their usage are given on page 288. The meter supports three general categories of transducer:

- RTD (Resistive Temperature Device)
- THERM (Thermistor)
- T/C (Thermocouple)

The meter supports specific types of transducers in each category:

RTD's R_0 = 4.9 Ω to 2.1 kΩ	Thermistor	Thermocouples
α = 0.00385 (DIN/IEC 751) α = 0.00391	5k Ω	B, E, J, K, N, R, S, T

Once you set the measurement units, probe type, and transducer type, the setting is stored in *non-volatile* memory. This makes the last probe type used the default probe type.

Measurement Units

- The meter can report temperature in °C (Celsius), °F (Fahrenheit), or K (Kelvins).
- The measurement units set are stored in *non-volatile* memory.
- **Front Panel Operation**: Set the units under the **2: UNITS** command in the TEMPerature menu.
- **Remote operation**: Set the units using the following command:

```
UNIT:TEMPerature {C|F|K}
```

RTD Measurements

- The meter supports RTDs with α = 0.00385 (DIN/IEC 751) or α = 0.00391. The nominal value of the RTD (R_0) must be in the range of 4.9 Ω to 2.1 kΩ. *RTD Connections are shown on* page 29.
- The meter makes a 4-wire measurement for RTD's. All four input leads must be connected.
- **Front Panel operation**: Set RTD as the probe type under the **1: PROBE TYPE** command in the TEMPerature menu.

  Select the RTD type under the **3: RTD TYPE** command in the TEMPerature menu.

  Enter the 0 °C resistance of the RTD (R_0) in the **4: RTD R0** command in the TEMPerature menu.
- **Remote operation**: Use the following commands for temperature measurements with RTD's:

```
CONFigure:TEMPerature FRTD,{85|91}[,1,<resolution>]
MEASure:TEMPerature? FRTD, {85|91}[,1,<resolution>]
SENSe:FUNCtion "TEMPerature"
SENSe:TEMPerature:TRANsducer:TYPE FRTD
SENSe:TEMPerature:TRANsducer:FRTD:TYPE {85|91}
SENSe:TEMPerature:TRANsducer:FRTD:RESistance <value>
```

Thermistor Measurements

- Only the 5 kΩ thermistor type is supported. *Thermistor connections are shown on* page 28.
- The thermistor must be connected to the Channel 1 inputs.
- **Front Panel operation**: Set THERM as the transducer type under the **1: PROBE TYPE** command in the TEMPerature menu.
- **Remote operation**: use the following commands for temperature measurements with a thermistor:

```
CONFigure:TEMPerature THER, DEF,[,1,<resolution>]
MEASure:TEMPerature? THER, DEF,[,1,<resolution>]
SENSe:FUNCtion "TEMPerature"
SENSe:TEMPerature:TRANsducer:TYPE THER
```

Thermocouple Measurements

- The thermocouple must be connected to the **Channel 2** inputs. Be sure to observe the correct polarity. *Thermocouple connections are shown on page 30*.
- Thermocouple measurements require a reference junction temperature. You can input a known fixed junction temperature, use an external thermistor temperature, or use the internal thermistor temperature as the reference junction temperature.

NOTE The accuracy of the measurement is very dependent upon the thermocouple connections and the reference junction temperature. Use a fixed temperature reference for highest accuracy measurements. The internal thermistor reference produces the lowest accuracy measurements. See page 290 for a discussion of the reference junction temperature measurements.

- **Front Panel operation**: Set T/C as the probe type under the 1: PROBE TYPE command in the TEMPerature menu.

  Select the thermocouple type under the 4: T/C TYPE command in the Temperature menu.

  Set the reference junction temperature source under the 5: COLD JUNCT command in the TEMPerature menu.

  If you used a FIXed reference junction temperature, set the reference temperature under the 6: JUNCT TEMP command in the TEMPerature menu. The number entered must be in the range of –1 °C to 55 °C.

- **Remote operation**: use the following commands for temperature measurements using thermocouples:

```
CONFigure:TEMPerature TC,{B|E|J|K|N|R|S|T}[,1,<resolution>]
MEASure:TEMPerature? TC,{B|E|J|K|N|R|S|T}[,1,<resolution>]
SENSe:
     FUNCtion "TEMPerature"
     TEMPerature:TRANsducer
          :TYPE TC
          :TCouple
               :TYPE {B|E|J|K|N|R|S|T}
               :RJUNction:TYPE {FIXed|INTernal|THERmistor}
               :RJUNction {<reference>|MIN|MAX}
```

Math Operations

There are two math operations available, only one of which can be enabled at a time. You can choose to either scale the readings as they are taken, or keep statistics on a group of readings. The selected math operation remains in effect until you disable it, change functions, turn off the power, or perform a remote interface reset.

Scaling

– The scaling is applied as:

*reading = gain * (measurement - offset)*

– **Front Panel operation:** Set the values of *gain* and *offset* in the MATH menu. Enable scaling by pressing the scale key. The Stats annunciator will light in the display.

Shift Scale (Stats)

– **Remote operation:** the math operations and registers are controlled using commands within the CALCulate command subsystem. First, select the math operation you want to use

```
CALCulate:FUNCtion SCALe
```

Then, set the values for *gain* and *offset*.

```
CALCulate:SCALe:GAIN <gain>
```

```
CALCulate:SCALe:OFFSet <offset>
```

Finally, enable the selected math function by turning the math state on:

```
CALCulate:STATe ON
```

Statistics

- After you enable statistics, the first reading that the meter takes is stored as both the minimum and maximum value. The *minimum* is replaced with any subsequent value that is less. The *maximum* is replaced with any subsequent value that is greater.
- All values are stored in volatile memory; the meter clears the values when stats is turned on, when power has been off, or after a remote interface reset or preset.
- When enabled, the following statistics are available:

MIN	minimum reading
MAX	maximum reading
AVER	arithmetic average of all readings
STD DEV	standard deviation of all readings
PEAK – PEAK	peak to peak value of minimum and maximum
COUNT	the total number of readings

- **Front Panel operation:** Enable the statistics by pressing the [Stats] key. Read the statistics gathered in the MATH menu under the STATS command.

  After reading the stored statistics, press [Shift] [<] (Menu On/Off) to return to the measurement and continue taking statistics.

- **Remote operation:** the math operations and registers are controlled using math operation commands. First, select the math operation you want to use

```
CALCulate:FUNCtion AVERage
```

  Then, enable the selected math function by turning the math state on:

```
CALCulate:STATe ON
```

  When you have finished collecting readings, obtain the desired statistics:

```
CALCulate:AVERage
     :MINimum?
     :MAXimum?
     :AVERage?
     :SDEViation?
     :PTPeak?
     :COUNt?
```

  After reading the statistics, supply another trigger to continue taking measurements and collecting statistics. Measurements are not taken while reading the results.

If you are interested in the statistics on a number of readings, but not the readings themselves, you can use the following command:

```
DATA:FEED RDG_STORE, ""
```

The null value parameter in the command tells the meter not to store the individual readings. The meter does, however, accumulate statistics on the readings. This method removes the 1024 reading limitation and you can calculate statistics on any number of readings. *See page 166 for more information.*

Null (Relative) Operation

When making null measurements, also called *relative*, each reading is the difference between the input signal and a stored null value. You could, for example, make a more accurate two-wire ohms measurement by shorting the test leads and pressing Null to remove the test lead resistance. *See also "To Make Null (Relative) Measurements"* on page 56.

Reading = *measurement – null value*

- The null value is adjustable and you can set it to any value between 0 and ± 120% of the highest range, for the present function.
- The null value is stored in *volatile* memory; the value is cleared when power has been off or after a remote interface reset.
- Each function and channel has an independent null value. For voltage measurements, each input channel has an independent null. Temperature measurement null is applied to all temperature measurements (if you change the probe type, null is still enabled and applied). The null for resistance measurements applies to both 2-wire and 4-wire measurements.
- The null value is stored in the meter's Null Register. There are two ways you can specify the null value. First, you can enter a specific number into the register from the front-panel menu or from the remote interface. Any previously stored value is replaced with the new value. *If you are operating the meter from the front panel, entering a null value also turns on the null function.*

  The second way to enter the null value is to let the meter store the first reading in the register. Press the Null key to make the displayed measurement the null value and enable the Null operation. After you enable null, the first reading displayed will be zero. If you entered a number into the register, as described in the paragraph above, the first reading *does not* overwrite the stored value.
- Since an independent null value is stored for each input channel, difference or ratio functions also use these null values. Additionally, when using the difference function, an additional difference null can be applied (the difference null is only available from the front panel). See page 79 for a description of how null is applied to difference and ratio functions.

- **Front-panel operation:** After enabling null, you can edit the stored null value by pressing Shift > (Menu Recall). Any previously stored value is replaced with the new value. Turning on the menu does not disable the null operation; the meter will resume taking measurements when you turn off the menu.

  When null is enabled, the '**Null**' annunciator lights in the display. For ratio or difference measurement, if either input channel has a null enabled, the **Null** annunciator will flash. If you apply the difference null, the Null annunciator will light (not flash).

- **Remote interface operation:** You can use the following commands to make null measurements. *Null must be enabled before you can store a value in the Null Register.*

```
SENSe1 | SENSe2
     :VOLTage:DC
          :NULL {OFF|ON}
          :NULL {<value>|MIN|MAX}
[SENSe:]
     FRESistance|RESistance
          :NULL {OFF|ON}
          :NULL {<value>|MIN|MAX}
     TEMPerature
          :NULL {OFF|ON}
          :NULL {<value>|MIN|MAX}
```

  To enable null and set a null value (0.10) for the 2-wire resistance function execute the following commands in the order shown.

```
SENS:RES:NULL ON
SENS:RES:NULL 0.10
```

  You can use the first reading as a null value from the remote interface by first setting the function, range, and resolution and then executing the following command:

```
[SENSe:]NULL [{OFF|ON|ONCE}]
```

Reading Hold

The reading hold feature allows you to capture and hold a stable reading on the front-panel display. This is especially useful in situations where you want to take a reading, remove the test probes, and have the reading remain on the display. When a stable reading is detected, the meter beeps and holds the reading on the display.

The reading hold feature is available only from the front panel. If you go to remote when reading hold is enabled, the meter ignores it; when you return to local (front panel), reading hold is enabled again.

- Reading hold has an adjustable *sensitivity band* (adjustable only from the front panel) to allow you to select which readings are considered stable enough to be displayed. The band is expressed as a percent of reading, on the selected range. The meter will capture and display a new value only after *three* consecutive readings are within the band.

  Select one of these values: 0.01%, **0.10%** (default), 1.00%, or 10.0% of reading.

  For example, assume that the 1.00% band is selected and a 5 volt signal is applied to the meter. If three consecutive readings are between 4.95 volts and 5.05 volts, the meter will beep and display a new reading.

- The sensitivity band is stored in *volatile* memory; the meter sets the band to 0.10% when power has been off or after an interface reset.

- If the meter is in *autorange* when you enable reading hold, it will autorange to the correct range. If the meter is in the *manual range* mode, the same fixed range will be used for reading hold.

- For certain applications, it may be useful to use reading hold with reading memory. *See also "Reading Memory"* on page 103.

- Front Panel operation: After enabling reading hold, you can select a different sensitivity band by pressing Shift > (Menu Recall).

  1: READ HOLD

Triggering

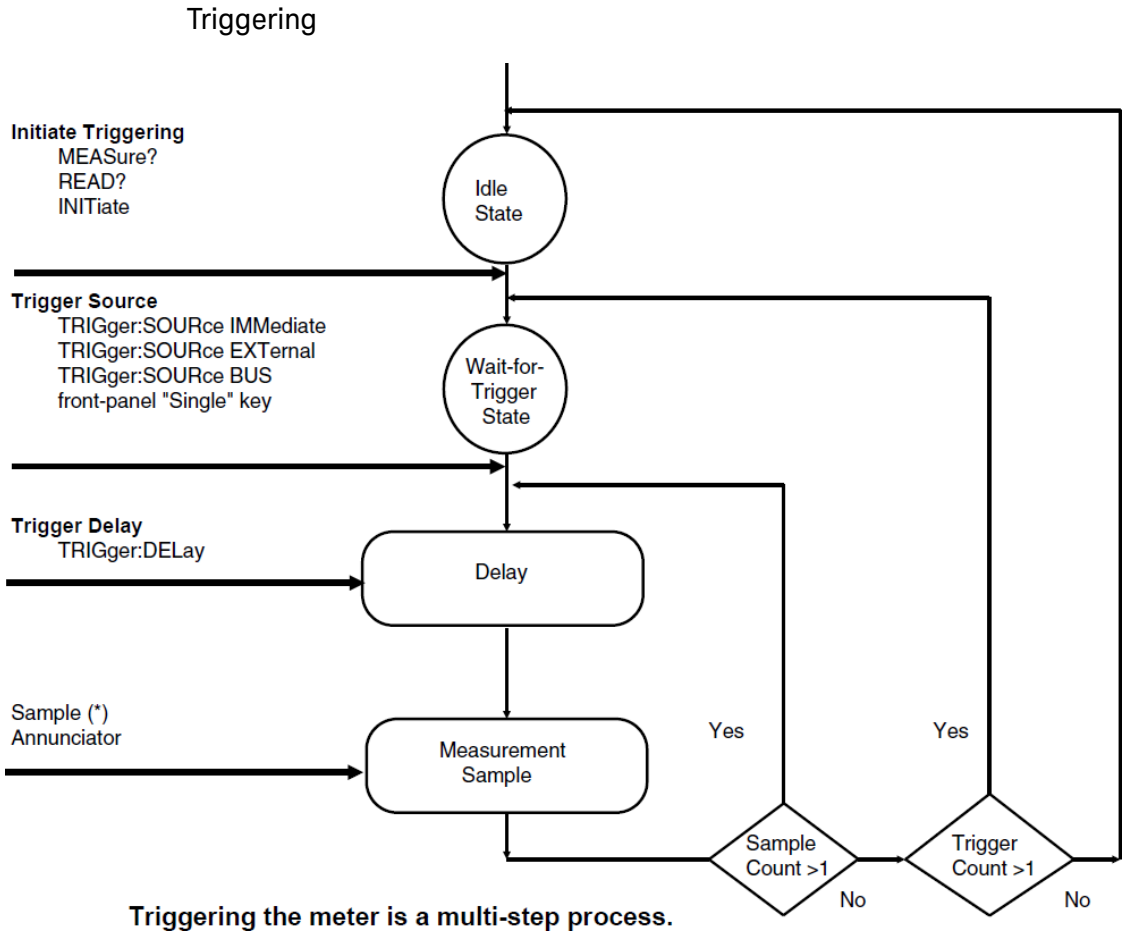
The meter's triggering system allows you to generate triggers either manually or automatically, take multiple readings per trigger, and insert a delay before each reading. Normally, the meter will take one reading each time it receives a trigger, but you can specify multiple readings (up to 50,000) per trigger.

- You can trigger the meter from the front panel, with an external trigger, or use autotriggering. Single triggering takes one reading each time you press the Single key. External triggering is like single triggering, but the meter waits for a pulse on the rear-panel Ext Trig (external trigger) terminal before taking a reading. Autotriggering takes continuous readings at the fastest rate possible for the present configuration. *See also "To Trigger the Meter"* on page 59.
- Press Shift Auto/Hold to return the meter to auto triggering.

Triggering the meter from the remote interface is a multi-step process that offers triggering flexibility.

- First, you must configure the meter for the measurement by selecting the function, range, integration time, etc.
- Then, you must specify the source from which the meter will accept the trigger. The meter will accept a software (bus) trigger from the remote interface, a hardware trigger from the *Ext Trig* terminal, or an immediate internal trigger.
- Then, you must make sure that the meter is ready to accept a trigger from the specified trigger source (this is called the *wait-for-trigger* state).
- Finally, to make the measurement, you must provide an INITiate, READ?, or MEASure command.

The diagram on the next page shows the meter's triggering system.



Triggering the meter is a multi-step process.

Trigger Source Choices

You must specify the source from which the meter will accept a trigger.

- The trigger source is stored in *volatile* memory; the source is set to autotrigger (front panel) or immediate (remote interface) when power has been off or after a remote interface reset.
- **Front Panel operation:** the meter will accept a either a front panel single trigger or a hardware trigger from the *Ext Trig* terminal, or continuously take readings using autotrigger. At power-on, autotriggering is used.
- **Remote operation:** the meter will accept a software (bus) trigger, a hardware trigger from the *Ext Trig* terminal, or an immediate internal trigger. *See "Triggering" on page 93 for more information.*

Autotriggering In the autotrigger mode (front panel only), the meter continuously takes readings at the fastest rate possible for the present configuration. This is the power-on trigger source for front-panel operation.

Single Triggering In the single trigger mode (front panel only), you can manually trigger the meter by pressing Single. The meter takes one reading, or the specified number of readings (sample count), each time you press the key. The '**Trig**' annunciator turns on when the meter is waiting for a trigger.

Pressing the Single key also allows the meter to accept triggers on the *Ext Trig* terminal.

The front-panel Single key is disabled when in remote.

External Triggering In the external trigger mode, the meter will accept a hardware trigger applied to the *Ext Trig* terminal. The meter takes one reading, or the specified number of readings (sample count), each time *Ext Trig* receives a low-true pulse.

See also "External Trigger Terminal" on page 110.

- The meter buffers one external trigger. This means that if the meter is taking a reading and another external trigger occurs, that trigger is accepted (a "Trigger ignored" error is not reported). After the reading in progress is complete, the stored trigger satisfies the trigger source and then the trigger is issued.
- **Front-panel operation:** The external trigger mode is like the single trigger mode except that you apply the trigger to the *Ext Trig* terminal. Pressing Single to enable the single trigger mode also enables the external trigger mode. The '**Trig**' annunciator turns on when the meter is waiting for an external trigger.

  The front-panel Single key is disabled when in remote.
- **Remote operation:** Use the following command to enable external trigger.

  ```
  TRIGger:SOURce EXTernal
  ```

Internal Triggering In the internal trigger mode (remote interface only), the trigger signal is always present. When you place the meter in the wait-for-trigger state, the trigger is issued immediately. This is the power-on trigger source for remote interface operation.

To select the internal trigger source, send the following command. The CONFigure and MEASure? commands automatically set the trigger source to IMMediate.

```
TRIGger:SOURce IMMediate
```

After setting the source to IMMediate, an INITitate or a READ? command will trigger the measurement.

Software (Bus) Triggering The bus trigger mode is available only from the remote interface. This mode is similar to the single trigger mode from the front panel, but you trigger the meter by sending a bus trigger command.

- To select the bus trigger source, send the following command.

  ```
  TRIGger:SOURce BUS
  ```

- To trigger the meter from the remote interface (GPIB or RS-232), send the *TRG (trigger) command. The *TRG command will not be accepted unless the meter is in the wait-for-trigger state.
- You can also trigger the meter from the GPIB interface by sending the IEEE-488 Group Execute Trigger (GET) message. The meter must be in the wait-for-trigger state. The following statement shows how to send a GET from a Hewlett-Packard controller.

  ```
  TRIGGER 722 (group execute trigger)
  ```

The Wait-for-Trigger State

After you have configured the meter and selected a trigger source, you must place the meter in the *wait-for-trigger* state. A trigger will not be accepted until the meter is in this state. If a trigger signal is present, and if meter is in the "wait-for-trigger" state, the measurement sequence begins and readings are taken.

The "wait-for-trigger" state is a term used primarily for remote interface operation. From the front panel, the meter is always in the "wait-for-trigger" state and will accept triggers at any time, unless a measurement is already in progress.

You can place the meter in the "wait-for-trigger" state by executing any of the following commands from the remote interface.

```
MEASure?
READ?
INITiate
```

Halting a Measurement in Progress

You can send a device clear at any time to halt a measurement in progress and place the meter in the "idle state." The following statement shows how to send a device clear from a Hewlett-Packard controller if you are using the GPIB interface. *See also page 203.*

```
CLEAR 722 (device clear)
```

For operation with the RS-232, you must establish a means to send a device clear (for example, by pressing ^C (Control C)).

A device clear does not affect the configuration of the triggering system. The trigger source, sample count, trigger delay, and number of triggers are not changed.

NOTE **If you have set the meter to infinite sampling (TRIGger:COUNt INFinity), the device clear is the only way to halt the measurements.**

Number of Samples

Normally, the meter takes one reading (or sample) each time it receives a trigger from the selected trigger source (if the meter is in the wait-for-trigger state). You can, however, instruct the meter to take multiple readings for each trigger received.

- Number of samples: 1 to 50,000. *The default is 1 sample per trigger.*
- The sample count is stored in *volatile* memory; the meter sets the sample count to 1 when power has been off or after a remote interface reset. The CONFigure and MEASure? commands automatically set the sample count to 1.
- **Front Panel operation:** set the number of samples per trigger in the 3: N SAMPLES command under the TRIGger menu.
- **Remote operation:** use the following command:

  ```
  SAMPle:COUNt {<value>|MIN|MAX}
  ```

Number of Triggers

Normally, the meter will accept only one trigger before returning to the "idle" trigger state. You can, however, instruct the meter to accept multiple triggers.

This feature is available only from the remote interface. If you set the trigger count and then go to local (front panel), the meter ignores the trigger count setting; when you return to remote, the trigger count returns to the value you selected.

- The selected number of triggers is stored in *volatile* memory; the meter sets the trigger count to 1 when power has been off or after a remote interface reset or preset. The CONFigure and MEASure? commands automatically set the trigger count to 1.
- The number of triggers can be set to any number between 1 (MIN) and 50000 (MAX) or can be set to be INFinity. When the trigger count is set to be infinite, use the READ? command to collect the readings. Send a device clear to stop the measurements. *See page 203 for information about device clear.*

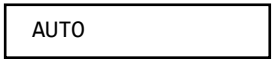
  ```
  TRIGger:COUNt {<value>|MIN|MAX|INFinity}
  ```

Trigger Delay

You can insert a delay between the trigger signal and each sample that follows. This may be useful in applications where you want to allow the input to settle before taking a reading, or for pacing a burst of readings. If you do not specify a trigger delay, the meter automatically selects a delay for you.

- Delay range: 0 to 3600 seconds. *The default trigger delay is automatic; the delay is determined by function, range, and integration time (see also "Automatic Trigger Delays" on page 102).*
- The trigger delay is stored in volatile memory; the meter selects an automatic trigger delay when power has been off or after a remote interface reset. The CONFigure and MEASure? commands set the trigger delay to automatic.
- If you specify a delay other than automatic, that same delay is used for all functions and ranges.
- If you have configured the meter to take more than one reading per trigger (sample count > 1), the specified trigger delay is inserted between the trigger and each reading.
- **Front Panel operation:** You can use an automatic trigger delay or you can specify a delay in seconds. Set the delay time in the **2: TRIG DELAY** command of the TRIGger menu.

  If an automatic trigger delay is enabled, "AUTO" is displayed momentarily before the actual number of seconds is displayed.



– **Front Panel operation** (continued)

To set the delay to 0 seconds, select the "parameter" level of the TRIG DELAY command. Move the flashing cursor to the "units" location on the right side of the display. Press [v] until ZERO DELAY is reached, then press [Enter].

– To select the automatic trigger delay, select the "parameter" level of the TRIG DELAY command. Move the flashing cursor to the "units" location on the right side of the display. Press [v] until AUTO DELAY is reached, then press [Enter].

– Remote operation:

You can use the following command to set the trigger delay.

```
TRIGger:DELay {<seconds>|MIN|MAX}
```

You can use the following command to set an automatic trigger delay.

```
TRIGger:DELay:AUTO {OFF|ON}
```

Automatic Trigger Delays

If you do not specify a trigger delay, the meter selects an automatic delay for you. The delay is determined by function, range, and integration time.

– *DC voltage:*

Range	NPLC < 1 Trigger Delay	NPLC ≥1 Trigger Delay
1 mV	15 ms	15 ms
10 mV to 120 V	1.0 ms	1.0 ms

– *Resistance (2-wire and 4-wire):*

Range	NPLC < 1 Trigger Delay	NPLC ≥1 Trigger Delay
1 Ω	1.0 ms	1.5 ms
10 Ω	1.0 ms	1.5 ms
100 Ω	1.0 ms	1.5 ms
1 kΩ	1.0 ms	1.5 ms
10 kΩ	1.0 ms	1.5 ms
100 kΩ	4 ms	6 ms
1 MΩ	40 ms	60 ms

– *Temperature:*

	NPLC < 1 Trigger Delay	NPLC ≥1 Trigger Delay
RTD	1.0 ms	1.5 ms
THERmistor	1.0 ms	1.5 ms
ThermoCouple	1.0 ms	1.5 ms

System-Related Operations

This section gives information on topics such as reading memory, errors, self-test, and front-panel display control. This information is not directly related to making measurements but is an important part of operating the meter.

Reading Memory

The meter can store up to 1024 readings in internal memory. Readings are stored in first-in-first-out (FIFO) order. The first reading returned is the first reading stored. *See also "To Use Reading Memory"* on page 62.

- You can use reading memory with all functions, math operations, and also reading hold. After you have enabled reading memory, you can change the function or input channel. *Be aware, however, that the function labels and channel labels (V, OHM, etc.) are not stored with the reading.*
- Readings taken while reading memory is enabled are stored in *volatile* memory; the meter clears the stored readings when reading memory is turned on again, when power has been off, after a self-test, or after a remote interface reset.
- You can use reading memory with autotrigger, single trigger, external trigger, and reading hold. If you configure the meter for multiple readings per trigger, the specified number of readings are stored in memory each time a trigger is received.
- Front-panel operation: enable readings storage under the **1: RDGS STORE** command in the SYStem menu. The '**Mem**' annunciator lights in the display.

  Recall the stored readings under the **2: SAVED RDGS** command in the SYStem menu. Press Shift > (Menu Recall) after enabling reading memory to quickly go to the Saved Readings command.

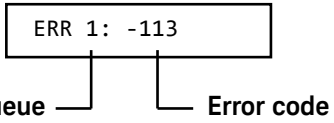
  Reading memory is automatically turned off when you go to the "parameter" level in the menu to recall the readings.
- Remote interface operation: The INITiate command uses reading memory to store readings prior to a FETCh? command. You can query the number of stored readings in memory by sending the DATA:POINts? command from the remote interface.

Error Conditions

When the front-panel '**ERROR**' annunciator turns on, one or more command syntax or hardware errors have been detected. A record of up to 20 errors is stored in the meter's error queue. *See Chapter 5, "Error Messages" for a complete listing of the errors.*

- Errors are retrieved in first-in-first-out (FIFO) order. The first error returned is the first error that was stored. When you have read all errors from the queue, the ERROR annunciator turns off. The meter beeps once for each error generated.
- If more than 20 errors have occurred, the last error stored in the queue (the most recent error) is replaced with *–350 "Queue Overflow"*. No additional errors are stored until you remove errors from the queue. If no errors have occurred and you read the error queue, the meter responds with *+0, "No error"*.
- The error queue is cleared when power has been off or after a *CLS (clear status) command has been executed.
- **Front-panel operation:** read the errors under the **3: ERROR** command in the SYStem menu.

  If the **ERROR** annunciator is on, press Shift > (Menu Recall) to read the errors stored in the queue. The errors are listed *horizontally* on the "parameter" level. The error queue is cleared when you go to the "parameter" level and then turn off the menu.



- **Remote interface operation:** use the following command to read the errors.

  ```
  SYSTem:ERRor?
  ```

  Errors have the following format (the error string may contain up to 80 characters).

  ```
  -113, "Undefined header"
  ```

Self-Test

A *power-on* self-test occurs automatically when you turn on the multimeter. This limited test assures you that the meter is operational. This self-test does not perform the extensive set of tests that are included as part of the complete self-test described below.

A *complete* self-test runs a series of tests and takes approximately 15 seconds to execute. If all tests pass, you can have a high confidence that the meter is operational.

- The results of the complete self-test are stored in internal reading memory (*see page 62*). Memory is cleared as the self-test stores this information. Other than clearing memory, the complete self-test *does not* alter the state of the meter.
- If the complete self-test is successful, "PASS" is displayed on the front panel. If the self-test fails, "FAIL" is displayed and the **ERROR** annunciator turns on. See the *34420A Service Guide* for instructions on returning the meter to Keysight for service.
- **Front Panel operation:** You can perform some of the tests individually or you can perform all tests together at once. Scroll through the parameters under the **4: TEST** command in the SYStem menu.

  Another way to perform the complete self-test from the front panel is as follows: Hold down Shift as you press the Power switch to turn on the meter; *hold down the key for more than 5 seconds*. The self-test will begin when you release the key.
- **Remote operation:** use the following command to initiate a self-test and return the results:

  ```
  *TST?
  ```

  Returns "0" if the self-test is successful, or "1" if it fails.

Display Control

To speed up your measurement rate, or for security reasons, you may want to turn off the front-panel display. From the remote interface, you can also display a 11-character message on the front panel. *See page 54 for information about the number of digits displayed.*

- When the display is turned off, readings are not sent to the display and all display annunciators except '**ERROR**' and '**Shift**' are disabled. Front-panel operation is otherwise unaffected by turning off the display.
- The display state is stored in *volatile* memory; the display is enabled when power has been off or after a remote interface reset.
- You can display a message on the front panel by sending a command from the remote interface. The meter can display up to 11 characters of the message on the front panel; any additional characters are truncated. Commas, periods, and semicolons share a display space with the preceding character, and *are not* considered individual characters. When a message is displayed, readings are not sent to the display.
- Sending a message to the display from the remote interface overrides the display state; this means that you can display a message even if the display is turned off.
- **Front-panel operation:** set the display state under the **5: DISPLAY** command in the SYStem menu.

  The display always turns on for menu operation; this means that even when the display is turned off, you can still operate the menu.
- **Remote operation:** use the following commands to control the display:

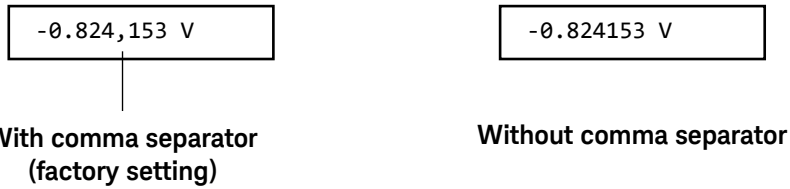
`DISPlay {OFF|ON}`	disable/enable the display
`DISPlay:TEXT <quoted string>`	display the string enclosed in quotes
`DISPlay:TEXT:CLEar`	clear the message in the display

  The following statement shows how to display a message on the front panel from a Hewlett-Packard controller.

```
OUTPUT 722; "DISP:TEXT 'HELLO'"
```

Comma Separators

The meter can display readings on the front panel with or without a comma separator. *This feature is available only from the front panel. See* *"Menu Example 1"* on page 45.

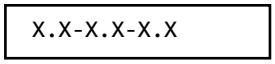


- The display format is stored in *non-volatile* memory, and *does not* change when power has been off or after a remote interface reset. The comma separator is enabled when the meter is shipped from the factory.
- **Front-panel operation:** change the state of the comma display under the **9: COMMA** command in the SYStem menu.

Firmware Revision Query

The meter has three microprocessors for control of various internal systems. You can query the meter to determine which revision of firmware is installed for each microprocessor.

- The meter returns three numbers. The first number is the firmware revision number for the measurement processor; the second is for the input/output processor; and the third is for the front-panel processor.
- Front-panel operation: read the revisions under the **11: REVISION** command in the SYStem menu:



- Remote operation: use the following command to read the revision:

  `*IDN?` returns `"KEYSIGHT TECHNOLOGIES,34420A,0,X.X-X.X-X.X"`

  Be sure to dimension a string variable with at least 40 characters.

SCPI Language Version

The meter complies with the rules and regulations of the present version of SCPI (Standard Commands for Programmable Instruments). You can determine the SCPI version with which the meter is in compliance by sending a command from the remote interface.

You cannot query the SCPI version from the front panel.

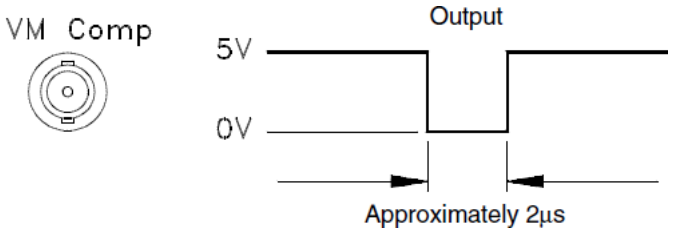
– The following command returns the SCPI version.

```
SYSTem:VERSion?
```

Returns a string in the form "YYYY.V" where the "Y's" represent the year of the version, and the "V" represents a version number for that year (for example, 1994.0).

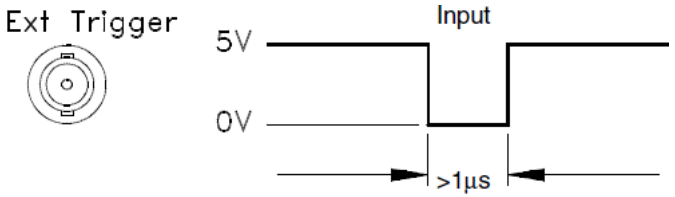
Voltmeter Complete Terminal

The rear-panel *VM Comp* (voltmeter complete) terminal provides a low-true pulse after the completion of each measurement. Voltmeter complete and external trigger (see below) implement a standard hardware handshake sequence between measurement and switching devices.



External Trigger Terminal

You can trigger the meter by applying a low-true pulse to the rear-panel *Ext Trig* (external trigger) terminal. To use this terminal from the remote interface, you must select the external trigger source (TRIGger:SOURce EXTernal).



You can use a simple switch to generate an external trigger using the *Ext Trig* input as shown below.

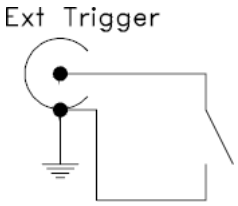


Chart Output (Analog Output)

You can use the Chart Output (Analog Output) connector on the rear panel to run a strip chart recorder or similar instrument. The connector provides an output voltage proportional to the measured voltage.

- The output can range from –3.00 V to +3.00 V.
- When chart output is disabled (Off), the output is set to 0 V.
- The output impedance is 1 kΩ.
- The output voltage is related to the input signal as follows:

$$\frac{(reading - chart\ null)}{span} = chart\ output\ voltage$$

  Note that *reading* includes the results of measurement unit conversion, filtering, null, and any math operation performed on the measurement.
- The chart output state (including the span and null values) is stored in *non-volatile* memory.
- *span* cannot have a negative value.
- The chart output has a voltage "rollover" to prevent the loss of data, *see page 113*.

Examples:

See page 48 for a menu procedure to enter a number.

1 To set the chart span so that a ±1 mV input signal produces a ±1.0 Volt chart output:

$$\frac{(0.001 - 0.0)}{span} = 1.00$$

$$span = \frac{0.001}{1.00} = 0.001$$

When chart null is set to 0, the value can be considered the *gain* and *chart output = gain * reading*.

2 To set the chart output voltage to ±3.0 V to correspond to an input temperature range of 5° C to 45° C with a mid temperature of 25° C (chart output is 0 V at 25° C).

$$\frac{(25 - \textit{chart null})}{\textit{span}} = 0 \qquad \textit{chart null} = 25$$

and

$$\frac{(45 - 25)}{\textit{span}} = 3.0 \qquad \textit{span} = \frac{20}{3.0} = 6.66$$

– **Front Panel operation:** Enable the chart output under the **5: CHART OUT** command in the SYStem menu.

Set the chart span under the **6: CHART SPAN** command in the SYStem menu.

Set the chart offset under the **7: CHART NULL** command in the SYStem menu.

You can also press Shift Null (Chart Null) to null the chart output voltage based upon the displayed reading. This action places a null value in the Chart Offset register. Pressing Shift > (Menu Recall) will take you to the **7: CHART NULL** command where you can manually edit the null value.

– **Remote operation:** use the following commands to enable and set the chart output values.

```
OUTPut {OFF|ON}
OUTPut:REFerence
     :OFFSet {<number>|MIN|MAX}
     :OFFSet:NULL
     :SPAN {<number>|MIN|MAX}
     :SPAN?
     :VALue {<number>|MIN|MAX}
     :GAIN?
```

Chart Rollover

To prevent the loss of data, the chart output will “rollover” if a measurement would cause the chart output to exceed the limits (± 3 V). This feature is especially useful with strip chart recorders.

Rollover can be used to increase the resolution of the chart since smaller values of *span* can be specified. As smaller values of *span* are used, chart rollover will occur more frequently.

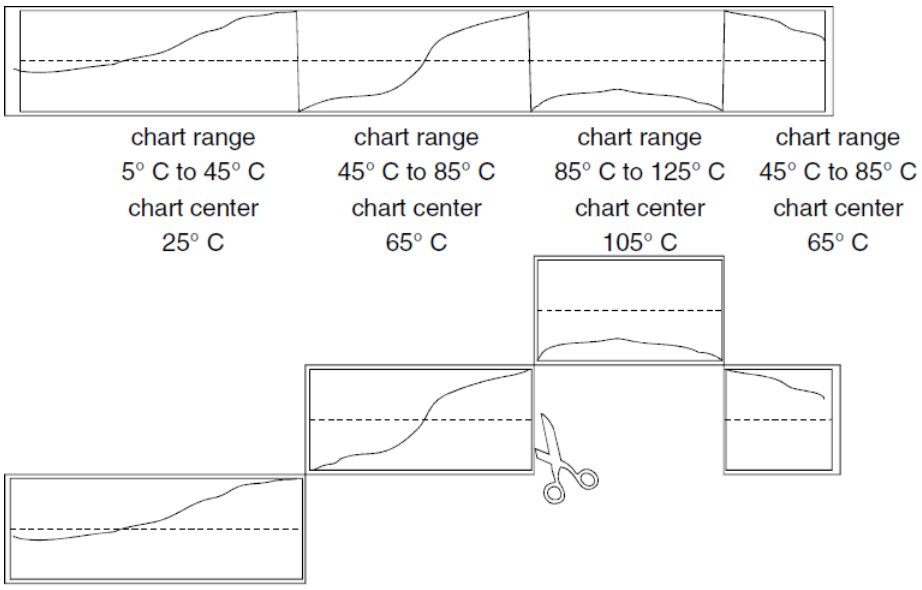
Rollover occurs when a chart limit is exceeded. The chart output voltage rolls over to the opposite limit. The result, on a strip chart, is a step from one limit to the opposite limit, effectively shifting the chart range and center values. By counting the steps and noting their direction, the actual measurement values can be deduced.

Example:

The chart output voltage is set to ± 3 V to correspond to an input temperature range of 5° C to 45° C with the 0 V output at 25° C,

span = 6.66 and *chart null* = 25.

If the actual measured temperatures ranged from 20° C to 100° C, a strip chart recorder might look like the one shown. The actual measurements recorded could be reconstructed by cutting and pasting the chart between the rollovers.



Remote Interface Configuration

This section gives information on configuring the remote interface. For programming information, see Chapter 4, "Remote Interface Reference" starting on page 129.

Remote Interface Selection

The meter is shipped with both a GPIB (IEEE-488) interface and an RS-232 interface. Only one interface can be enabled at a time. The GPIB interface is selected when the meter is shipped from the factory.

The remote interface can be set only from the front-panel.

- The interface selection is stored in *non-volatile* memory, and *does not* change when power has been off or after a remote interface reset.
- If you select the GPIB interface, you must select a unique address for the meter. The GPIB address is displayed when you turn on the meter.
- If you select the RS-232 interface, you must set the baud rate and parity for the meter. “RS-232” is displayed when you turn on the meter.
- If you select the RS-232 interface and then set the GPIB address to the talk only address (31), the meter will *send* readings over the RS-232 interface when in the local mode.
- There are certain restrictions to be aware of when you are selecting the remote interface (*see also “Programming Language Selection”* on page 119). The only programming language supported on RS-232 is SCPI.
- **Front Panel operation:** set the interface under the **2: INTERFACE** command in the I/O menu.

  See also “To Select the Remote Interface” on page 207.

GPIB Address

Each device on the GPIB (IEEE-488) interface must have a unique address. You can set the meter's address to any value between 0 and 31. The address is set to "22" when the meter is shipped from the factory. The GPIB address is displayed when you turn on the meter.

The GPIB address can be set only from the front-panel.

- The address is stored in *non-volatile* memory, and *does not* change when power has been off or after a remote interface reset.
- You can set the address to "31" which is the *talk only* mode. In this mode, the meter can output readings directly to a printer without being addressed by a bus controller. Address 31 is *not* a valid address if you are operating the multimeter from the GPIB interface with a bus controller.
- If you select the RS-232 interface and then set the GPIB address to the talk only address (31), the meter will *send* readings over the RS-232 interface when in the local mode.
- Your GPIB bus controller has its own address. Be sure to avoid using the bus controller's address for any instrument on the interface bus. Hewlett-Packard controllers generally use address "21".
- **Front Panel operation:** set the address under the **1: GPIB ADDR** command in the I/O menu.

  See also "To Set the GPIB Address" on page 205.

Baud Rate Selection (RS-232)

You can select one of six baud rates for RS-232 operation. The rate is set to 9600 baud when the meter is shipped from the factory.

- Select one of the following: 300, 600, 1200, 2400, 4800, or **9600** baud (factory setting).
- The baud rate selection is stored in *non-volatile* memory, and *does not* change when power has been off or after a remote interface reset.
- **Front Panel operation:** set the baud rate under the **3: BAUD RATE** command in the I/O menu.

  See also "To Set the Baud Rate" on page 209.

- **Remote operation:** use the following commands to set the baud rate:

  `SYStem:COMMunicate:SERial:BAUD <baud>`

If you change the baud rate over the remote interface, you may lose control of the meter.

Parity Selection (RS-232)

You can select the parity for RS-232 operation. The meter is configured for even parity with 7 data bits when shipped from the factory.

- Select one of the following: None (8 data bits), **Even** (7 data bits), or Odd (7 data bits). When you set the parity, you are indirectly setting the number of data bits.
- The parity selection is stored in *non-volatile* memory, and *does not* change when power has been off or after a remote interface reset.
- **Front Panel operation:** set the parity under the **4: PARITY** command in the I/O menu.

  See also "To Set the Parity" on page 211.
- **Remote operation:** use the following commands to set the parity:

  ```
  SYStem:COMMunicate:SERial:PARity {EVEN|ODD|NONE}
  ```

If you change the parity over the remote interface, you may lose control of the meter.

Programming Language Selection

You can select one of two languages to program the meter from the selected remote interface. The programming language is SCPI when the meter is shipped from the factory.

- Select one of the following: **SCPI** or 181 (Keithley).
- The language selection is stored in *non-volatile* memory, and *does not* change when power has been off or after a remote interface reset.
- There are certain restrictions to be aware of when you are selecting the interface language (*see also* *"Remote Interface Selection"* on page 115). The Keithley 181 language *is not* supported on the RS-232 interface.
- **Front Panel operation:** set the programming language under the **5: LANGUAGE** command in the I/O menu.

  See also "To Select the Programming Language" on page 213.
- **Remote operation:** use the following command to set the language:

  `SYStem:LANGuage <Language>`

Connection to a Terminal or Printer (RS-232)

The RS-232 connector on the meter's rear panel is a 9-pin connector (DB-9, male connector). You can connect the meter to any terminal or printer with a properly configured DTE connector (DB-25). You can use a standard serial interface cable and the *34399A Adapter Kit* to make connections. *Additional information about the interface is given on* *page 191*.

Pin Number	Input/Output	Description
1		
2	Input	Receive Data (RxD)
3	Output	Transmit Data (RxD)
4	Output	Data Terminal Ready(DTR)
5	-	Signal Ground (SG)
6	Input	Data Set Ready (DSR)
9		

If you are using the RS-232 interface for connection to a printer, you may wish to use the Talk Only mode. This mode is enabled by setting the GPIB address to 31. *See* *page 204* *for more information.*

Calibration

This section gives a brief introduction to the calibration features of the meter. For a more detailed discussion of the calibration procedures, see Chapter 4 in the *Service Guide*.

Calibration Security

This feature allows you to enter a security code to prevent accidental or unauthorized calibrations of the meter. When you first receive your meter, it is secured. Before you can calibrate the meter, you must unsecure it by entering the correct security code.

- *The security code is set to "KT034420" when the meter is shipped from the factory*. The security code is stored in *non-volatile* memory, and *does not* change when power has been off or after a remote interface reset.
- To secure the meter from the remote interface, the security code may contain up to 11 alphanumeric characters as shown below. The first character *must* be a letter, but the remaining characters can be letters or numbers. You do not have to use all 11 characters but the first character must always be a letter.

  A _ _ _ _ _ _ _ _ _ _ _ *(11 characters)*

- To secure the meter from the remote interface so that it can be unsecured from the front panel, use the eight-character format shown below. The first two characters must be "KT" and the remaining characters must be numbers. Only the last six characters are recognized from the front panel, but all eight characters are required. (To unsecure the meter from the front panel, omit the "KT" and enter the remaining numbers as shown on the following pages.)

  K T _ _ _ _ _ _ *(8 characters)*

NOTE If you secure the meter from the remote interface and the secure code you use does not have the first two characters "KT" and the next 6 characters as digits, you cannot unsecure the meter from the front panel.

To Unsecure for Calibration You can unsecure the meter for calibration either from the front panel or remote interface. *The meter is secured when shipped from the factory, and the security code is set to "KT034420".*

– **Front-panel operation:**

```
1: SECURED
```

If the meter is secured, you will see the above command when you go into the CAL MENU. (If you move across the "commands" level in the menu, you will notice that the "2: CALIBRATE" command is "hidden" if the meter is secured.) To unsecure the meter, select the "parameter" level of the SECURED command, enter the security code, then press Enter.

```
^ 000000 CODE
```

When you go to the "commands" level in the CAL MENU again, you will notice that the meter is unsecured. Notice also that the **2: CALIBRATE** command is no longer hidden and you can perform a calibration.

```
1: UNSECURED
```

– **Remote operation:** use the following command:

`CALibration:SECure:STATe {OFF|ON},<code>`

To unsecure the meter, send the above command with the same code used to secure. For example,

```
CAL:SEC:STAT OFF, KT034420
```

To Secure Against Calibration You can secure the meter against calibration either from the front panel or remote interface. *The meter is secured when shipped from the factory, and the security code is set to "KT034420".*

Be sure to read the security code rules on page 121 before attempting to secure the meter.

– **Front-panel operation:**

```
1: UNSECURED
```

If the meter is unsecured, you will see the above command when you go into the CAL MENU. To secure the meter, select the "parameter" level of the UNSECURED command, enter the security code, then press Menu Enter.

```
^ 000000 CODE
```

When you go to the "commands" level in the CAL MENU again, you will notice that the meter is secured. Notice also that the **2: CALIBRATE** command is now hidden and you cannot perform a calibration.

```
1: SECURED
```

– **Remote operation:** use the following command:

```
CALibration:SECure:STATe {OFF|ON},<code>
```

To secure the meter, send the above command with the same code as used to unsecure. For example,

```
CAL:SEC:STAT ON, KT034420
```

To Change the Security Code To change the security code, you must first unsecure the meter, and then enter a new code. Make sure you have read the security code rules on page 121 before attempting to secure the meter.

- **Front-Panel operation:** To change the security code, first make sure that the meter is unsecured. Select the "parameter" level of the UNSECURED command, enter the new security code, then press Menu Enter. Changing the code from the front panel also changes the code as seen from the remote interface.
- **Remote interface operation:**

  `CALibration:SECure:CODE <new code>`

  To change the security code, first unsecure the meter using the old security code. Then, enter the new code. For example,

  `CAL:SEC:STAT OFF, KT034420` *unsecure with old code*

  `CAL:SEC:CODE KT010495` *enter new code*

Calibration Count

You can determine the number of times that your meter has been calibrated. Since the value increments by one for each calibration point, a complete calibration increases the value by several counts.

- The calibration count is stored in *non-volatile* memory, and *does not* change when power has been off or after a remote interface reset. Your meter was calibrated before it left the factory. When you receive your meter, read the count to determine its initial value.
- The calibration count increments up to a maximum of 32,767 after which it wraps-around to 0.
- **Front-Panel operation:** read the calibration count under the **3: CAL COUNT** command in the CALibrate menu.
- **Remote operation:** use the following command:

  `CALibration:COUNt?`

Calibration Message

You can use the calibration message feature to record calibration information about your meter. For example, you can store such information as the last calibration date, the next calibration due date, the meter's serial number, or even the name and phone number of the person to contact for a new calibration.

You can record information in the calibration message only from the remote interface. You can read the message from either the front-panel menu or the remote interface.

- The calibration message may contain up to 40 characters. The meter can display up to 11 characters of the message on the front panel; any additional characters are truncated.
- The calibration message is stored in *non-volatile* memory, and *does not* change when power has been off or after a remote interface reset.
- **Front Panel operation:** read the calibration message (if any) under the 4: MESSAGE command in the CALibrate menu.
- **Remote operation:** use the following command to store a calibration message:

  ```
  CALibration:STRing <quoted string>
  ```

  The following statement shows how to store a calibration message from a Hewlett-Packard controller.

  ```
  OUTPUT 722; "CAL:STR 'CAL 9-1-94'"
  ```

Defaults, Power-On and Reset States

The meter stores settings in either *volatile* or *non-volatile* memory. Settings stored in *volatile* memory are returned to default settings at power-on or after a remote reset. Settings stored in *non-volatile* memory are not changed by power-on or a remote reset (*RST), CONFigure, or MEASure command.

– *Volatile* memory settings and their defaults:

Setting	Front Panel Default	Remote Default[a]
Function	DCV	DCV
Input Channel	Channel 1	Channel 1
Digital Filter	**On (50 rdgs)**	**Off**[c]
Analog Filter	Off	Off
Offset Compensation	**On**	**Off**
N Samples	1	1
Display (On or Off)	On	On
Trigger	Autotrigger	Autotrigger
Math	Off	Off
Readings Hold	Off	Off
Readings Store	Off	Off
Trigger Delay	(10 plc)	(10 plc)
Integration Time (NPLC)[b]	Off	Off
Null[b]	Autorange	Autorange
Range[b]	Auto	Auto

[a] Remote defaults are used following reset (*RST), preset (SYSTem:PRESet), CONFigure, or MEASure commands.

[b] Integration Time, Null, and Range are independent in each function. CONFigure or MEASure commands will only affect the Integration Time, Null, or Range for the function used.

[c] You should ensure the digital filter is turned off when using the remote interface. *See page 70*.

– *Non-volatile* memory settings can be returned to their factory defaults.

Front Panel operation: return the settings to factory defaults under the 11: PRESET command in the SYStem menu.

Remote operation: use the SYStem:PRESet command to return the meter to the factory defaults.

Non-volatile memory settings and their factory defaults:

Setting	Factory Default
Digital Filter[a]	MEDium 50 rdgs
Digital Filter Precharge	On
Low Power Ω	Off
Low Volt Ω	Off
LoV Limit Ω	20 mV
Probe Type	RTD
RTD Type	α = .00385
RTD R_0	100 Ω
Temperature Units	°C
Chart Out	Off
Chart Span	1 mV
Chart Offset	0.00 V
Comma	On

[a] The digital filter is ON following a Front Panel PRESET.
The digital filter is OFF following a Remote Interface PRESET.

– *Independent vs. common settings:* Some settings are independent to the channel or function being used and some are common to multiple channels or functions:

<table>
<tr><th colspan="2">Voltage</th><th colspan="2">Resistance</th><th rowspan="2">Temperature</th></tr>
<tr><th>Channel 1</th><th>Channel 2</th><th>2-Wire Ω</th><th>4-Wire Ω</th></tr>
<tr><td>Null (On/Off)
Range</td><td>Null (On/Off)
Range</td><td colspan="2" rowspan="2">Null (On/Off)
Range[a]
Integration Time
Number of Digits</td><td rowspan="2">Null (On/Off)
Range
Integration Time
Number of Digits</td></tr>
<tr><td colspan="2">Integration Time
Number of Digits</td></tr>
<tr><td colspan="5">Math (Stats or Scale)
Digital Filter (On /Off)
Analog Filter (On/Off)[b]
Trigger
Readings Hold
Readings Store</td></tr>
</table>

[a] The meter may change the set resistance measurement range when voltage limited measurements are enabled.

[b] Analog filter available for voltage and thermocouple measurements on the 1 mV, 10 mV, and 100 mV ranges.

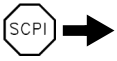
4 Remote Interface Reference

Command Summary 133
Simplified Programming Sequence 142
MEASure? and CONFigure Defaults 143
Setting the Function, Range, and Resolution 152
Selecting the Input Channel 155
Special Resistance Measurement Commands 156
Temperature Measurement Commands 157
Null (Relative) Commands 160
Input Filter Commands 162
Math Operation Commands 164
Triggering 167
Triggering Commands 170
Chart Output (Analog Out) Commands 172
System-Related Commands 173
The SCPI Status Model 175
Status Reporting Commands 186
Calibration Commands 189
RS-232 Interface Configuration 191
RS-232 Interface Commands 197
An Introduction to the SCPI Language 198
Input Message Terminators 201
Output Data Formats 202
Using Device Clear to Halt Measurements 203
TALK ONLY for Printers 204
To Set the GPIB Address 205
To Select the Remote Interface 207

To Set the Baud Rate 209
To Set the Parity 211
To Select the Programming Language 213
To Select the Programming Language 213
Alternate Programming Language Compatibility 215
SCPI Compliance Information 216
IEEE-488 Compliance Information 219

Remote Interface Reference

This chapter is divided into the following sections:

- "Command Summary" on page 133
- 

"Simplified Programming Sequence" on page 142
- "MEASure? and CONFigure Defaults" on page 143
- "Setting the Function, Range, and Resolution" on page 152
- "Selecting the Input Channel" on page 155
- "Special Resistance Measurement Commands" on page 156
- "Temperature Measurement Commands" on page 157
- "Null (Relative) Commands" on page 160
- "Input Filter Commands" on page 162
- "Math Operation Commands" on page 164
- "Triggering" on page 167
- "Triggering Commands" on page 170
- "Chart Output (Analog Out) Commands" on page 172
- "System-Related Commands" on page 173
- "The SCPI Status Model" on page 175
- "Status Reporting Commands" on page 186
- "Calibration Commands" on page 189
- "RS-232 Interface Configuration" on page 191
- "RS-232 Interface Commands" on page 197
-

"An Introduction to the SCPI Language" on page 198
- "Input Message Terminators" on page 201
- "Output Data Formats" on page 202
- "Using Device Clear to Halt Measurements" on page 203
- "TALK ONLY for Printers" on page 204
- "To Set the GPIB Address" on page 205
- "To Select the Remote Interface" on page 207

- "To Set the Baud Rate" on page 209
- "To Set the Parity" on page 211
- "To Select the Programming Language" on page 213
- "To Select the Programming Language" on page 213
- "Alternate Programming Language Compatibility" on page 215
- "SCPI Compliance Information" on page 216
- "IEEE-488 Compliance Information" on page 219

If you are a first-time user of the SCPI language, refer to these sections to become familiar with the language before attempting to program the meter.

Command Summary

This section summarizes the SCPI (Standard Commands for Programmable Instruments) commands available to program the meter. Refer to the later sections in this chapter for more complete details on each command.

NOTE

Throughout this manual, the following conventions are used for SCPI command syntax. Square brackets ([]) indicate optional keywords or parameters. Braces ({}) enclose a list of parameters separated with a vertical bar (|), use only one of the values. Triangle brackets (<>) indicate that you must substitute a value for the enclosed parameter.

The actual command string you send to the meter should not include any of the conventions indicated above.

First-time SCPI users, see page 198.

Voltage Measurement Configuration Commands

```
MEASure
   [:VOLTage][:DC]?
      [{<range> | AUTO | MIN | MAX | DEF}][,{<resolution> | MIN | MAX | DEF}][,(@<channel>)][1]
   [:VOLTage][:DC]:RATio?
      [{<range> | AUTO | MIN | MAX | DEF}][,{<resolution> | MIN | MAX | DEF}]
   [:VOLTage][:DC]:DIFFerence?
      [{<range> | AUTO | MIN | MAX | DEF}][,{<resolution> | MIN | MAX | DEF}]

CONFigure
   [:VOLTage][:DC]
      [{<range> | AUTO | MIN | MAX | DEF}][,{<resolution> | MIN | MAX | DEF}][,(@<channel>)]1
   [:VOLTage][:DC]:RATio [{<range> | AUTO | MIN | MAX | DEF}] [,{<resolution> | MIN | MAX | DEF}]
   [:VOLTage][:DC]:DIFFerence[{<range> | AUTO | MIN | MAX | DEF}]
      [,{<resolution> | MIN | MAX | DEF}]

[SENSe:] | SENSe1: | SENSe2:
   FUNCtion "VOLTage[:DC]"
   FUNCtion "VOLTage[:DC]:RATio"
   FUNCtion "VOLTage[:DC]:DIFFerence"
   FUNCtion?

[SENSe:] SENSe1: | SENSe2:
   VOLTage[:DC]:RANGe {<range> | MIN | MAX}
   VOLTage[:DC]:RANGe? [MIN | MAX]
   VOLTage[:DC]:RANGe:AUTO {OFF | ON}
   VOLTage[:DC]:RANGe:AUTO?

[SENSe:] | SENSe1: | SENSe2:
   VOLTage[:DC]:RESolution {<resolution> | MIN | MAX}
   VOLTage[:DC]:RESolution? [MIN | MAX]

[SENSe:] | SENSe1: | SENSe2:
   VOLTage[:DC]:NPLCycles {0.02 | 0.2 | 1 | 2 | 10 | 20 | 100 | 200 | MIN | MAX}
   VOLTage[:DC]:NPLCycles? [MIN | MAX]

[SENSe:] | SENSe1: | SENSe2:
   VOLTage[:DC]:NULL[:STATe] {OFF | ON}
   VOLTage[:DC]:NULL[:STATe]?
   VOLTage[:DC]:NULL:VALue {<value> | MIN | MAX}
   VOLTage[:DC]:NULL:VALue?

ROUTe
   :TERMinals {FRONt | FRONt1 | FRONt2}
   :TERMinals?
```

[1] *<channel>* is one of: {FRONt | FRONt1 | FRONt2 | 1 | 2}

Resistance Measurement Configuration Commands

```
MEASure
   :FRESistance? [{<range> | AUTO | MIN | MAX | DEF}][,{<resolution> | MIN | MAX | DEF}]
   :RESistance? [{<range> | AUTO | MIN | MAX | DEF}][,{<resolution> | MIN | MAX | DEF}]

CONFigure
   :FRESistance | :RESistance
      [{<range> | AUTO | MIN | MAX | DEF}][,{<resolution> | MIN | MAX | DEF}]

[SENSe:]
   FUNCtion "RESistance" (2-wire ohms)
   FUNCtion "FRESistance" (4-wire ohms)
   FUNCtion?

[SENSe:]
   FRESistance | RESistance
      :RANGe {<range> | MIN | MAX}
      :RANGe? [MIN | MAX]
      :RANGe:AUTO {OFF | ON}
      :RANGe:AUTO?

[SENSe:]
   FRESistance | RESistance
      :RESolution {<resolution> | MIN | MAX}
      :RESolution? [MIN | MAX]

[SENSe:]
   FRESistance | RESistance
      :NPLCycles {0.02 | 0.2 | 1 | 2 | 10 | 20 | 100 | 200 | MIN | MAX}
      :NPLCycles? [MIN | MAX]

[SENSe:]
   FRESistance | RESistance
      :NULL[:STATe] {OFF | ON}
      :NULL[:STATe]?
      :NULL:VALue {<value> | MIN | MAX}
      :NULL:VALue?

[SENSe:]
   FRESistance | RESistance
      :OCOMpensated {OFF | ON}
      :OCOMpensated?
```

```
[SENSe:]
   FRESistance
      :POWer:LIMit[:STATe] {OFF | ON}
      :POWer:LIMit?
      :VOLTage:LIMit[:STATe] {OFF | ON}
      :VOLTage:LIMit?
      :VOLTage:LIMit:VALue {<value> | MIN | MAX}
      :VOLTage:LIMit:VALue? [MIN | MAX]
```

Temperature Measurement Configuration Commands

```
MEASure
   :TEMPerature? [{TC | THER | FRTD | DEF}][,{<type> | DEF}][, 1, {<resolution> | MAX | MIN | DEF}]
```

```
CONFigure
   :TEMPerature [{TC | THER | FRTD | DEF}][,{<type> | DEF}][, 1, {<resolution> | MAX | MIN | DEF}]
```

```
[SENSe:]
   FUNCtion "TEMPerature"
   FUNCtion?
```

```
[SENSe:]
   TEMPerature:TRANsducer:TYPE {TCouple | THERmistor | FRTD}
   TEMPerature:TRANsducer:TYPE?
```

```
[SENSe:]
   TEMPerature:TRANsducer:TCouple:TYPE {B | E | J | K | N | R | S | T}
   TEMPerature:TRANsducer:TCouple:TYPE?
   TEMPerature:TRANsducer:TCouple:RJUNction {<reference> | MIN | MAX}
   TEMPerature:TRANsducer:TCouple:RJUNction? [{MIN | MAX}]
   TEMPerature:TRANsducer:TCouple:RJUNction:TYPE {FIX | INT | THER}
   TEMPerature:TRANsducer:TCouple:RJUNction:TYPE?
```

```
[SENSe:]
   TEMPerature:TRANsducer:FRTD:TYPE {85 | 91}
   TEMPerature:TRANsducer:FRTD:TYPE?
   TEMPerature:TRANsducer:FRTD:RESistance[:REFerence] <value>
   TEMPerature:TRANsducer:FRTD:RESistance[:REFerence]?
```

```
[SENSe:]
   TEMPerature:NULL[:STATe] {OFF | ON}
   TEMPerature:NULL[:STATe]?
   TEMPerature:NULL:VALue {<value> | MIN | MAX}
   TEMPerature:NULL:VALue?
```

```
[SENSe:]
   TEMPerature:NPLCycles {0.02 | 0.2 | 1 | 2 | 10 | 20 | 100 | 200 | MIN | MAX}
   TEMPerature:NPLCycles? {MIN | MAX}
```

```
UNIT
   :TEMPerature {C | CEL | F | FAR | K}
   :TEMPerature?
```

General Measurement Configuration Commands

```
CONFigure?
```

```
[SENSe:]NULL [{OFF | ON | ONCE}]
```

```
INPut:FILTer
   :STATe {OFF | ON}
   :STATe?
   :TYPE {ANALog | DIGital | BOTH}
   :TYPE?
   :DIGital:RESPonse {SLOW | MEDium | FAST}
   :DIGital:RESPonse?
   :DIGital:PRECharge:AUTO {ON | OFF}
   :DIGital:PRECharge:AUTO?
```

```
ROUTe:TERMinals {FRONt | FRONt1 | FRONt2}
```

```
ROUTe:TERMinals?
```

Math Operation Commands

```
CALCulate
   :FUNCtion {AVERage | SCALe}
   :FUNCtion?
   [:STATe] {OFF | ON}
   [:STATe]?
```

```
CALCulate
   :AVERage:MINimum?
   :AVERage:MAXimum?
   :AVERage:AVERage?
   :AVERage:COUNt?
   :AVERage:PTPeak?
   :AVERage:SDEViation?
```

```
CALCulate
   :SCALe:GAIN {<gain>}
   :SCALe:GAIN?
   :SCALe:OFFSet {<offset>}
   :SCALe:OFFSet?
```

```
DATA
   :FEED RDG_STORE,[{"CALC" | ""}]
   :FEED? RDG_STORE
   :POINts?
```

Triggering Commands

```
INITiate

READ?

TRIGger
   :SOURce {BUS | IMMediate | EXTernal}
   :SOURce?

TRIGger
   :DELay {<seconds> | MIN | MAX}
   :DELay? [MIN | MAX]
   :DELay:AUTO {OFF | ON}
   :DELay:AUTO?

TRIGger
   :COUNt {<value> | MIN | MAX | INFinity}
   :COUNt? [MIN | MAX]

SAMPle
   :COUNt {<value> | MIN | MAX}
   :COUNt? [MIN | MAX]
```

Chart Output Commands

```
OUTPut
   [:STATe] {OFF | ON}
   [:STATe]?
   :REFerence:OFFSet {<number> | MIN | MAX}
   :REFerence:OFFSet? [{MIN | MAX}]
   :REFerence:OFFSet:NULL
   :REFerence:SPAN [{<number> | MIN | MAX}]
   :REFerence:SPAN? [{MIN | MAX}]
   :REFerence:GAIN?
   :REFerence:VALue {<value> | MIN | MAX}
```

System-Related Commands

```
FETCh?

READ?

DATA
   :FEED RDG_STORE,[{"CALC" | ""}]
   :FEED? RDG_STORE
   :POINts?
```

```
DISPlay[:STATe] {OFF | ON}
DISPlay?
DISPlay
   :TEXT <quoted string>
   :TEXT?
   :TEXT:CLEar

SYSTem:ERRor?

SYSTem:PRESet

SYSTem:VERSion?

*RST

*TST?

*IDN?
```

Status Reporting Commands

```
SYSTem:ERRor?

STATus
   :OPERation:CONDition?
   :OPERation:ENABle <enable value>
   :OPERation:ENABle?
   :OPERation[:EVENt]?
   :QUEStionable:CONDition?
   :QUEStionable:ENABle <enable value>
   :QUEStionable:ENABle?
   :QUEStionable[:EVENt]?

STATus:PRESet

*CLS                      *STB?

*ESE <enable value>       *PSC {0 | 1}
*ESE?                     *PSC?

*ESR?                     *SRE <enable value>
                          *SRE?

*OPC
*OPC?
```

Calibration Commands

```
CALibration?

CALibration:COUNt?

CALibration
   :ICURrent?

CALibration
   :OUTPut {ZERO | GAIN}
   :OUTPut?

CALibration
   :SECure:CODE <new code>
   :SECure:STATe {OFF | ON} [,<code>]
   :SECure:STATe?

CALibration
   :STRing <quoted string>
   :STRing?

CALibration
   :VALue <value>
   :VALue?
```

Remote Interface Commands

```
SYSTem
   :INTerface {GPIB | RS232}
   :INTerface?

SYSTem
   :COMMunicate:SERial:BAUD <baud>
   :COMMunicate:SERial:BAUD?
   :COMMunicate:SERial:PARity {EVEN | ODD | NONE}
   :COMMunicate:SERial:PARity?

SYSTem
   :LANGuage {SCPI | 181}
   :LOCal
   :REMote
   :RWLock

L1X (SCPI language)
L2X (181 language)
```

IEEE-488.2 Common Commands

*CLS

*ESE *<enable value>*
*ESE?

*ESR?

*IDN?

*OPC

*OPC?

*PSC {0 | 1}
*PSC?

*RST

*SRE *<enable value>*
*SRE?

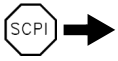
*STB?

*TRG

*TST?

*WAI

Simplified Programming Sequence



You can program the meter to take measurements from the remote interface using the following seven-step sequence.

NOTE **Throughout this manual, the following conventions are used for SCPI command syntax. Square brackets ([]) indicate optional keywords or parameters. Braces ({}) enclose a list of parameters separated with a vertical bar (|), use only one of the values . Triangle brackets (<>) indicate that you must substitute a value for the enclosed parameter.**

1. Place the meter in a known state (often the reset state).
2. Change the meter's settings to achieve the desired configuration.
3. Set-up the triggering conditions.
4. Initiate or arm the meter for a measurement.
5. Trigger the meter to make a measurement.
6. Retrieve the readings from the output buffer or internal memory.
7. Read the measured data into your bus controller.

The **MEASure?** and **CONFigure** commands provide the most straightforward method to program the meter for measurements. You can select the measurement function, range, and resolution all in one command. The meter uses defaults for other measurement parameters.

Integration time, range, and resolution are local to the function or channel used in the **MEASure?** and **CONFigure** commands. For example, if you use the MEASure command to make a resistance measurement, the range, resolution, and integration time set for voltage measurements are not affected.

Other measurement parameters affect all measurement functions. When you use the **MEASure?** and **CONFigure** commands the meter uses default parameters. The default measurement parameters, common to all functions or channels, used by the meter are shown in the table on the next page.

NOTE **Using the digital filter is not recommended from the remote interface, see page 70.**

MEASure? and CONFigure Defaults

	Setting	SCPI Command System	Defaults Used
Common to all Functions	Input Channel	ROUTe:TERMinals	Channel 1
	Digital Filter	INPut:FILTer	Off
	Digital Filter Precharge	INPut:FILTer	Last value set
	Analog Filter	INPut:FILTer	Off
	Trigger Source	TRIGger:SOURce	Immediate
	Trigger Delay	TRIGger:DELay	Auto
	Trigger Count	TRIGger:COUNt	1
	N Samples	SAMPle:COUNt	1
	Math	CALCulate	Off
	Readings Store	DATA:FEED RDG_STORE	"CALC"
	Display (On or Off)	DISPlay	On
Specific To Each Function	Integration Time (NPLC)	<*resolution*>	(10 plc)
	Range	<*range*>	Autorange
	Null	SENSe: <*function*> :NULL	Off
Resistance Specific	Offset Compensated Ω	SOURce:FRESistance:OCOMPensated	Off
	Voltage Limited Ω	SOURce:FRESistance:VOLTage:LIMit	Last value set
	Low Power Ω	SOURce:FRESistance:POWer:LIMit	Last value set
Temperature Specific	Transducer	SENSe:TEMPerature:TRANsducer	Last probe set
	Probe Type	SENSe:TEMPerature:TRANsducer	Last type set
	RTD type and resistance	SENSe:TEMPerature:TRANsducer	Last type set

NOTE The MEASure? and CONFigure commands use default parameters for volatile memory parameters. Non-volatile memory parameters are not modified and the MEASure? or CONFigure commands use parameters in memory. *See page 126*. For example, the temperature units last set are stored in non-volatile memory and are not changed by a MEASure? or CONFigure command.

Using the MEASure? Command

The easiest way to program the meter for measurements is by using the **MEASure?** command. However, this command does not offer much flexibility. When you execute the command, the meter uses defaults for the requested configuration and immediately performs the measurement. You cannot change any settings (other than function, range, and resolution) before the measurement is taken. The results are sent to the output buffer.

Sending the MEASure? command is functionally the same as sending a CONFigure command followed immediately by a READ? command.

Using the CONFigure Command

For a little more programming flexibility, use the **CONFigure** command. When you execute the command, the meter sets the defaults for the requested configuration (like the **MEASure?** command). However, the measurement *is not* automatically started and you can change measurement parameters before making measurements. This allows you to "incrementally" change the meter's configuration from the default conditions. The meter offers a variety of low-level commands in the **INPut**, **SENSe**, **CALCulate**, and **TRIGger** subsystems. (You can use the SENSe:FUNCtion command to change the measurement function without using **MEASure?** or **CONFigure**.)

Use the INITiate or READ? command to initiate the measurement.

Using the *range* and *resolution* Parameters

With the **MEASure?** and **CONFigure** commands, you can select the measurement function, range, and resolution all in one command. Use the *range* parameter to specify a fixed range larger than the expected value of the input signal. You can set the range parameter to AUTO to set autoranging.

Use the *resolution* parameter to specify the desired resolution for the measurement. Setting the resolution sets the integration time of the meter. The resolution parameter can be set to any number but the resulting integration time (NPLC) is one of the following:

<resolution> Parameter	Integration Time (NPLC)
< 0.0001 x Range	0.02
< 0.00001 x Range	.2
< 0.000003 x Range	1
< 0.0000022 x Range	2
< 0.000001 x Range	10
< 0.0000008 x Range	20
< 0.0000003 x Range	100
< 0.00000022 x Range	200

Specify the resolution in the same units as the measurement function, *not in number of digits*. For example, for volts measurements, specify the resolution in volts. For resistance, specify the resolution in ohms.

You must specify a range to use the resolution parameter.

Using the READ? Command

The **READ?** command changes the state of the trigger system from the "idle" state to the "wait-for-trigger" state. Measurements will begin when the specified trigger conditions are satisfied following the receipt of the **READ?** command. Readings are sent *immediately* to the output buffer. You *must* enter the reading data into your bus controller or the meter will stop making measurements when the output buffer fills. Readings *are not* stored in the meter's internal memory when using the READ? command.

Sending the READ? command is like sending the INITiate command followed immediately by the FETCh? command, except readings are not buffered internally.

CAUTION 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.

Using the INITiate and FETCh? Commands

The **INITiate** and **FETCh?** commands provide the lowest level of control (with the most flexibility) of measurement triggering and reading retrieval. Use the INITiate command after you have configured the meter for the measurement. This changes the state of the triggering system from the "idle" state to the "wait-for-trigger" state. Measurements will begin when the specified trigger conditions are satisfied after the **INITiate** command is received. The readings *are* placed in the meter's internal memory (up to 1024 readings can be stored). Readings *are stored* in memory until you are able to retrieve them.

Use the **FETCh?** command to transfer the readings from the meter's internal memory to the meter's output buffer where you can read them into your bus controller.

Configure the meter using the SENSe commands. For voltage measurements on two channels, you can configure each channel independently by specifying either SENSe1 or SENSe2. Once configured, use the ROUTe:TERM command to switch between the input channels.

NOTE The INIT and FETCh? commands use the meter's memory. You can defeat the storing of readings, and still obtain the statistics on the readings using the DATA:FEED command. *See page 166*.

MEASure? Example

The following program segment shows how to use the **MEASure?** command to make a measurement. This example configures the meter for voltage measurements on channel 1 using autoranging on the input signal, automatically places the meter in the "wait-for-trigger" state, internally triggers the meter to take one reading, and then sends the reading to the output buffer.

```
MEAS:VOLT:DC? AUTO,MIN,(@FRONt1)
bus enter statement
```

This is the simplest way to take a reading. However, you do not have any flexibility with **MEASure?** to set the trigger count, sample count, trigger delay, etc. All measurement parameters except function, range, and resolution are preset for you automatically *(see the table on page 143)*.

CONFigure Example 1

The following program segment shows how to use the **READ?** command with **CONFigure** to make an externally-triggered measurement. The program configures the meter for dc voltage measurements. **CONFigure** does not place the meter in the "wait-for-trigger" state. The **READ?** command places the meter in the "wait-for-trigger" state, takes a reading when the *Ext Trig* terminal is pulsed, and sends the reading to the output buffer.

```
CONF:VOLT:DC 10, MIN, (@FRONt1)
TRIG:SOUR EXT
READ?
bus enter statement
```

CONFigure Example 2

The following program segment is similar to the example above but it uses INITiate to place the meter in the "wait-for-trigger" state. The INITiate command places the meter in the "wait-for-trigger" state, takes a reading when the *Ext Trig* terminal is pulsed, and sends the reading to the meter's internal memory. The FETCh? command transfers the reading from internal memory to the output buffer.

```
CONF:VOLT:DC 10, MIN, (@FRONt1)
TRIG:SOUR EXT
INIT
FETC?
bus enter statement
```

Storing readings in memory using the INITiate command is faster than sending readings to the output buffer using the READ? command. The meter can store up to 1024 readings in internal memory. If you configure the meter to take more than 1024 readings (using the sample count and trigger count), and then send INITiate, a memory error is generated.

After you execute an INITiate command, no further commands are accepted until the measurement sequence is completed. However, if you select TRIGger:SOURce BUS, the meter will accept the *TRG command (bus trigger) or an IEEE-488 *Group Execute Trigger* message. You can stop a measurement started with an INITiate command by sending a device clear (*see page 203*)

CONFigure Example 3

The following program segment sets the state of two input channels and then uses the ROUTe and READ? commands to take measurements on each. Channel 1 is set to the 10 Volt range with maximum resolution. Channel 2 is set to the 1 volt range. Note that resolution is common to both channels, the NPLC in effect will be the NPLC set by the last CONFigure command the meter receives.

```
CONFigure:VOLT:DC 10, MAX, (@FRONt1)
CONFigure:VOLT:DC 1, MAX, (@FRONt2)
ROUTe:TERMinals FRONt1
READ?
bus enter statement
ROUTe:TERMinals FRONt2
READ?
bus enter statement
```

The MEASure? and CONFigure Commands

See also "General Measurement Configuration," starting on page 68 in Chapter 3. Both the MEASure? and CONFigure commands reset measurement parameters to defaults. See page 143.

- For the *range* parameter, MIN selects the lowest range for the selected function; MAX selects the highest range; AUTO or DEF selects autoranging.
- For the *resolution* parameter, specify the resolution in the same units as the measurement function, *not in number of digits*. MIN selects the smallest value accepted, which gives the highest resolution; MAX selects the largest value accepted, which gives the least resolution; DEF selects the default resolution which is 0.000001 x range (10 PLC). *See the table on page 144*.
- *You must specify a range to use the resolution parameter*.
- For voltage measurements, you can also specify the input channel. The parameter *channel* can be set to FRONt1, FRONt2, 1, or 2. In the command, the channel must be preceded by the '@' symbol and enclosed in parenthesis ().

MEASure[:VOLTage][:DC]?
[{*<range>* | AUTO | MIN | MAX | DEF}]
[,{*<resolution>* | MIN | MAX | DEF}][,(@*<channel>*)]

This command presets and makes a voltage measurement with the specified range, resolution, and input channel. The reading is sent to the output buffer.

MEASure[:VOLTage][:DC]:RATio?
[{*<range>* | AUTO | MIN | MAX | DEF}][,{*<resolution>* | MIN | MAX | DEF}]

This command presets and makes a voltage ratio measurement with the specified range and resolution. The reading is sent to the output buffer. For ratio measurements, the specified range applies to the signal connected to the **Channel 1** terminals. Autoranging is selected for voltage measurements on the **Channel 2** terminals.

MEASure[:VOLTage][:DC]:DIFFerence?
[{*<range>* | AUTO | MIN | MAX | DEF}][,{*<resolution>* | MIN | MAX | DEF}]

This command presets and makes a difference measurement with the specified range and resolution. The reading is sent to the output buffer.

For difference measurements, the specified range applies to the signal connected to the Channel 1 terminals. Autoranging is selected for voltage measurements on the Channel 2 terminals.

```
MEASure:FRESistance?
MEASure:RESistance?
   [{<range> | AUTO | MIN | MAX | DEF}][,{<resolution> | MIN | MAX |
      DEF}]
```

This command presets and makes a 2-wire or 4-wire ohms measurement with the specified range and resolution. The reading is sent to the output buffer.

```
MEASure:TEMPerature?
   {TC | DEF},{B | E | J | K | N | R | S | T | DEF}[,1,{<resolution> |
      MIN | MAX | DEF}]
   {THER | DEF},{DEF}[,1,{<resolution> | MIN | MAX | DEF}]
   {FRTD | DEF},{85 | 91 | DEF}[,1,{<resolution> | MIN | MAX | DEF}]
```

This command presets and configures the meter for temperature measurements using the specified probe type and resolution. The reading is sent to the output buffer. DEF defaults to the last probe type set.

```
CONFigure[:VOLTage][:DC]
   [{<range> | AUTO | MIN | MAX | DEF}][,{<resolution> | MIN | MAX |
      DEF}][,(@<channel>)]
```

This command presets and configures the meter for dc voltage measurements with the specified range, resolution, and input channel. This command *does not* initiate the measurement.

```
CONFigure[:VOLTage][:DC]:RATio
   [{<range> | AUTO | MIN | MAX | DEF}][,{<resolution> | MIN | MAX |
      DEF}]
```

This command presets and configures the meter for dc:dc ratio measurements with the specified range and resolution. This command *does not* initiate the measurement. For ratio measurements, the specified range applies to the signal connected to the **Channel 1** terminals. Autoranging is selected for voltage measurements on the **Channel 2** terminals.

CONFigure[:VOLTage][:DC]:DIFFerence
[{*<range>* | AUTO | MIN | MAX | DEF}[,{*<resolution>* | MIN | MAX | DEF}]

This command presets and configures the meter for difference measurements with the specified range and resolution. This command *does not* initiate the measurement. For difference measurements, the specified range applies to the signal connected to the **Channel 1** terminals. Autoranging is selected for voltage measurements on the **Channel 2** terminals.

CONFigure:FRESistance
CONFigure:RESistance
[{*<range>* | AUTO | MIN | MAX | DEF}[,{*<resolution>* | MIN | MAX | DEF}]

This command presets and configures the meter for 2-wire or 4-wire ohms measurements with the specified range and resolution. This command *does not* initiate the measurement.

CONFigure:TEMPerature
{TC | DEF},{B | E | J | K | N | R | S | T | DEF}[,1,{*<resolution>* | MIN | MAX | DEF}]
{THER | DEF},{DEF}[,1,{*<resolution>* | MIN | MAX | DEF}]
{FRTD | DEF},{85 | 91 | DEF}[,1,{*<resolution>* | MIN | MAX | DEF}]

This command presets and configures the meter for temperature measurements using the specified probe type and resolution. This command *does not* initiate the measurement. DEF defaults to the last probe type set.

CONFigure?

This command queries the meter's present configuration and return a quoted string.

Setting the Function, Range, and Resolution

See also Chapter 3, "Features and Functions" starting on page 65.

[SENSe:]FUNCtion "*<function>*"

Select a measurement function. The function must be enclosed in quotes in the command string (for example, `FUNC "VOLT:DC"`). Use one of the following strings to set the function:

VOLTage:DC	**RESistance (2-wire ohms)**
VOLTage:DC:RATio	**FRESistance (4-wire ohms)**
VOLTage:DC:DIFFerence	**TEMPerature**

[SENSe:]FUNCtion?

This command queries the measurement function and return a quoted string.

[{SENSe1: | SENSe2:}]VOLTage[:DC]:RANGe {*<range>* | MIN | MAX}
[SENSe:]
FRESistance | RESistance
:RANGe {*<range>* | MIN | MAX}

This command sets the range for the function specified. MIN selects the lowest range. MAX selects the highest range. [volatile memory]

[{SENSe1: | SENSe2:}]VOLTage[:DC]:RANGe? [MIN | MAX]
[SENSe:]
FRESistance | RESistance
:RANGe? {MIN | MAX}

This command queries the function range setting.

```
[{SENSe1: | SENSe2:}]VOLTage[:DC]:RANGe:AUTO {OFF | ON}
[SENSe:]
   FRESistance | RESistance
      :RANGe:AUTO {OFF | ON}
```

This command disables or enables autoranging for the function. Autorange thresholds: Down range at <10% of range; Up range at >120% of range. [volatile memory]

```
[{SENSe1: | SENSe2:}]VOLTage[:DC]:RANGe:AUTO?
[SENSe:]
   FRESistance | RESistance
      :RANGe:AUTO?
```

This command queries the autorange setting for the specified function.

Returns “0” (OFF) or “1” (ON).

```
[SENSe1: | SENSe2:]VOLTage:[DC:]RESolution {<resolution> | MIN | MAX}
[SENSe:]
   FRESistance | RESistance
      :RESolution {<resolution> | MIN | MAX}
```

This command sets the resolution for the specified function. Specify the resolution in the same units as the measurement function, *not in number of digits*. MIN selects the smallest value accepted, which gives the most resolution. MAX selects the largest value accepted which gives the least resolution. [volatile memory]

```
[SENSe1: | SENSe2:]VOLTage:[DC:]RESolution? [{MIN | MAX}]
[SENSe:]
   FRESistance | RESistance
      :RESolution? [{MIN | MAX}]
```

This command queries the resolution for the selected function.

```
[SENSe1: | SENSe2:]VOLTage[:DC]:NPLCycles
   {0.02 | 0.2 | 1 | 2 | 10 | 20 | 100 | 200 | MIN | MAX}
[SENSe:]
   FRESistance | RESistance
      :NPLCycles {0.02 | 0.2 | 1 | 2 | 10 | 20 | 100 | 200 | MIN | MAX}
[SENSe:]TEMPerature:NPLCycles {0.02 | 0.2 | 1 | 2 | 10 | 20 | 100 | 200
   | MIN | MAX}
```

This command selects the integration time in number of power line cycles for the present function (the default is 10 PLC). MIN = 0.02. MAX = 200. [volatile memory]

```
[SENSe1: | SENSe2:]VOLTage[:DC]:NPLCycles? {MIN | MAX}
[SENSe:]
   FRESistance | RESistance
      :NPLCycles? {MIN | MAX}
[SENSe:]TEMPerature:NPLCycles? {MIN | MAX}
```

This command queries the integration time for the selected function.

Selecting the Input Channel

The correct input channel is automatically selected for resistance and temperature measurement functions. For voltage measurements, you must select the input channel.

The easiest, but least flexible, method to specify the input channel is by using the channel parameter in the MEASure? or CONFigure commands. For example,

```
CONFigure:VOLTage:DC MAX, MAX, (@FRONt1)    Channel 1
CONFigure:VOLTage:DC MAX, MAX, (@FRONt2)    Channel 2
```

In the SENSe subsystem, the keyword is modified to select the inputchannel. For example,

```
SENSe1:VOLTage:RANGe 10                     Channel 1 range setting
SENSe2:VOLTage:RANGe 10                     Channel 2 range setting
```

For increased flexibility, and to take advantage of the independant null and range settings for each input channel, you can set each input channel as desired and switch between the input channels using the ROUTe:TERMinals command. For example,

```
SENSe1:VOLTage:RANGe 10                     Channel 1 range 10 V
SENSe2:VOLTage:RANGe .1                     Channel 2 range 100 mV
ROUTe:TERMinals FRONt1                      Select channel 1
READ?                                       Channel 1 measurement
ROUTe:TERMinals FRONt2                      Select channel 2
READ?                                       Channel 2 measurement
```

You can determine which input channel is active with the query:

```
ROUTe:TERMinals?
```

The query returns either FRON1 or FRON2.

Special Resistance Measurement Commands

The following commands provide offset compensation and low power or voltage limited resistance measurement capability.

```
[SENSe:]
   FRESistance | RESistance
      :OCOMpensated {OFF | ON}
      :OCOMpensated?
```

This command enables or disables the offset compensated resistance measurements (*see page 82*). Once enabled, offset compensation will apply to both 4-wire and 2-wire resistance measurements.

```
[SENSe:]
   FRESistance
      :POWer:LIMit[:STATe] {OFF | ON}
      :POWer:LIMit[:STATe]?
```

This command enables or disable the 4-wire low power resistance mode (*see page 82*).

```
[SENSe:]
   FRESistance
      :VOLTage:LIMit[:STATe] {OFF | ON}
      :VOLTage:LIMit[:STATe]?
      :VOLTage:LIMit:VALue {<value> | MIN | MAX}
      :VOLTage:LIMit:VALue? {MIN | MAX}
```

This command enables or disables the 4-wire voltage limited resistance mode (*see* page 83). The voltage limit <*value*> parameter will be rounded to one of three values:

20 mV,
100 mV, or
500 mV

MIN sets 20 mV and MAX sets 500 mV.

Temperature Measurement Commands

See "Temperature Measurement Configuration" on page 84 *in Chapter 3*.

– The meter stores the last settings used for temperature measurements in non-volatile memory and uses these values as the default for future measurements. You can, therefore, make temperature measurements without having to set the transducer type or reference each time.

– A *range* setting is not meaningful for temperature measurements. The *range* parameter is included in the commands to allow you to set the resolution. For temperature measurements, set the *range* to 1. Other numbers can be entered and are ignored. The range is reported as 1 in response to a query.

```
MEASure:TEMPerature?
   {TC | DEF},{B | E | J | K | N | R | S | T | DEF}[, 1, {<resolution> |
      MIN | MAX | DEF}]
   {THER | DEF},DEF[, 1, {<resolution> | MIN | MAX | DEF}]
   {FRTD | DEF},{85 | 91 | DEF}[, 1, {<resolution> | MIN | MAX | DEF}]
```

These commands preset and configure the meter for temperature measurements using the specified probe type and resolution. The reading is sent to the output buffer. Set the measurement probe type in the first parameter, either TC (thermocouple), THER (Thermistor), or FRTD (four-wire RTD). The DEF (default) option sets the last probe type used. Set the thermocouple or RTD type in the second parameter. Set to DEF for thermistors. The optional parameter, **1**, is a place holder to allow you to set the resolution.

```
CONFigure:TEMPerature
   {TC | DEF},{B | E | J | K | N | R | S | T | DEF}[, 1, {<resolution> |
      MIN | MAX | DEF}]
   {THER | DEF},DEF[, 1, {<resolution> | MIN | MAX | DEF}]
   {FRTD | DEF},{85 | 91 | DEF}[, 1, {<resolution> | MIN | MAX | DEF}]
```

These commands preset and configure the meter for temperature measurements using the specified probe type and resolution. This command *does not* initiate the measurement. Set the measurement probe type in the first parameter, either TC (thermocouple), THER (Thermistor), or FRTD (four-wire RTD). The DEF (default) option sets the last probe type used. Set the thermocouple type in the second parameter. Set to DEF for thermistors. The optional parameter, 1, is a place holder to allow you to set the resolution.

[SENSe:]TEMPerature:TRANsducer:TYPE {TC | THER | FRTD | DEF}

This command sets the type of temperature transducer to use for temperature measurements. Choose TC (thermocouples), THER (thermistors), or FRTD (four-wire RTD). DEF sets FRTD.

[SENSe:]TEMPerature:TRANsducer:TYPE?

This command queries for the current temperature measurement transducer type.

[SENSe:]TEMPerature:TRANsducer:TCouple:TYPE {B | E | J | K | R | S | T | DEF}

This command sets the type of thermocouple to use for the temperature measurement. DEF sets the last thermocouple type used.

[SENSe:]TEMPerature:TRANsducer:TCouple:TYPE?

This command queries for the thermocouple type in use.

[SENSe:]TEMPerature:TRANsducer:TCouple:RJUNction {*<reference>* | MIN | MAX}

This command sets the reference junction temperature for thermocouple measurements when using a fixed temperature reference.

MIN sets 0 °C, MAX sets 55 °C.

[SENSe:]TEMPerature:TRANsducer:TCouple:RJUNction? [{MIN | MAX}]

This command queries for the reference junction temperature, if set, when using a fixed temperature reference.

MIN returns 0 °C, MAX returns 55 °C.

[SENSe:]TEMPerature:TRANsducer:TCouple:RJUNction:TYPE {FIX | INT | THER}

This command sets the reference junction type for thermocouple measurements. The type can be se to a fixed value (FIX), can be set to use the internal thermistor (INT), or can use a thermistor connected to channel 1 (THER). When type is set to FIX, use `SENSe:TEMPerature:TRANsducer:TCouple:RJUNction` *<value>* to set the reference temperature.

[SENSe:]TEMPerature:TRANsducer:TCouple:RJUNction:TYPE ?

This command queries for the type of reference junction being used for thermocouple measurements.

[SENSe:]TEMPerature:TRANsducer:FRTD:TYPE {85 | 91}

This command sets the four-wire RTD type. The alpha is entered as either 85 (for α = .000385) or 91 (for α = .000391).

[SENSe:]TEMPerature:TRANsducer:FRTD:TYPE?

This command queries for the type of RTD being used. Returns either +91 or +85.

[SENSe:]TEMPerature:TRANsducer:FRTD:RESistance[:REFerence] *<value>*

This command sets the four-wire RTD R_0 value. The value is in the range of 4.9 Ω to 2.1 kΩ.

[SENSe:]TEMPerature:TRANsducer:FRTD:RESistance[:REFerence]?

This command queries for the RTD R_0 value being used.

UNIT:TEMPerature {C | CEL | F | FAR | K}

This command sets the measurement and reporting units for temperature measurements.

UNIT:TEMPerature?

This command queries the temperature measurement reporting units.

Null (Relative) Commands

The meter uses independant null values for channel 1 and channel 2 voltage measurements, resistance measurements, and temperature measurements. *See page 90 in Chapter 3*.

```
[SENSe:] | SENSe1: | SENSe2
   VOLTage[:DC]:NULL[:STATe] {OFF | ON}
   VOLTage[:DC]:NULL[:STATe]?
   VOLTage[:DC]:NULL:VALue {<value> | MIN | MAX}
   VOLTage[:DC]:NULL:VALue?
```

These commands enable or disable the null state for voltage measurements on the selected input channel. Null must be enabled before the null value can be set. The null value set must be within ±132 V on each input channel.

```
[SENSe:]
   FRESistance | RESistance
      :NULL[:STATe] {OFF | ON}
      :NULL[:STATe]?
      :NULL:VALue {<value> | MIN | MAX}
      :NULL:VALue?
```

These commands enable or disable the null state for resistance measurements. Null must be enabled before the null value can be set. The null value set must be within ±1.2 MΩ. Null is common to both 2-wire and 4-wire resistance measurements, that is, enabling null on 4-wire ohms measurements also enables null on 2-wire ohms measurements.

```
[SENSe:]
   TEMPerature
      :NULL[:STATe] {OFF | ON}
      :NULL[:STATe]?
      :NULL:VALue {<value> | MIN | MAX}
      :NULL:VALue?
```

These commands enable or disable the null state for temperature measurements. Null must be enabled before the null value can be set. The null value set must be in the range of -300 °C to 2000 °C. Null is common to all temperature measurements, that is, enabling null for one probe type enables null for all probe types.

[SENSe:]NULL {OFF | ON | ONCE}

This command enables or disables null for the active channel or function. OFF or ON disables or enables null using a value set in the null register. ONCE enables null and makes the next reading the null value.

Input Filter Commands

See also page 68 in Chapter 3.

Using the input filters from the remote interface *is not* recommended. To use the filters with the remote interface be aware of the following:

- The digital filter is a moving average (boxcar) filter. The filter rate sets the number of readings that will be averaged. A reading is taken and included in the average with each trigger. You must, therefore, supply enough triggers to fill up the average before the filter will settle. If your measurement requires averaging readings, use the math operations (Stats) instead of the digital filter (*see page 164*).
- The digital filter average may reset, based upon the measurement value (*see page 70*). You must use the STATus:OPERation:CONDition register to determine if the filter is settled with each reading taken.
- The analog filter significantly slows the reading rate.

The input filters are disabled following a remote interface reset (*RST) or preset (SYStem:PREset).

```
INPut:FILTer
   :STATe {OFF | ON}
   :STATe?
```

These commands enable or disable the filter state and query the filter state.

```
INPut:FILTer
   :TYPE {ANALog | DIGital | BOTH}
   :TYPE?
```

These commands set the filter type and query the filter type.

BOTH enables the digital and analog filters.

```
INPut:FILTer
   :DIGital:RESPonse {SLOW | MEDium | FAST}
   :DIGital:RESPonse?
```

These commands set the averaging for the digital filter and query the digital filter setting. SLOW sets a 100 reading average, MEDium sets a 50 reading average, and FAST sets a 10 reading average.

```
INPut:FILTer
   :DIGital:PRECharge {OFF | ON}
   :DIGital:PRECharge?
```

These commands enable or disable the digital filter precharge and query the digital filter precharge setting. Disabling the digital filter precharge prevents the filter from resetting based upon the measured value (*see page 70 in Chapter 3*).

Math Operation Commands

See also "Math Operations" on page 87 *in Chapter 3*.

There are two main math operations available; stats and scale, only one of which can be enabled at a time. Stats performs mathematical operations on a series of readings. Scale performs a mathematical operation on each reading.

The selected math operation remains in effect until you disable it, change functions, turn off the power, or perform a remote interface reset.

CALCulate:FUNCtion {AVERage | SCALe}

This command selects the main math function. Only one function can be enabled at a time. [volatile memory]

CALCulate:FUNCtion?

This command queries the present math function. Returns SCAL or AVER.

CALCulate[:STATe] {OFF | ON}

This command disables or enables the selected math function. [volatile memory]

CALCulate[:STATe]?

This command queries the state of the math function.

Returns "0" (OFF) or "1" (ON).

Statistics operation commands (AVERage)

– You must have set CALC:FUNC AVER and CALC ON to use these commands.

CALCulate:AVERage:MINimum?

This command returns the minimum value found during a math average operation. The meter clears the value when math is turned on, when power has been off, or after a remote interface reset. [volatile memory]

CALCulate:AVERage:MAXimum?

This command returns the maximum value found during a math average operation. The meter clears the value when math is turned on, when power has been off, or after a remote interface reset. [volatile memory]

CALCulate:AVERage:AVERage?

This command returns the average of all readings taken since math average was enabled. The meter clears the value when math is turned on, when power has been off, or after a remote interface reset. [volatile memory]

CALCulate:AVERage:COUNt?

This command returns the number of readings taken since math average was enabled. The meter clears the value when math is turned on, when power has been off, or after a remote interface reset. [volatile memory]

CALCulate:AVERage:PTPeak?

This command returns the peak-to-peak value of all readings taken since math was enabled. The meter clears the value when math is turned on, when power has been off, or after a remote interface reset. [volatile memory]

CALulate:AVERage:SDEViation

This command returns the standard deviation of all readings taken since math was enabled. The meter clears the value when math is turned on, when power has been off, or after a remote interface reset. [volatile memory]

DATA:FEED RDG_STORE, {"CALCulate" | " "}

This command selects whether readings taken using the INITiate command are stored in the meter's internal memory (default) or not stored at all.

In the default state (DATA:FEED RDG_STORE, "CALC"), up to 1024 readings are stored in memory when INITiate is executed. The MEASure? and CONFigure commands automatically select "CALC".

With memory disabled (DATA:FEED RDG_STORE, " "), readings taken using INITiate are not stored. You can use this method to gather statistics on any number of readings (no 1024 reading limit). An error will be generated if you attempt to transfer readings to the output buffer using the FETCh? command.

DATA:FEED?

This command queries the reading memory state. Returns "CALC" or " ".

DATA:POINts?

This command queries the number of readings in memory.

Scale operation commands

You must have set CALC:FUNC SCAL and CALC ON to use these commands.

CALCulate:SCALe:GAIN *<gain>*

This comand sets the slope of the readings scale according to $y = mx + b$. The gain is the m value.

CALCulate:SCALe:GAIN?

This command queries the current gain.

CALCulate:SCALe:OFFSet *<offset>*

This command sets the offset of the readings scale according to $y = mx + b$. The offset is the b value.

CALCulate:SCALe:OFFSet?

This command queries the current offset.

Triggering

See also "Triggering" on page 93 *in Chapter 3*.

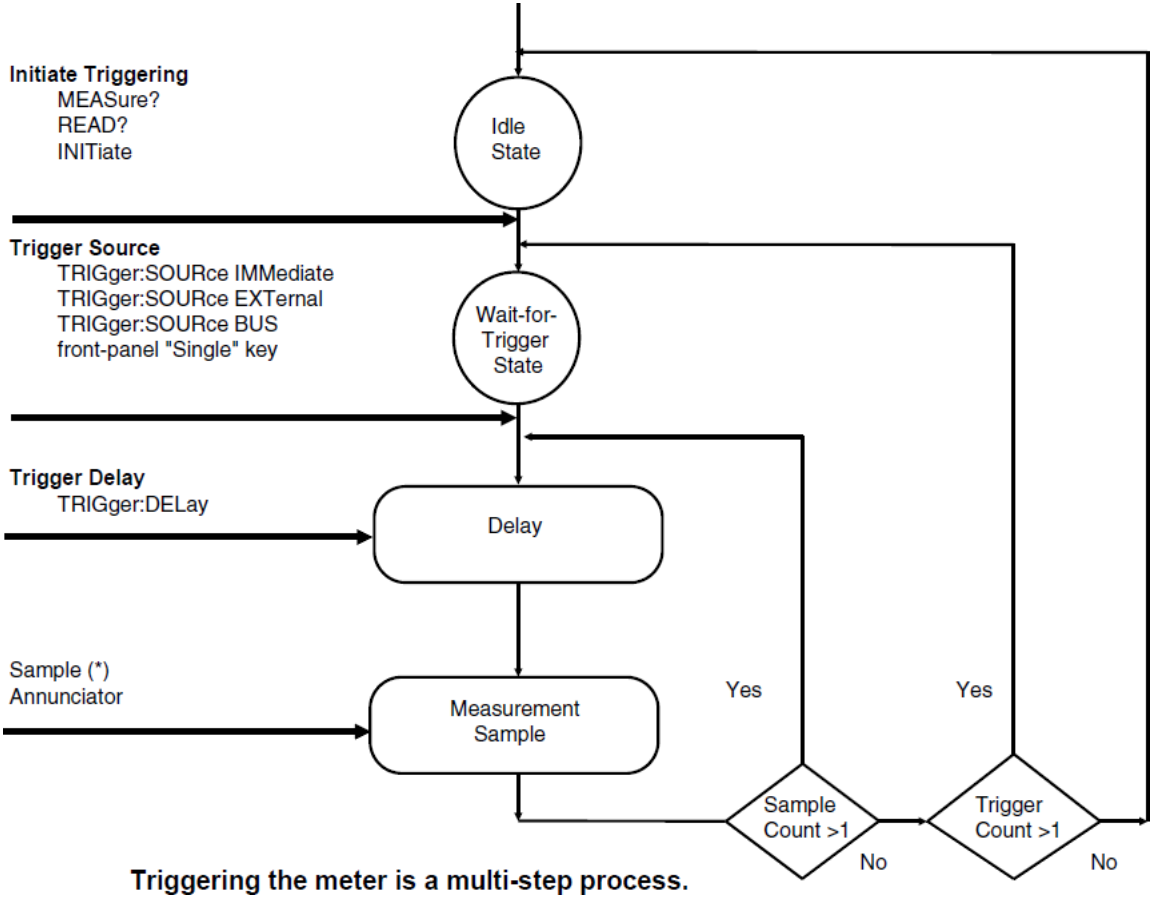
First-time SCPI users, see page 198.

The meter's triggering system allows you to generate triggers either manually or automatically, take multiple readings per trigger, and insert a delay before each reading. Normally, the meter will take one reading each time it receives a trigger, but you can specify multiple readings (up to 50,000) per trigger. You can also set an infinite trigger.

Triggering the meter from the remote interface is a multi-step process that offers triggering flexibility.

- First, you must configure the meter for the measurement by selecting the function, range, resolution, etc.
- Then, you must specify the source from which the meter will accept the trigger. The meter will accept a software (bus) trigger from the remote interface, a hardware trigger from the rear-panel *Ext Trig* (external trigger) terminal, or an immediate internal trigger.
- Then, you must make sure that the meter is ready to accept a trigger from the specified trigger source (this is called the *wait-for-trigger* state).

The diagram on the next page shows the meter's triggering system.



Triggering the meter is a multi-step process.

The Wait-for-Trigger State

After you have configured the meter and selected a trigger source, you must place the meter in the *wait-for-trigger* state. A trigger will not be accepted until the meter is in this state. If a trigger signal is present, and if the meter is in the "wait-for-trigger" state, the measurement sequence begins and readings are taken.

The "wait-for-trigger" state is a term used primarily for remote interface operation. From the front panel, the meter is always in the "wait-for-trigger" state and will accept triggers at any time, unless a measurement is already in progress.

You can place the meter in the "wait-for-trigger" state by executing any of the following commands from the remote interface.

```
MEASure?
READ?
INITiate
```

NOTE **The meter requires approximately 20 ms of set-up time after you send a command to change to the "wait-for-trigger" state. Any external triggers that occur during this set-up time are ignored.**

Triggering Commands

See also "Triggering" on page 93 *in Chapter 3*.

INITiate

This command changes the state of the triggering system from the "idle" state to the "wait-for-trigger" state. Measurements will begin when the specified trigger conditions are satisfied after the INITiate command is received. The readings are placed in the meter's internal memory (up to 1024 readings can be stored). Readings *are stored* in memory until you are able to retrieve them. Use the FETCh? command to retrieve reading results.

READ?

This command changes the state of the trigger system from the "idle" state to the "wait-for-trigger" state. Measurements will begin when the specified trigger conditions are satisfied following the receipt of the READ? command. Readings are sent immediately to the output buffer.

TRIGger:SOURce {BUS | IMMediate | EXTernal}

This command selects the source from which the meter will accept a trigger. The meter will accept a software (bus) trigger, an immediate internal trigger (this is the default source), or a hardware trigger from the rear-panel *Ext Trig* (external trigger) terminal. After you set the trigger source, you still need to send an INITiate command to move the meter to the *wait-for-trigger* state. [volatile memory]

TRIGger:SOURce?

This command queries the present trigger source. Returns "BUS", "IMM", or "EXT".

TRIGger:DELay {*<seconds>* | MIN | MAX}

This command inserts a trigger delay between the trigger signal and each sample that follows. If you do not specify a trigger delay, the meter automatically selects a delay for you (*see page 102*). Select from 0 to 3600 seconds. MIN = 0 seconds. MAX = 3600 seconds. [volatile memory]

TRIGger:DELay? [{MIN | MAX}]

This command queries the trigger delay.

TRIGger:DELay:AUTO {OFF | ON}

This command enables or disables an automatic trigger delay. The delay is determined by function, range, integration time, and ac filter setting. Selecting a specific trigger delay value automatically turns off automatic trigger delay. [volatile memory]

TRIGger:DELay:AUTO?

This command queries the automatic trigger delay setting.

Returns “0” (OFF) or “1” (ON).

SAMPle:COUNt {*<value>* | MIN | MAX}

This command sets the number of readings (samples) the meter takes per trigger. Select from 1 to 50,000 readings per trigger.

MIN = 1 and MAX = 50,000. [volatile memory]

SAMPle:COUNt? [{MIN | MAX}]

This command queries the sample count.

TRIGger:COUNt {*<value>* | MIN | MAX | INFinity}

This command sets the number of triggers the meter will accept before returning to the “idle” state. Select from 1 to 50,000 triggers. The INFinity parameter instructs the meter to continuously accept triggers (you must send a device clear to return to the “idle” state). Trigger count is ignored while in local operation. MIN = 1 and MAX = 50,000. [volatile memory]

TRIGger:COUNt? [{MIN | MAX}]

This command queries the trigger count. If you specify an infinity trigger count, the query command returns “9.90000000E+37”.

Chart Output (Analog Out) Commands

See also "Chart Output (Analog Output)" on page 111 *in Chapter 3*.

```
OUTPut
   [:STATe] {OFF | ON}
   [:STATe]?
```

These commands enable or disable the chart output (analog output) and query the output state. When disabled, the output is held at 0 V.

```
OUTPut
   :REFerence:OFFset {<number> | MIN | MAX}
   :REFerence:OFFset?
   :REFerence:NULL
```

These commands set or query the offset value. The offset can be set in the range of -1.0E9 to 1.0E9. MIN sets –1.0E9 V and MAX sets 1.0E9 V. The :NULL form of this command makes the next measurement value the null value.

```
OUTPut
   :REFerence:SPAN [{<number> | MIN | MAX}]
   :REFerence:SPAN? [{MIN | MAX}]
   :REFerence:GAIN?
```

These commands set or query the span value. The span can be set in the range of 5E-9 to 1.0E9. MIN sets 5E-9 and MAX sets 1.0E9. The :GAIN? query returns the reciprocal of the span value.

```
OUTPut
   :REFerence:VALue {<value> | MIN | MAX}]
```

This command allows you to set a fixed value output. The value can be set in the range of ±3.00 V. MIN sets -3.00 V and MAX sets +3.00 V.

System-Related Commands

See also "System-Related Operations" on page 103 *in Chapter 3*.

FETCh?

This command transfers readings stored in the meter's internal memory by the INITiate command to the meter's output buffer where you can read them into your bus controller.

READ?

This command changes the state of the trigger system from the "idle" state to the "wait-for-trigger" state. Measurements will begin when the specified trigger conditions are satisfied following the receipt of the READ? command. Readings are sent immediately to the output buffer.

DISPlay {OFF | ON}

This command turns the front-panel display off or on. [volatile memory]

DISPlay?

This command queries the front-panel display setting.

Returns "0" (OFF) or "1" (ON).

DISPlay:TEXT *<quoted string>*

This command displays a message on the front panel. The meter will display up to 11 characters in a message; any additional characters are truncated. [volatile memory]

DISPlay:TEXT?

This command queries the message sent to the front panel and returns a quoted string.

DISPlay:TEXT:CLEar

This command clears the message in the display.

SYSTem:ERRor?

This command queries the meter's error queue. Up to 20 errors can be stored in the queue. Errors are retrieved in first-in-first out (FIFO) order. Each error string may contain up to 80 characters.

SYSTem:VERSion?

This command queries the meter to determine the present SCPI version.

Returns "1994.0"

***RST**

This command resets the meter to its power-on configuration.

***TST?**

This command performs a complete self-test of the meter. Returns “0” if the self-test is successful, or “1” if the test fails.

***IDN?**

This command reads the meter's identification string (be sure to dimension a string variable with at least 40 characters).

The SCPI Status Model

All SCPI instruments implement status registers in the same way. The status system records various instrument conditions in four register groups: the Status Byte register, the Standard Event register, The Operational Status register, and the Questionable Data register. The status byte register records high-level summary information reported in the other register groups. The diagram on the next page illustrates the SCPI status system used by the meter.

Chapter 6, "Application Programs" contains example programs showing the use of the status registers. You may find it useful to refer to the programs after reading the following section in this chapter.

What is an *Event* Register?

The standard event and questionable data registers have *event registers*. An event register is a read-only register that reports defined conditions within the meter. Bits in the event registers *are* latched. Once an event bit is set, subsequent state changes are ignored. Bits in an event register are automatically cleared by a query of that register (such as *ESR? or STAT:QUES:EVEN?) or by sending the *CLS (clear status) command. A reset (*RST) or device clear will not clear bits in event registers. Querying an event register returns a decimal value which corresponds to the binary-weighted sum of all bits set in the register.

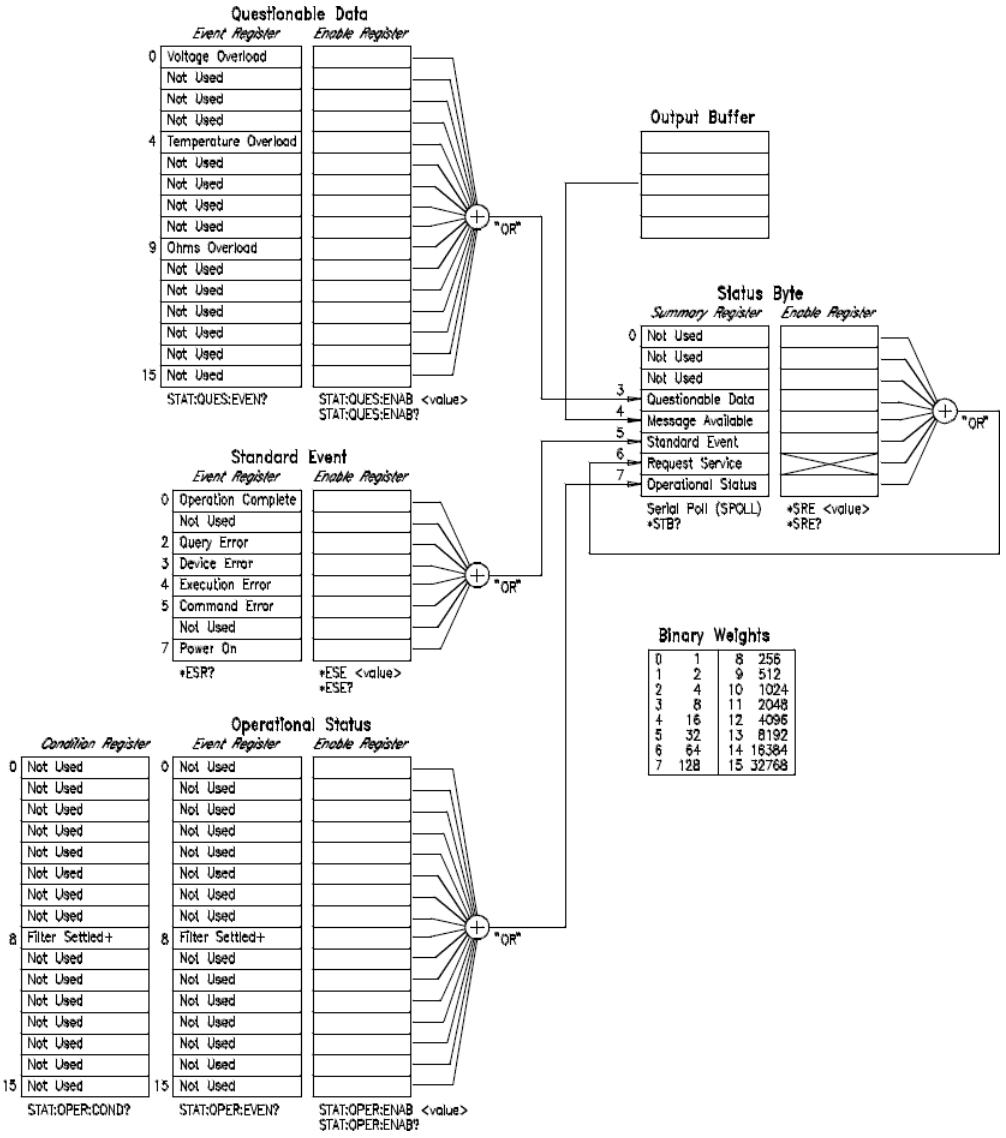
What is an *Enable* Register?

An *enable register* defines which bits in the corresponding event register are logically ORed together to form a single summary bit. Enable registers are both readable and writable. Querying an enable register *will not* clear it. The *CLS (clear status) command does not clear enable registers but it does clear the bits in the event registers. The STATus:PRESet command *will* clear the questionable data enable register. To enable bits in an enable register, you must write a decimal value which corresponds to the binary-weighted sum of the bits you wish to enable in the register.

What is a *Condition* Register?

A *condition register* actively indicates the current state of the instrument. Bits in the condition register **are not** latched, and change as the conditions within the instrument change. The event register latches the changes in the condition register bits.

SCPI Status System



+ See page 184 for information on the use of this bit.

The Status Byte

The status byte *summary register* reports conditions from other status registers. Query data that is waiting in the meter's output buffer is immediately reported through the "message available" bit (bit 4). Bits in the summary registers are *not* latched. Clearing an event register will clear the corresponding bits in the status byte summary register. Reading all messages in the output buffer, including any pending queries, will clear the message available bit.

Bit Definitions – Status Byte Register

Bit	Decimal Value	Definition
0 Not Used	1	Always set to 0
1 Not Used	2	Always set to 0
2 Not Used	4	Always set to 0
3 Questionable Data	8	One or more bits are set in the Questionable Data register (bits must be "enabled" in the enable register).
4 Message Available	16	Data is available in the meter output buffer.
5 Standard Event	32	One or more bits are set in the Standard Event register (bits must be "enabled" in the enable register).
6 Request Service	64	The meter is requesting service (serial poll).
7 Operational Status	128	One or more bits are set in the Operational Event register (bits must be "enabled" in the enable register).

The status byte *summary register* is cleared when:

- You execute a *CLS (clear status) command.
- Querying the standard event and questionable data registers will clear only the respective bits in the summary register.

The status byte *enable register* (request service) is cleared when:

- You turn on the power and you have previously configured the meter using the *PSC 1 command.
- You execute a *SRE 0 command.

The status byte enable register *will not* be cleared at power-on if you have previously configured the meter using *PSC 0.

Using Service Request (SRQ) and Serial POLL

You must configure your bus controller to respond to the IEEE-488 service request (SRQ) interrupt to use this capability. Use the status byte enable register (SRE) to select which summary bits will set the low-level IEEE-488 SRQ signal. When the status byte "request service" bit (bit 6) is set, an IEEE-488 SRQ interrupt message is automatically sent to the bus controller. The bus controller may then poll the instruments on the bus to identify which one requested service (the one with bit 6 set in its status byte). The request service bit is only cleared by reading the status byte using an IEEE-488 serial poll or by reading the event register whose summary bit is causing the service request.

To read the status byte summary register, send the IEEE-488 serial poll message. Querying the summary register will return a decimal value which corresponds to the binary-weighted sum of the bits set in the register. Serial poll will automatically clear the "request service" bit in the status byte summary register. No other bits are affected. Performing a serial poll will not affect instrument throughput.

CAUTION **The IEEE-488.2 standard does not ensure synchronization between your bus controller program and the instrument. Use the *OPC? command to guarantee that commands previously sent to the instrument have completed. Executing a serial poll before a *RST, *CLS, or other commands have completed can cause previous conditions to be reported.**

Using *STB? to Read the Status Byte

The *STB? (status byte query) command is similar to a serial poll except it is processed like any other instrument command. The *STB? command returns the same result as an IEEE-488 serial poll except that the "request service" bit (bit 6) *is not* cleared if a serial poll has occurred. The *STB? command is not handled automatically by the IEEE-488 bus interface hardware and the command will be executed *only* after previous commands have completed. Polling is not possible using the *STB? command. Using the *STB? command does not clear the status byte summary register.

To Interrupt Your Bus Controller Using SRQ

- Send a bus device clear message.
- Clear the event registers with the *CLS (clear status) command.
- Set the *ESE (standard event register) and *SRE (status byte register) enable masks.
- Send the *OPC? (operation complete query) command and enter the result to assure synchronization.
- Enable your bus controller's IEEE-488 SRQ interrupt.

To Determine When a Command Sequence is Completed

Either:

- Send the *OPC? (operation complete query) command and enter the result to assure synchronization.

  Or, to prevent holding the interface bus for commands that take longer to execute (such as CALibrate commands), :

- Send a device clear message to clear the meter's output buffer.
- Clear the event registers with the *CLS (clear status) command.
- Enable "operation complete" using the *ESE 1 command (standard event register).
- Send your programming command string, and place the *OPC (operation complete) command as the last command.

– Use a serial poll to check to see when bit 5 (standard event) is set in the status byte summary register. You could also configure the meter for an SRQ interrupt by sending *SRE 32 (status byte enable register, bit 5).

How to Use the Message Available Bit (MAV)

You can use the status byte "message available" bit (bit 4) to determine when data becomes available to read into your bus controller. The meter sets bit 4 when the first reading trigger occurs (which can be TRIGger:SOURce:IMMediate). The meter subsequently clears bit 4 *only* after all messages have been read from the output buffer.

The message available (MAV) bit can only indicate when the *first* reading is available following a READ? command. This can be helpful if you do not know when a trigger event such as BUS or EXTernal will occur.

The MAV bit is set only after *all* specified measurements have completed when using the INITiate command followed by FETCh?. Readings are placed in the meter's internal memory when using INITiate. Sending the FETCh? command transfers readings (stored in internal memory by the INITiate command) to the meter's output buffer. Therefore, the MAV bit can only be set after *all* measurements have been completed.

Using *OPC to Signal When Data is in the Output Buffer

Generally, it is best to use the "operation complete" bit (bit 0) in the standard event register to signal when a command sequence is completed. This bit is set in the register after an *OPC command has been executed. If you send *OPC after a command which loads a message in the meter's output buffer (either reading data or query data), you can use the operation complete bit to determine when the message is available. However, if too many messages are generated before the *OPC command executes (sequentially), the output buffer will fill and the meter will stop taking readings.

The Standard Event Register

The *standard event* register reports the following types of instrument events: power-on detected, command syntax errors, command execution errors, self-test or calibration errors, query errors, or when an *OPC command is executed. Any or all of these conditions can be reported in the standard event summary bit through the enable register. You must write a decimal value using the *ESE (event status enable) command to set the enable register mask.

NOTE

An error condition (standard event register bits 2, 3, 4, or 5) will always record one or more errors in the meter's error queue, with the following exception:

A reading overload condition is always reported in both the standard event register (bit 3) and the questionable data event register (bits 0 or 9). However, no error message is recorded in the meter's error queue.

Read the error queue using SYSTem:ERRor?.

Bit Definitions – Standard Event Register

Bit	Decimal Value	Definition
0 Operation Complete	1	All commands prior to and including an *OPC command have been executed.
1 Not Used	2	Always set to 0.
2 Query Error	4	The meter tried to read the output buffer but it was empty. Or, a new command line was received before a previous query had been read. Or, both the input and output buffers are full.
3 Device Error	8	A self-test, calibration, or reading overload error occurred.
4 Execution Error	16	An execution error occurred (see error numbers -211 through -230 in Chapter 5).
5 Command Error	32	A command syntax error occurred (see error numbers -101 through -158 in Chapter 5).
6 Not Used	64	Always set to 0.
7 Power On	128	Power has been turned off and on since the last time the event register was read or cleared.

The standard *event register* is cleared when:

- You send a *CLS (clear status) command.
- You query the event register using the *ESR? (event status register) command.

The standard event *enable register* is cleared when:

- You turn on the power and you have previously configured the meter using the *PSC 1 command.
- You execute a *ESE 0 command.

The standard event enable register *will not* be cleared at power-on if you have previously configured the meter using *PSC 0.

The Questionable Data Register

The *questionable data* register provides information about the quality of the meter's measurement results. Overload conditions can be reported in the questionable data summary bit through the enable register. You must write a decimal value using the STATus:QUEStionable:ENABle command to set the enable register mask.

NOTE **A reading overload condition is always reported in both the standard event register (bit 3) and the questionable data event register (bits 0, 1, or 9). However, no error message is recorded in the meter's error queue.**

Bit Definitions – Questionable Data Register

Bit	Decimal Value	Definition
0 Voltage Overload	**1**	**Range overload on dc volts**
1 Not Used	2	Always set to 0.
2 Not Used	4	Always set to 0.
3 Not Used	8	Always set to 0.
4 Temperature Overload	**16**	**Range overload on temperature measurements.**
5 Not Used	32	Always set to 0.
6 Not Used	64	Always set to 0.

Bit	Decimal Value	Definition
7 Not Used	128	Always set to 0.
8 Not Used	256	Always set to 0.
9 Ohms Overload	**512**	**Range overload on 2-wire or 4-wire Ohms**
10 Not Used	1024	Always set to 0.
11 Not Used	2048	Always set to 0.
12 Not Used	4096	Always set to 0.
13 Not Used	8192	Always set to 0.
14 Not Used	16384	Always set to 0.
15 Not Used	32768	Always set to 0.

The questionable data *event register* is cleared when:

- You execute a *CLS (clear status) command.
- You query the event register using STATus:QUEStionable:EVENt?.

The questionable data *enable register* is cleared when

- You turn on the power (*PSC does not apply).
- You execute the STATus:PRESet command.
- You execute the STATus:QUEStionable:ENABle 0 command.

The Operational Status and Condition Registers

The operation status register provides information about the operation of the meter. Bits in the operational status event register are latched from changes in the bits in the operational status condition register.

The meter only uses a single bit (bit 8) in these registers. This bit indicates the settling state of the digital filter. If the digital filter is enabled, bit 8 is set to 1 to indicate that the filter has settled. *See page 70 for information about the operation of the digital filter.*

You can read the operational status condition register to determine the digital filter state on a per measurement basis. You can read the operational status event

register to determined if the digital filter *ever* settled. The digital filter count can be reset based upon input signal levels. *See page 70*.

NOTE **If you need to know the digital filter state for a given measurement, use the condition register (STATus:OPERation:CONDition?). This bit changes with each measurement. Changes to this bit are latched in the operational status event register and so the event register bit only indicates if the filter ever settled.**

Bit Definitions – Operational Status Register

	Bit	Decimal Value	Definition
0	Not Used		Always set to 0
1	↓		↓
↓			
7			
8	Filter Settled	256	Digital filter is settled.
9	Not Used		Always set to 0.
↓	↓		↓
15			

The operational status *event register* is cleared when:

- You execute a *CLS (clear status) command.
- You query the event register using STATus:OPERation:EVENt?.

The operational status *enable register* is cleared when:

- You turn on the power (*PSC does not apply).
- You execute the STATus:PRESet command.
- You execute the STATus:OPERation:ENABle 0 command.

Status Reporting Commands

SYSTem:ERRor?

This command queries the meter's error queue. Up to 20 errors can be stored in the queue. Errors are retrieved in first-in-first out (FIFO) order. Each error string may contain up to 80 characters.

STATus:QUEStionable:CONDition?

This command queries the questionable status condition register. The meter returns a decimal value which corresponds to the binary-weighted sum of all bits set in the register.

STATus:QUEStionable:ENABle *<enable value>*

This command enables bits in the Questionable Data enable register. The selected bits are then reported to the Status Byte.

STATus:QUEStionable:ENABle?

This command queries the Questionable Data enable register. The meter returns a binary-weighted decimal representing the bits set in the enable register.

STATus:QUEStionable:EVENt?

This command queries the Questionable Data event register. The meter returns a decimal value which corresponds to the binary-weighted sum of all bits set in the register.

STATus:OPERation:CONDition?

This command queries the operational status condition register. The meter returns a decimal value which corresponds to the binary-weighted sum of all bits set in the register. Only bit 8 (decimal 256) is used (indicates the digital filter has settled).

STATus:OPERation:ENABle *<enable value>*

This command enables bits in the Questionable Data enable register. The selected bits are then reported to the Status Byte.

STATus:OPERation:ENABle?

This command queries the Questionable Data enable register. The meter returns a binary-weighted decimal representing the bits set in the enable register.

STATus:OPERation:EVENt?

This command queries the Questionable Data event register. The meter returns a decimal value which corresponds to the binary-weighted sum of all bits set in the register. Only bit 8 (decimal 256) is used and latches the digital filter settled state of the condition register.

STATus:PRESet

This command clears all bits in the Questionable Data enable and Operational Status enable registers.

***CLS**

This command clears the Status Byte summary register and all event registers.

ESE** ***<enable value>

This command enables bits in the Standard Event enable register. The selected bits are then reported to the Status Byte.

***ESE?**

This command queries the Standard Event enable register. The meter returns a decimal value which corresponds to the binary-weighted sum of all bits set in the register.

***ESR?**

This command queries the Standard event register. The meter returns a decimal value which corresponds to the binary-weighted sum of all bits set in the register.

***OPC**

This command sets the “operation complete” bit (bit 0) in the Standard Event register after the command is executed.

***OPC?**

This command returns “1” to the output buffer after the command is executed.

***PSC {0 | 1}**

(Power-on status clear.) This command clears the Status Byte and Standard Event register enable masks when power is turned on (*PSC 1). When *PSC 0 is in effect, the Status Byte and Standard Event register enable masks *are not* cleared when power is turned on. [non-volatile memory]

***PSC?**

This command queries the power-on status clear setting.

Returns “0” (*PSC 0) or “1” (*PSC 1).

SRE *<enable value>

This command enables bits in the Status Byte enable register.

***SRE?**

This command queries the Status Byte enable register. The meter returns a decimal value which corresponds to the binary-weighted sum of all bits set in the register.

Calibration Commands

See the Service Guide for a more detailed description of the meter's calibration procedures.

CALibration?

This command performs a calibration using the specified calibration value (CALibration:VALue command).

CALibration:COUNt?

This command queries the meter to determine the number of times it has been calibrated. Since the value increments by one for each calibration point, a complete calibration increases the value by several counts. [non-volatile memory]

CALibration:ICURrent?

This command begins an injected current calibration. The query returns +0 if the calibration was successful and a +1 if the calibration failed.

CALibration:OUTPut {ZERO | GAIN}

This command sets the Chart Output (Analog Output) to either 0V or 1V output for calibration purposes. Use the CALibration:VALue and CALibration? commands to set the values and perform the calibration.

CALibration:OUTPut?

This command query returns "ZERO" or "GAIN" .

CALibration:SECure:CODE *<new code>*

This command enters a new security code. To change the security code, first unsecure the meter using the old security code. Then, enter the new code. The calibration code may contain up to 11 characters. [non-volatile memory]

CALibration:SECure:STATe {OFF | ON}, *<code>*

This command unsecures or secures the meter for calibration. The calibration code may contain up to 11 characters. [non-volatile memory]

CALibration:SECure:STATe?

This command queries the secured state of the meter.

Returns “0” (OFF) or “1” (ON).

CALibration:STRing *<quoted string>*

This command records calibration information about your meter. For example, you can store such information as the last calibration date or the next calibration due date. The calibration message may contain up to 40 characters. [non-volatile memory]

CALibration:STRing?

This command queries the calibration message and return a quoted string.

CALibration:VALue *<value>*

This command specifies the value of the known calibration signal used by the calibration procedure.

CALibration:VALue?

This command queries the present calibration value.

RS-232 Interface Configuration

See also "Remote Interface Configuration" on page 115 *in Chapter 3*.

You connect the meter to the RS-232 interface using the 9-pin (DB-9) serial connector on the rear panel. The meter is configured as a DTE (*Data Terminal Equipment*) device. For all communications over the RS-232 interface, the meter uses two handshake lines: DTR (*Data Terminal Ready*) on pin 4 and DSR (*Data Set Ready*) on pin 6.

The following sections contain information to help you use the meter over the RS-232 interface. The programming commands for RS-232 are listed beginning on page 197.

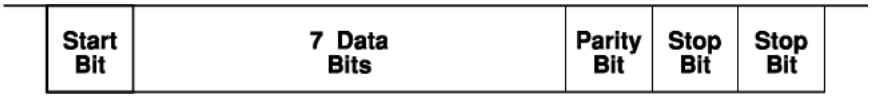
RS-232 Configuration Overview

Configure the RS-232 interface using the parameters shown below. Use the front-panel I/O MENU to select the baud rate, parity, and number of data bits (*see page 115 for more information*).

- Baud Rate: 300, 600, 1200, 2400, 4800, or **9600 baud** (*factory setting*)
- Parity and Data Bits: **None / 8 data bits** (*factory setting*)
  Even / 7 data bits, or
  Odd / 7 data bits
- Number of Start Bits **1 bit** (*fixed*)
- Number of Stop Bits **2 bits** (*fixed*)

RS-232 Data Frame Format

A character frame consists of all the transmitted bits that make up a single character. The frame is defined as the characters from the *start bit* to the last *stop bit*, inclusively. Within the frame, you can select the baud rate, number of data bits, and parity type. The meter uses the following frame formats for seven and eight data bits.



Connection to a Computer or Terminal

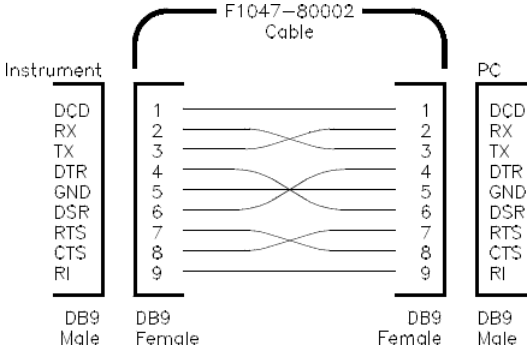
To connect the meter to a computer or terminal, you must have the proper interface cable. Most computers and terminal are DTE (*Data Terminal Equipment*) devices. Since the meter is also a DTE device, you must use a DTE-to-DTE interface cable. These cables are also called *null-modem, modem-eliminator, or crossover cables*.

The interface cable must also have the proper connector on each end and the internal wiring must be correct. Connectors typically have 9 pins (DB-9 connector) or 25 pins (DB-25 connector) with a "male" or "female" pin configuration. A male connector has pins inside the connector shell and a female connector has holes inside the connector shell.

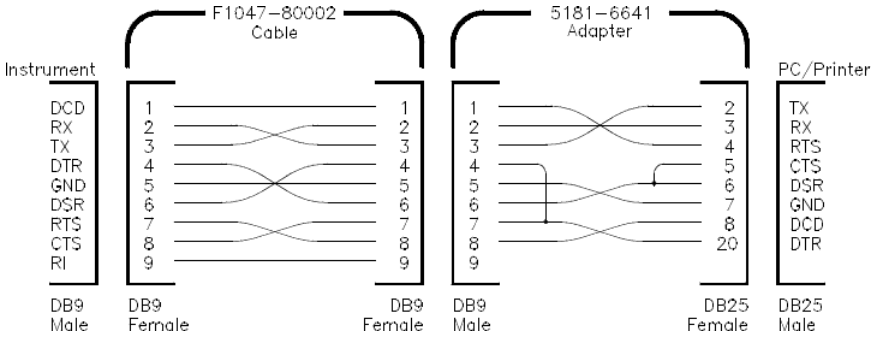
If you cannot find the correct cable for your configuration, you may have to use a *wiring adapter*. If you are using a DTE-to-DTE cable, make sure the adapter is a "straight-through" type. Typical adapters include gender changers, null-modem adapters, and DB-9 to DB-25 adapters.

Refer to the cable and adapter diagrams below to connect the meter to most computers or terminals. If you configuration is different than those described, order the *34399A Adapter Kit*. This kit contains adapters for connection to other computers, terminals, and modems. Instructions and pin diagrams are included with the adapter kit.

DB-9 Serial Connection If your computer or terminal has a 9-pin serial port with a male connector, use the null-modem cable included with the *34398A Cable Kit*. This cable has a 9-pin female connector on each end. The cable pin diagram is shownbelow.



DB-25 Serial Connection If your computer or terminal has a 25-pin serial port with a male connector, use the null-modem cable and 25-pin adapter included with the *34398A Cable Kit*. The cable pin diagram is shown below.



Connection to a Printer

To connect to a printer you must use a DTE to DCE (*Data Communications Equipment*) cable.

Set the meter to the TALK ONLY mode by setting the GPIB address to 31 from the front panel. Then set the RS-232 interface as the active interface. *See page 204 and page 207*.

DTR / DSR Handshake Protocol

The meter is configured as a DTE (*Data Terminal Equipment*) device and uses the DTR (*Data Terminal Ready*) and DSR (*Data Set Ready*) lines of the RS-232 interface to handshake. The meter uses the DTR line to send a hold-off signal. The DTR line must be TRUE before the meter will accept data from the interface. When the meter sets the DTR line FALSE, the data must cease within 10 characters.

To disable the DTR/DSR handshake, *do not* connect the DTR line and tie the DSR line to logic TRUE. If you disable the DTR/DSR handshake, also select a slower baud rate (300, 600, or 1200 baud) to ensure that the data is transmitted correctly.

The meter sets the DTR line FALSE in the following cases:

1. When the meter's input buffer is full (when approximately 100 characters have been received), it sets the DTR line FALSE (pin 4 on the RS-232 connector). When enough characters have been removed to make space in the input buffer, the meter sets the DTR line TRUE, unless the second case (see below) prevents this.
2. When the meter wants to "talk" over the interface (which means that it has processed a query) and has received a <*new line*> message terminator, it will set the DTR line FALSE. This implies that once a query has been sent to the meter, the controller should read the response before attempting to send more data. It also means that a <*new line*> must terminate the command string. After the response has been output, the meter sets the DTR line TRUE again, unless the first case (see above) prevents this.

   The meter monitors the DSR line to determine when the controller is ready to accept data over the interface. The meter monitors the DSR line (pin 6 on the RS-232 connector) before each character is sent. The output is suspended if the DSR line is FALSE. When the DSR line goes TRUE, transmission will resume.

   The meter holds the DTR line FALSE while output is suspended. A form of interface *deadlock* exists until the controller asserts the DSR line TRUE to allow the meter to complete the transmission. You break the interface deadlock by sending the <*Ctrl-C*> character, which clears the operation in progress and discards pending output (this is equivalent the the IEEE-488 device clear action).

RS-232 Troubleshooting

Here are a few things to check if you are having problems using the RS-232 interface. If you need additional help, refer to the documentation that came with your computer.

- Verify that the meter and your computer are configured for the same baud rate, parity, and number of data bits. Make sure that your computer is set up for *1 start bit and 2 stop bits* (these values are fixed on the meter).
- Make sure to execute the SYSTEM:REMOTE command to place the meter in the REMOTE mode.

- Verify that you have connected the correct interface cable and adapters. Even if the cable has the proper connectors for your system, the internal wiring may not be correct. The *34398A Cable Kit* can be used to connect the meter to most computers or terminals.
- Verify that you have connected the interface cable to the correct serial port on your computer (COM1, COM2, etc).

RS-232 Interface Commands

SYSTem:LOCal

This command places the meter in the local mode during RS-232 operation. All keys on the front panel are fully functional.

SYSTem:REMote

This command places the meter in the remote mode during RS-232 operation. All keys on the front panel except the LOCAL key are disabled. You MUST send this command before any other RS-232 command can be received.

SYSTem:RWLock

This command places the meter in the remote mode during RS-232 operation. All keys on the front panel are disabled, including the LOCAL key.

CAUTION Sending or receiving data over the RS-232 interface when not configured for REMOTE operation can cause unpredictable results. Always make sure the meter is configured for remote operation when using the RS-232 interface.

An Introduction to the SCPI Language

Standard Commands for Programmable Instruments (SCPI) defines how you communicate with an instrument from a bus controller. The SCPI language uses a hierarchical structure similar to the file systems used by many bus controllers. The command "tree" is organized with root-level commands (also called "subsystems") positioned at the top, with multiple levels below each root-level command. You must specify the complete path to execute the individual lower-level commands.

Using a Colon (:) When a colon is the first character of a command keyword, it indicates that the next command mnemonic is a root-level command. When a colon is inserted between two command mnemonics, the colon moves the path down one level in the present path (for the specified root-level command) of the command tree. You *must* separate command mnemonics from each other using a colon. *You can omit the leading colon if the command is the first of a new program line*.

Using a Semicolon (;) Use a semicolon to separate two commands within the same command string. The semicolon does not change the present path specified. For example, the following two statements are equivalent.

```
:TRIG:DELAY 1;:TRIG:COUNT 10
:TRIG:DELAY 1;COUNT 10
```

Using a Comma (,) If a command requires more than one parameter, you must separate adjacent parameters using a comma.

Using Whitespace You *must* use whitespace characters, [tab], or [space] to separate a parameter from a command keyword. Whitespace characters are generally ignored only in parameter lists.

Using " ? " Commands The bus controller may send commands at any time, but a SCPI instrument may only send responses when *specifically* instructed to do so. Only query commands (commands that end with a "?") will instruct the instrument to send a response message. Queries return either measured values or internal instrument settings.

CAUTION **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.**

Using " * " Commands Commands starting with a " * " are called *common commands*. They are required to perform the identical function for all instruments that are compliant with the IEEE-488.2 interface standard. The " * " commands are used to control reset, self-test, and status operations in the meter.

SCPI Data Types

The SCPI language defines different data formats for use in program messages and response messages. Instruments are flexible listeners and can accept commands and parameters in various formats. However, SCPI instruments are precise talkers. This means that SCPI instruments will always respond to a particular query in a predefined, rigid format.

Numeric Parameters Commands that require numeric parameters will accept all commonly used decimal representations of numbers including optional signs, decimal points, and scientific notation. Special values for numeric parameters like MAXimum, MINimum, and DEFault are also accepted. You can also send engineering unit suffixes (M, k, or u) with numeric parameters. If only specific numeric values are accepted, the meter will automatically round the input numeric parameters.

Discrete Parameters Discrete parameters are used to program settings that have a limited number of values (like BUS, IMMediate, EXTernal). They have a short form and a long form just like command keywords. You can mix upper-case and lower-case letters. Query responses will *always* return the short form in upper-case letters.

Boolean Parameters Boolean parameters represent a single binary condition that is either true or false. For a false condition, the meter will accept “OFF” or “0”. For a true condition, the meter will accept “ON” or “1”. When you query a boolean setting, the instrument will *always* return “0” or “1”.

String Parameters String parameters can contain virtually any set of ASCII characters. A string *must* begin and end with matching quotes; either with a single quote or with a double quote. You can include the quote delimiter as part of the string by typing it twice without any characters in between.

Input Message Terminators

Program messages sent to a SCPI instrument *must* terminate with a <newline> character. The IEEE-488 EOI (end or identify) signal is interpreted as a <newline> character and may also be used to terminate a message in place of the <newline> character. A <carriage return> followed by a <newline> is also accepted. Many programming languages allow you to specify a message terminator character or EOI state to be automatically sent with each bus transaction. Message termination will *always* set the current path back to the root-level.

Output Data Formats

Output data will be in one of formats shown in the table below.

Type of Output Data	Output Data Format
Non-reading queries	<80 ASCII character string
Single reading (IEEE-488)	SD.DDDDDDDDESDD<nl>
Multiple readings (IEEE-488)	SD.DDDDDDDDESDD,...,... <nl>
Single reading (RS-232)	SD.DDDDDDDDESDD<cr><nl>
Multiple readings (RS-232)	SD.DDDDDDDDESDD,...,...<cr><nl>
	S Negative sign or positive sign
	D Numeric digits
	E Exponent
	<nl> newline character
	<cr> carriage return character

Using Device Clear to Halt Measurements

Device clear is an IEEE-488 low-level bus message which can be used to halt measurements in progress. Different programming languages and IEEE-488 interface cards provide access to this capability through their own unique commands. The status registers, the error queue, and all configuration states are left unchanged when a device clear message is received. Device clear performs the following actions.

- All measurements in progress are aborted.
- The meter returns to the trigger "idle state".
- The meter's input and output buffers are cleared.
- The meter is prepared to accept a new command string.
- For RS-232 operation, sending the control-C character will perform the equivalent operations of the IEEE-488 device clear message. The meter's DTR (data terminal ready) handshake line will be true following a device clear message. *See "DTR / DSR Handshake Protocol"* on page 194 *for further details*.

TALK ONLY for Printers

You can set the meter's GPIB address to "31" which is the *talk only* mode. In this mode, the meter can output readings directly to a printer without being addressed by a bus controller. Address 31 is not a valid address if you are operating the meter from the GPIB interface with a bus controller.

If you select the RS-232 interface and then set the GPIB address to the talk only address (31), the meter will *send* readings over the RS-232 interface when in the local mode.

To Set the GPIB Address

Each device on the GPIB (IEEE-488) interface must have a unique address. You can set the meter's address to any value between 0 and 31. The address is set to "22" when the meter is shipped from the factory. The address is displayed on the front panel when you turn on the meter.

1 Turn on the front-panel menu.

Shift (On/Off) <

```
A: MEAS MENU
```

2 Move across to the I/O MENU choice on this level.

< <

```
F: I/O MENU
```

3 Move down a level to the GPIB ADDR command.

∨

```
1: GPIB ADDR
```

4 Move down to the "parameter" level to set the address.

∨

Use the left/right and down/up arrow keys to change the address.

```
^22 ADDR
```

5 Save the change and turn off the menu.

Auto/Man
ENTER

The address is stored in *non-volatile* memory, and *does not* change when power has been off or after a remote interface reset.

To Select the Remote Interface

The meter is shipped with both a GPIB (IEEE-488) interface and an RS-232 interface. Only one interface can be enabled at a time. The GPIB interface is selected when the meter is shipped from the factory. *See also "Remote Interface Configuration"* on page 115.

1 Turn on the front-panel menu.

Shift (On/Off) <

```
A: MEAS MENU
```

2 Move across to the I/O MENU choice on this level.

< <

```
F: I/O MENU
```

3 Move down a level and then across to the INTERFACE command.

v >

```
2: INTERFACE
```

4 Move down to the "parameter" level to select the interface.

v

```
GPIB / 488
```

Use the left/right arrow keys to see the interface choices. Choose from the following: GPIB / 488 or RS-232.

5 Save the change and turn off the menu.

Auto/Man (ENTER)

The interface selection is stored in non-volatile memory, and does not change when power has been off or after a remote interface reset.

Remote Interface: you can set or query the interface using the following commands:

```
SYStem
   :INTerface {GPIB | RS232}
   :INTerface?
```

To Set the Baud Rate

You can select one of six baud rates for RS-232 operation. The rate is set to 9600 baud when the meter is shipped from the factory. *See also* *"Baud Rate Selection (RS-232)"* on page 117.

1 Turn on the front-panel menu.

Shift On/Off <

```
A: MEAS MENU
```

2 Move across to the I/O MENU choice on this level.

< <

```
F: I/O MENU
```

3 Move down a level and then across to the BAUD RATE command.

v > >

```
3: BAUD RATE
```

4 Move down to the "parameter" level to select the baud rate.

v

Use the left/right arrow keys to see the baud rate choices. Choose from one of the following: 300, 600, 1200, 2400, 4800, or 9600 baud.

```
9600 BAUD
```

5 Save the change and exit the menu.

Auto/Man
ENTER

The baud rate selection is stored in *non-volatile* memory, and *does not* change when power has been off or after a remote interface reset.

Remote Interface: you can set or query the baud rate using the following commands:

```
SYStem:COMMunicate:SERial
   :BAUD <baud>
   :BAUD?
```

To Set the Parity

You can select the parity for RS-232 operation. The meter is configured for even parity with 7 data bits when shipped from the factory. *See also* "*Parity Selection (RS-232)*" on page 118.

1 Turn on the front-panel menu.

Shift On/Off <

```
A: MEAS MENU
```

2 Move across to the I/O MENU choice on this level.

< <

```
F: I/O MENU
```

3 Move down a level and then across to the PARITY command.

v < <

```
PARITY
```

4 Move down to the "parameter" level to select the parity.

v

Use the left/right arrow keys to see the parity choices. Choose from one of the following: None (8 data bits), Even (7 data bits), or Odd (7 data bits). When you set parity, you are indirectly setting the number of data bits.

```
EVEN: 7 BITS
```

5 Save the change and turn off the menu.

Auto/Man
ENTER

The parity selection is stored in *non-volatile* memory, and *does not* change when power has been off or after a remote interface reset.

Remote Interface: you can set or query the parity using the following commands:

```
SYStem:COMMunicate:SERial
   :PARity {EVEN | ODD | NONE}
   :PARity?
```

To Select the Programming Language

You can select one of three language to program the meter from the selected remote interface. The language is SCPI when the meter is shipped from the factory. The language setting is stored in *non-volatile* memory and does not change with power off or reset. *See also "Programming Language Selection"* on page 119.

1 Turn on the front-panel menu.

Shift On/Off <

```
A: MEAS MENU
```

2 Move across to the I/O MENU choice on this level.

< <

```
F: I/O MENU
```

3 Move down a level and then across to the LANGUAGE command.

v <

```
5: LANGUAGE
```

4 Move down to the "parameter" level to select the language.

v

Choose from one of the following: SCPI or KEITHLEY:181

```
SCPI
```

5 Save the change and turn off the menu.

Auto/Man
ENTER

The language selection is stored in *non-volatile* memory, and *does not* change when power has been off or after a remote interface reset.

Remote Interface: you can set the programming language using the following commands:

SYStem:LANGuage {SCPI | 181}
L1X *(SCPI)*
L2X *(Keithley 181)*

Alternate Programming Language Compatibility

You can configure the 34420A to accept and execute the commands of the the Keithley 181 meter. Remote operation will only allow you to access the functionality of the meter language selected. You can take advantage of the full functionality of the 34420A only through the SCPI programming language. For more information on selecting the alternate languages from the front panel menu, see To Select the Programming Language on the previous page.

All of the commands available for the Keithley 181 are implemented in the 34420A, with the exception of the self-test and calibration commands. You must always calibrate the 34420A using the SCPI language setting.

You can use L2X to switch to the Keithley 181 language. L1X switches back to the SCPI language. Note that once the Keithley 181 language has been selected you must append the Keithley "end of message" terminator (X) to any commands.

Be aware that measurement timing may be different in the alternate language compatibility modes.

Specific Issues

You should be aware of the following differences when using the Keithley 181 language:

- Unlike the Keithley 181, the status byte is not cleared when read unless the SRQ bit was set.
- A significant software delay exists between the time a syntax error is generated and the error is reported in the status byte.
- The meter will always use the next highest range than the one requested in a Keithley command. For example, if you specify the Keithley 2 mV range, the meter will use the 10 mV range.
- The digital filter, set to SLOW or MEDium, is used to emulate the Keithley input filter.
- The meter is always set to 2 NPLC.
- The 34420A has a maximum input range of 100 V. Be sure the test setup does not require a 1000 V range.

SCPI Compliance Information

The following commands are device-specific to the 34420A. They are not included in the 1994.0 version of the SCPI standard. However, these commands are designed with the SCPI format in mind and they follow all of the syntax rules of the standard.

Many of the required SCPI commands are accepted by the meter but are not described in this manual for simplicity or clarity. Most of these non-documented commands duplicate the functionality of a command already described in this chapter

```
CALibration
   :COUNt?
   :OUTPut { ZERO | GAIN }
   :OUTPut?
   :SECure:CODE <code string>
   :SECure:STATe { OFF | ON | 0 | 1 } [,<code>]
   :SECure:STATe? returns 0 or 1
   :STRing <string>
   :STRing?

CALCulate
   :FUNCtion { SCALe | AVERage }
   :FUNCtion?
   :AVERage:MINimum?
   :AVERage:MAXimum?
   :AVERage:AVERage?
   :AVERage:SDEViation?
   :AVERage:PTPeak?
   :AVERage:COUNt?
   :SCALe:GAIN
   :SCALe:GAIN?
   :SCALe:OFFSet
   :SCALe:OFFSet?

INPut
   :FILTer[:LPASS]:DIGital:PREcharge {ON | OFF}
   :FILTer[:LPASS]:DIGital:PREcharge?
   :FILTer[:LPASs]:DIGital:RESPonse { SLOW | MEDium | FAST }
   :FILTer[:LPASs]:DIGital:RESPonse?
   :FILTer[:LPASs]:TYPE { ANAlog | DIGital | BOTH }
   :FILTer[:LPASs]:TYPE?
```

```
OUTPut
   :REFerence:OFFSet { <offset> | MINimum | MAXimum }
   :REFerence:OFFSet? [ MINimum | MAXimum ]
   :REFerence:OFFSet:NULL [ONCE]
   :REFerence:SPAN { <span> | MINimum | MAXimum }
   :REFerence:SPAN? [ MINimum | MAXimum ]
   :REFerence:GAIN?
   :REFerence:VALue <value>

SAMPle
   :COUNt { <count> | MINimum | MAXimum }
   :COUNt? [ MINimum | MAXimum ]

[SENSe:]
   CURRent:BIAS:ZERO[:IMMediate]
   NULL [ONCE]
   NULL[:STATe] { OFF | ON | 0 | 1 }
   NULL[:STATe]?
   NULL:VALue { <value> | MINimum | MAXimum }
   NULL:VALue?

[SENSe:]
   FRESistance | RESistance
      :NULL[:STATe] { OFF | ON | 0 | 1 }
      :NULL[:STATe]?
      :NULL:VALue { <value> | MINimum | MAXimum }
      :NULL:VALue?
      :VOLTage:LIMit[:STATe] { OFF | ON | 0 | 1 }
      :VOLTage:LIMit[:STATe]?
      :POWer:LIMit[:STATe] { OFF | ON | 0 | 1 }
      :POWer:LIMit[:STATe]?
      :VOLTage:LIMit:VALue { <value> | MIN | MAX}
      :VOLTage:LIMit:VALue? [ MINimum | MAXimum ]

[SENSe:]
   TEMPerature
      :NULL[:STATe] { OFF | ON | 0 | 1 }
      :NULL[:STATe]?
      :NULL:VALue { <value> | MINimum | MAXimum }
      :NULL:VALue?
      :NPLCycles { <value> | MINimum | MAXimum }
      :NPLCycles? [ MINimum | MAXimum ]
      :TRANsducer:TYPE { DEFault | TCouple | THERmistor | FRTD}
      :TRANsducer:TYPE?
      :TRANsducer:TCouple:TYPE { DEFault | B|E|J|K|N|R|S|T }
```

```
      :TRANsducer:TCouple:TYPE?
      :TRANsducer:TCouple:RJUNction[:VALue] {<value>|MIN| MAX}
      :TRANsducer:TCouple:RJUNction[:VALue]? [MINimum | MAXimum]
      :TRANsducer:TCouple:RJUNction:TYPE {FIXed, INTernal, THERmistor }
      :TRANsducer:TCouple:RJUNction:TYPE?
      :TRANsducer:FRTD:TYPE <type>
      :TRANsducer:FRTD:TYPE?

[SENSe:]
   VOLTage[:DC]
      :NULL[:STATe] { OFF | ON | 0 | 1 }
      :NULL[:STATe]?
      :NULL:VALue { <value> | MINimum | MAXimum }
      :NULL:VALue?

SYSTem
   :INTerface[:TYPE] { GPIB | RS232 }
   :INTerface[:TYPE]?
   :REMote
   :RWLock
   :LOCal

L1X
L2X
```

IEEE-488 Compliance Information

Dedicated Hardware Lines		**Addressed Commands**	
ATN	Attention	DCL	Device Clear
IFC	Interface Clear	EOI	End or Identify
REN	Remote Enable	GET	Group Execute Trigger
SRQ	Service Request Interrupt	GTL	Go to Local
		LLO	Local Lock Out
		SDC	Selected Device Clear
		SPD	Derial Poll disable
		SPE	Serial Poll Enable

IEEE-488.2 Common Commands	
*CLS	*RST
*ESE <enable value>	*SRE <enable value>
*ESE?	*SRE?
*ESR?	*STB?
*IDN?	*TRG
*OPC	*TST
*OPC?	*WAI
*PSC {0 | 1}	
*PSC?	

THIS PAGE HAS BEEN INTENTIONALLY LEFT BLANK.

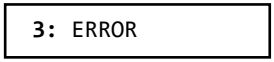
5 Error Messages

Error Messages 222
Execution Errors 223
Self-Test Errors 230
Calibration Errors 232

Error Messages

When the front-panel ERROR annunciator turns on, one or more command syntax or hardware errors have been detected. A record of up to 20 errors is stored in the meter's *error queue*. Errors are retrieved in first-in- first-out (FIFO) order. *See also* *"Error Conditions"* on page 104.

Front-Panel operation: Use the SYStem menu command:



If the ERROR annunciator is on, press Shift > (Menu Recall) to read the errors stored in the queue. The errors are listed *horizontally* on the "parameter" level. The error queue is cleared when power has been off or after a *CLS (clear status) command has been executed.

– To read the error queue from the remote interface:

  SYSTem:ERRor?

  Errors have the following format (the error string may contain up to 80 characters):

  -113, "Undefined header"

Execution Errors

-101 Invalid character

An invalid character was found in the command string. You may have inserted a character such as #, $, or % in the command header or within a parameter. Example: `CONF:VOLT#DC`

-102 Syntax error

Invalid syntax was found in the command string. You may have inserted a blank space before or after a colon in the command header, or before a comma. Example: `SAMP:COUN ,1`

-103 Invalid separator

An invalid separator was found in the command string. You may have used a comma instead of a colon, semicolon, or blank space – or you may have used a blank space instead of a comma. Example: `TRIG:COUN,1` or `CONF:TEMP 1000 0.1`

-104 Data type error

The wrong parameter type was found in the command string. You may have specified a number where a string was expected, or vice versa. Example: `DISP:TEXT 5.0`

-105 GET not allowed

A Group Execute Trigger (GET) is not allowed within a command string.

-108 Parameter not allowed

More parameters were received than expected for the command. You may have entered an extra parameter, or you added a parameter to a command that does not accept a parameter. Example: `READ? 10`

-109 Missing parameter

Fewer parameters were received than expected for the command. You omitted one or more parameters that are required for this command. Example: `SAMP:COUN`

-112 Program mnemonic too long

A command header was received which contained more than the maximum 11 characters allowed. Example: `CONFIGURATION:VOLT:DC`

-113 Undefined header

A command was received that is not valid for this meter. You may have misspelled the command or it may not be a valid command. If you are using the short form of the command, remember that it may contain up to four letters. Example: `TRIGG:COUN 3`

-121 Invalid character in number

An invalid character was found in the number specified for a parameter value. Example: `STAT:QUES:ENAB #B01010102`

-123 Numeric overflow

A numeric parameter was found whose exponent was larger than 32,000. Example: `TRIG:COUN 1E34000`

-124 Too many digits

A numeric parameter was found whose mantissa contained more than 255 digits, excluding leading zeros.

-128 Numeric data not allowed

A numeric parameter was found when a string or discrete parameter was expected.

-131 Invalid suffix

A suffix was incorrectly specified for a numeric parameter. You may have misspelled the suffix. Example: `TRIG:DEL 0.5 SECS`

-134 Suffix too long

A suffix for a numeric parameter contained too many characters.

-138 Suffix not allowed

A suffix was received following a numeric parameter which does not accept a suffix. Example: `SAMP:COUN 1 SEC` (SEC is not a valid suffix).

-141 Invalid character data

-144 Character data too long

-148 Character data not allowed

A discrete parameter was received but a character string or a numeric parameter was expected. Check the list of parameters to verify that you have used a valid parameter type. Example: `DISP:TEXT ON`

-151 Invalid string data

An invalid character string was received. Check to see if you have enclosed the character string in single or double quotes. Example: `DISP:TEXT 'ON` (the ending quote is missing).

-158 String data not allowed

A character string was received but is not allowed for the command. Check the list of parameters to verify that you have used a valid parameter type. Example: `CALC:STAT 'ON'`

-160 to -168 Block data errors

The meter does not accept block data.

-170 to -178 Expression errors

An incorrect expression for the channel specifier was received. The meter does not accept mathematical expressions.

-211 Trigger ignored

A Group Execute Trigger (GET) or `*TRG` was received but the trigger was ignored. Make sure the meter is in the "wait-for-trigger" state before issuing a trigger, and make sure the correct trigger source is selected.

-213 Init ignored

An `INITiate` command was received but could not be executed because a measurement was already in progress. Send a device clear to halt a measurement in progress and place the meter in the "idle" state.

-214 Trigger deadlock

A trigger deadlock occurs when the trigger source is BUS and a `READ?` command is received.

-221 Settings conflict

This error can be generated in one of the following situations:

You sent a `CONFigure` or `MEASure` command with autorange enabled and with a fixed resolution. Example: `CONF:VOLT:DC DEF,0.1`

You turned math on (`CALC:STAT ON`) and then changed to a math operation that was not valid with the present measurement function.

-222 Data out of range

A numeric parameter value is outside the valid range for the command. Example: `TRIG:COUN -3`

-223 Too much data

A character string was received but could not be executed because the string length was more than 11 characters. This error can be generated by the `CALibration:STRing` and `DISPlay:TEXT` commands.

-224 Illegal parameter value

A discrete parameter was received which was not a valid choice for the command. You may have used an invalid parameter choice. Example: `CALC:FUNC SCALE` (SCALE is not a valid choice).

-230 Data corrupt or stale

A `FETCh?` command was received but internal reading memory was empty. The reading retrieved may be invalid.

-330 Self-test failed

The meter's complete self-test failed from the remote interface (`*TST?` command). In addition to this error, more specific self-test errors are also reported. *See also "Self-Test Errors"* on page 230.

-350 Queue overflow

The error queue is full because more than 20 errors have occurred. No additional errors are stored until you remove errors from the queue. The error queue is cleared when power has been off, or after a `*CLS` (clear status) command has been executed.

-410 Query INTERRUPTED

A command was received which sends data to the output buffer, but the output buffer contained data from a previous command (the previous data is not overwritten). The output buffer is cleared when power has been off, or after a `*RST` (reset) command has been executed.

-420 Query UNTERMINATED

The meter was addressed to talk (i.e., to send data over the interface) but a command has not been received which sends data to the output buffer. For example, you may have executed a `CONFigure` command (which does not generate data) and then attempted an `ENTER` statement to read data from the remote interface.

-430 Query DEADLOCKED

A command was received which generates too much data to fit in the output buffer and the input buffer is also full. Command execution continues but all data is lost.

-440 Query UNTERMINATED after indefinite response

The `*IDN?` command must be the last query command within a command string. Example: `*IDN?;:SYST:VERS?`

501 Isolator UART framing error

502 Isolator UART overrun error

511 RS-232 framing error

512 RS-232 overrun error

513 RS-232 parity error

514 Command allowed only with RS-232

There are three commands which are only allowed with the RS-232 interface: `SYSTem:LOCal`, `SYSTem:REMote`, and `SYSTem:RWLock`.

521 Input buffer overflow

522 Output buffer overflow

531 Insufficient memory

There is not enough memory to store the requested number of readings in internal memory using the `INITiate` command. The product of the sample count (`SAMPle:COUNt`) and the trigger count (`TRIGger:COUNt`) must not exceed 1024 readings.

532 Cannot achieve requested resolution

The meter cannot achieve the requested measurement resolution. You may have specified an invalid resolution in the `CONFigure` or `MEASure` command.

540 Cannot use overload as math reference

The meter cannot store an overload reading (9.90000000E+37) as the math reference for null measurements. The math state is turned off as a result of this condition.

550 Command not allowed in local

The meter received a `READ?` command while in the local mode for RS-232 operation. You should always execute the `SYSTem:REMote` command before sending other commands over the interface.

552 Command not allowed in current function

750 **Invalid channel name**

The meter received a `MEAS` or `CONF` command with an invalid channel specifier.

810 **Invalid or unsupported transducer type**

An invalid transducer type was detected as a parameter.

820 **Temperature out of range for specified transducer**

1000 **Settings conflict; voltage limit mode has been disabled**

After being set to the voltage limited resistance mode, the meter received a command that changed the settings to an mode or range not allowed.

1001 **Settings conflict; power limit mode has been disabled**

After being set to the low power resistance mode, the meter received a command that changed the settings to an mode or range not allowed. For example, after the low power mode has been enabled, a `MEAS:RES` (2-wire Ohms) command is received.

Self-Test Errors

The following errors indicate failures that may occur during a self-test. Refer to the *Service Guide* for more information.

601	**Front panel does not respond**
602	**RAM read/write failed**
603	**A/D sync stuck**
604	**A/D slope convergence failed**
605	**Cannot calibrate rundown gain**
606	**Rundown gain out of range**
607	**Rundown too noisy**
608	**Serial configuration readback failed**
609	**DC gain x1 failed**
610	**DC gain x10 failed**
611	**DC gain x100 failed**

613	**Ohms 5 uA source failed**
614	**Ohms 10 uA source failed**
615	**Ohms 100 uA source failed**
616	**Ohms 1 mA source failed**
617	**Ohms 10 mA source failed**
618	**Ohms 20 mV voltage clamp failed**
619	**Ohms 100 mV voltage clamp failed**
620	**Ohms 500 mV voltage clamp failed**
621	**Low Impedance DC gain X100 failed**
622	**High Impedance DC gain X100 failed**
623	**Cannot calibrate precharge**
624	**Unable to sense line frequency**
625	**I/O processor does not respond**
626	**I/O processor failed self-test**
627	**Hardware overload comparator failed**
630	**Bias current DAC test failed**

Calibration Errors

The following errors indicate failures that may occur during a calibration. Refer to the *Service Guide* for more information.

701 **Cal security disabled by jumper**

The calibration security feature has been disabled with a jumper inside the meter. When applicable, this error will occur at power-on to warn you that the meter is unsecured.

702 **Cal secured**

The meter is secured against calibration.

703 **Invalid secure code**

An invalid calibration security code was received when attempting to unsecure or secure the meter. You must use the same security code to unsecure the meter as was used to secure it, and vice versa. The security code may contain up to 11 alphanumeric characters. The first character *must* be a letter.

704 **Secure code too long**

A security code was received which contained more than 11 characters.

705 **Cal aborted**

A calibration in progress is aborted when you press any front-panel key, send a device clear, or change the local/remote state of the meter.

706 **Cal value out of range**

The specified calibration value (`CAL:VALue`) is invalid for the present function and range.

707 **Cal signal measurement out of range**

The specified calibration value (`CAL:VALue`) does not match the signal applied to the meter.

709 **No cal for this function or range**

Attempt to calibrate gain on 1 mV range.

710	**Full scale correction out of range**
720	**Cal DCV offset out of range**
722	**Cal RES offset out of range**
723	**Cal FRES offset out of range**
730	**Precharge DAC convergence failed**
731	**A/D turnover correction out of range**
737	**Bias current selfcal failed**
738	**Charge compensation selfcal failed**
739	**Injected current selfcal failed**
740	**Cal checksum failed, secure state**
741	**Cal checksum failed, string data**
742	**Cal checksum failed, DCV corrections**
743	**Cal checksum failed, Low Power FRES corrections**
744	**Cal checksum failed, RES corrections**
745	**Cal checksum failed, FRES corrections**
746	**Cal checksum failed, Low Voltage FRES corrections**
747	**Cal checksum failed, GPIB address**
748	**Cal checksum failed, internal data**

THIS PAGE HAS BEEN INTENTIONALLY LEFT BLANK.

6 Application Programs

Application Programs 236
HP BASIC Language Programs 237
RS-232 Operations with QuickBASIC 245
Microsoft Visual Basic Language Program 247
C Language Programs 254
Microsoft Excel Macro Example 266

Application Programs

This chapter contains several remote interface application programs to help you develop programs for your measurement application. Chapter 4, "Remote Interface Reference" starting on page 129, lists the syntax for the SCPI (Standard Commands for Programmable Instruments) commands available to program the meter.

This chapter is divided into the following sections:

- "HP BASIC Language Programs" on page 237
- "RS-232 Operations with QuickBASIC" on page 245
- "Microsoft Visual Basic Language Program" on page 247
- "C Language Programs" on page 254
- "Microsoft Excel Macro Example" on page 266

HP BASIC Language Programs

All of the HP BASIC examples in this chapter were developed and tested on an HP 9000 Series 300 controller. Each device on the GPIB (IEEE-488) interface must have a unique address. You can set the meter's address to any value between 0 and 30. The current address is momentarily displayed on the front panel when you turn on the meter.

The GPIB (IEEE-488) address is set to "22" when the meter is shipped from the factory. The examples in this chapter assume a GPIB address of 22. When sending a remote interface command, you append this address to the GPIB interface's select code (normally 7). Therefore, with an address of 22 and a select code of 7, the combination is "722".

HP BASIC/GPIB Program Example 1

This example demonstrates the general sequence of programming and the use of the MEASure? command to set and obtain a 4-wire Ohms measurement.

```
10  !
20  ! This program configures the meter for a 4-wire ohms measurement on
30  ! Channel 1 using the MEAS command.
40  !
50    REAL Value,Rdg         ! Dimension real variables
60    CLEAR 7                ! Clear interface - send "device clear"
70    ASSIGN @Meter TO 722   ! Assign I/O path to address 722
80  !
90    OUTPUT @Meter;"*RST"   ! Reset the meter
100   OUTPUT @Meter;"*OPC?"  ! Send "1" to output buffer when reset complete
110   ENTER @Meter;Value     ! Enter value
120 !
130 ! Select 4-wire ohms on the 10 ohm range with the least resolution;
140 !
150   OUTPUT @Meter;"MEAS:FRES? 10,MAX"
160   ENTER @Meter;Rdg       ! Enter reading
170   PRINT Rdg              ! Print reading
180   END
```

HP BASIC/GPIB Program Example 2

This program example demonstrates how two input channels can be independently configured and used.

```
10  !
20  ! This program configures the meter for voltage measurements on two
30  ! input channels. The channels are set to different ranges and
40  ! resolutions to show the independence of the settings for each channel.
50  !
60    REAL Rdg_ch1,Rdg_ch2,Null_ch1,Null_ch2  ! Dimension variables
70    CLEAR 7                   ! Clear interface - send "device clear"
80    ASSIGN @Meter TO 722      ! Assign I/O path to address 722
90    OUTPUT @Meter;"*RST"      ! Reset the meter
100   OUTPUT @Meter;"*OPC?"     ! Send "1" to output buffer when complete
110   ENTER @Meter;Value        ! Enter value
120 !
130 ! Configure channel 1 for dc volts, 1 volt range with 10 mV resolution
140 ! Configure channel 2 for dc volts, 1 mV range with least resolution
150   OUTPUT @Meter;"CONF:VOLT:DC 1,0.01,(@FRON1);:CONF:VOLT:DC 0.001,MAX,(@FRON2)"
160 !
170   OUTPUT @Meter;"ROUT:TERM FRON1"         ! Select channel 1
180   OUTPUT @Meter;"READ?"                   ! Take reading; send to output buffer
190   ENTER @Meter;Rdg_ch1                    ! Enter reading
200 !
210   OUTPUT @Meter;"SENS1:VOLT:DC:NULL ON"              ! Enable null on channel 1
220   OUTPUT @Meter;"SENS1:VOLT:DC:NULL:VAL ";Rdg_ch1    ! Store value in register
230 !
240   OUTPUT @Meter;"READ?"     ! Take nulled reading; send to output buffer
250   ENTER @Meter;Null_ch1     ! Enter nulled reading
```

... continued

```
260   PRINT ;Null_ch1                      ! Print nulled reading
270 !
280   OUTPUT @Meter;"ROUT:TERM FRON2"      ! Select channel 2
290   OUTPUT @Meter;"READ?"                ! Take reading; send to output buffer
300   ENTER @Meter;Rdg_ch2                 ! Enter reading
310 !
320   OUTPUT @Meter;"SENS2:VOLT:DC:NULL ON"              ! Enable null on channel 2
330   OUTPUT @Meter;"SENS2:VOLT:DC:NULL:VAL ";Rdg_ch2    ! Store value in register
340 !
350   OUTPUT @Meter;"READ?"                ! Take nulled reading; send to output buffer
360   ENTER @Meter;Null_ch2                ! Enter nulled reading
370   PRINT Null_ch2                       ! Print nulled reading
380 !
390   END
```

HP BASIC/GPIB Program Example 3

This program example demonstrates the use of an SRQ to indicate when measurements are complete. The meter is set to use external triggering. Math scaling is also demonstrated.

```
10  !
20  ! This program configures the meter to make multiple dc voltage measurements
30  ! using an external trigger. The "Operation Complete" bit in the Standard
40  ! Event Register is used to determine when the trigger is received and
50  ! generates an SRQ. This program also shows the use of math scaling.
60  !
70    INTEGER Value,Task        ! Dimension variables
80    CLEAR 7                   ! Clear interface - send "device clear"
90    ASSIGN @Meter TO 722      ! Assign I/O path to address 722
100   COM @Meter                ! Use same address in subprogram
110   OUTPUT @Meter;"*RST"      ! Reset the meter
120   OUTPUT @Meter;"*CLS"      ! Clear status registers
130   OUTPUT @Meter;"*OPC?"     ! Send "1" to output buffer when reset complete
140   ENTER @Meter;Value        ! Enter value
150 !
160 ! Configure the Status Register system to generate an SRQ when the
170 ! "Operation Complete" bit is set in the Standard Event Register
180 !
190   OUTPUT @Meter;"*SRE 32"   ! Enable "Standard Event" bit in Status Byte
200                             ! to pull the IEEE-488 SRQ line
210   OUTPUT @Meter;"*ESE 1"    ! Enable "Operation Complete" to set
220                             ! "Standard Event" bit in Status Byte
```

... continued

```
230 !
240 ! Configure the GPIB interrupt system to generate an SRQ
250 !
260   Gpib=7                              ! GPIB select code is "7"
270   ON INTR Gpib CALL Read_data         ! Call subprogram when operation complete
280   Mask=2                              ! Bit 1 is SRQ
290   ENABLE INTR Gpib;Mask               ! Enable SRQ to interrupt program
300 !
310  OUTPUT @Meter;"SENS:FUNC 'VOLT:DC'"  ! Select dc volts on channel 1 (default)
320  OUTPUT @Meter;"SENS:VOLT:DC:RANG 10" ! Select 10 volt range
330  OUTPUT @Meter;"SENS:VOLT:DC:NPLC 10" ! Select 10 power line cycles
340  OUTPUT @Meter;"CALC:FUNC SCAL"       ! Select math scaling function
350  OUTPUT @Meter;"CALC:STAT ON"         ! Turn math scaling on
360  OUTPUT @Meter;"CALC:SCAL:GAIN 0.001" ! Set the gain value (1 mV)
370  OUTPUT @Meter;"CALC:SCAL:OFFS 0.5"   ! Set the offset value (50 mV)
380 !
390  OUTPUT @Meter;"SAMP:COUN 5"          ! Take 5 reading per trigger
400  OUTPUT @Meter;"TRIG:SOUR EXT"        ! Select external trigger source
410  OUTPUT @Meter;"INIT"                 ! Initiate measurement, wait for trigger,
420                                       ! and send readings to internal memory
430  OUTPUT @Meter;"*OPC"                 ! Set "Operation Complete" bit in
440                                       ! Standard Event Register when complete
```

... continued

```
450 !
460 ! Wait for "Operation Complete"
470 !
480  Task=1
490  WHILE Task=1
500    DISP "Reading..."
510    WAIT .5
520    DISP " "
530    WAIT .5
540  END WHILE
550 !
560  OFF INTR Gpib              ! Disable interrupts
570  END
580 !
590 ! After SRQ is generated, transfer the readings to the output buffer and print
600 !
610  SUB Read_data              ! Subprogram is called after INIT complete and trigger
                                  is received
620    DIM Rdgs(1:5)            ! Dimension array for readings
630    COM @Meter               ! Use same address as in main program
640    OUTPUT @Meter;"FETC?"    ! Transfer readings to output buffer
650    ENTER @Meter;Rdgs(*)     ! Enter readings
660    PRINT Rdgs(*)            ! Print readings
670    STOP
680  SUBEND
```

HP BASIC/GPIB Program Example 4

This program example demonstrates how data can be formatted and put in a file. The program also shows how to use math operations (stats) to average data.

```
10  ! RE-SAVE "EXAMP"
20  CONFIGURE SAVE ASCII OFF
30  ! This program will generate a DOS file, in the file will be three columns each
40  ! separated by the Separator$ specified below, The columns are:
50  ! Reading Number,Time,DMM Reading
60  ! This file may be imported into a spreadsheet for analysis
70  !
80  Separator$=","        !What separates the columns
90  Num_points=25         !How many measurements (rows in file)
100 Address=722        !34420A GPIB address
110 Range=1.E-3        !34420A Range
120 Nplc=1             !34420A Integration time
130 Num_samples=20     !How many samples per measurement
140 ALLOCATE Results$(1:Num_points)[80],Filename$[80]
150 Filename$="B:/EXAMP_DAT"
160 Setup_meter(Address,Range,Nplc,Num_samples)
170 FOR I=1 TO Num_points
180   DISP "Getting "&VAL$(I)&" of "&VAL$(Num_points)&" measurements"
190   OUTPUT Results$(I) USING
      "4D,A,8A,A,SD.12D";I,Separator$,TIME$(TIMEDATE),Separator$,FNGetdata(Address)
200 NEXT I
210 ON ERROR GOTO Save_file
220 CREATE Filename$,1
230 Save_file: ASSIGN @File TO Filename$
240 OUTPUT @File;Results$(*)
250 ASSIGN @File TO *
260 DISP "Done"
270 END
```

... continued

```
280 !************************************************************************
290 Getdata: DEF FNGetdata(Address)
300   REPEAT
310   UNTIL BIT(SPOLL(Address),5)          !Bit 5 is Standard Event Reg
320   OUTPUT Address;"*CLS"                !Clear status
330   OUTPUT Address;"CALC:AVER:AVER?"     !Get average
340   ENTER Address;Result
350   OUTPUT Address;"CALC:FUNC AVER"      !Resets average to 0
360   OUTPUT Address;"*OPC"
370   OUTPUT Address;"INIT"                !Start taking next readings
380   RETURN Result
390 FNEND
400 !************************************************************************
410 Setup_meter: SUB Setup_meter(Address,Range,Nplc,Num_samples)
420 OUTPUT Address;"*CLS"                                  !Clear Status
430 OUTPUT Address;"SENSE:VOLTAGE:RANGE "&VAL$(Range)      !
440 OUTPUT Address;"INPUT:FILTER:STATE OFF"                !
450 OUTPUT Address;"SENSE:VOLTAGE:DC:NPLC "&VAL$(Nplc)     !
460 OUTPUT Address;"CALC:FUNC AVER"                        !Take average of
470 OUTPUT Address;"CALC:STATE ON"                         !Num_samples
480 OUTPUT Address;"SAMPLE:COUNT "&VAL$(Num_samples)
490 OUTPUT Address;"TRIGGER:SOURCE IMM"
500 OUTPUT Address;"*ESE 1"                                !Enable register mask
510 OUTPUT Address;"*OPC"                                  !for *OPC
520 OUTPUT Address;"INIT"
530 SUBEND
540 !************************************************************************
```

RS-232 Operations with QuickBASIC

This program example uses QuickBASIC and the 82335B GPIB Interface Card and command library for IBM® PC compatibles.

```
CLS
LOCATE 1, 1
DIM cmd$(100), resp$(1000)
’ Set up serial port for 9600 baud, even parity, 7 bits;
’ Ignore Request to Send Carrier Detect; Send line feed;
’ enable parity check, reserve 1000 bytes for input buffer
’
OPEN "com1:9600,e,7,2,rs,cd,lf,pe" FOR RANDOM AS #1 LEN = 1000
’
’ Put the meter into remote operation mode
PRINT #1, "SYST:REM"
’
’ Query the meter’s id string
’ PRINT #1, "*IDN?"
LINE INPUT #1, resp$
PRINT "*IDN? returned: ", resp$
’
’ Get the SCPI revision number
PRINT #1, "SYST:VERS?"
LINE INPUT #1, resp$
PRINT "SYST:VERS returned: ", resp$
’
’ Send a message to the meter’s display
PRINT #1, "DISP:TEXT ’Keysight 34420A’ "
’
’ Configure the meter for dc voltage measurements, 10 V range,
’ 0.1 V resolution, take 4 readings
PRINT #1, "CONF:VOLT:DC 10, 0.1, ;:SAMP:COUN 4"
’ Trigger the readings and get the results
PRINT #1, "READ?"
```

... continued

```
LINE INPUT #1, resp$
PRINT "READ? returned: ", resp$
END
```

Microsoft Visual Basic Language Program

This example was created in Microsoft® Visual Basic for Windows™. The example uses the 82335B and GPIB control library. You must set up the GLOBAL.BAS to provide access to the GPIB.DLL. The GLOBAL.BAS file is included with the Keysight control libraries. In addition to the global declarations, the subroutine check_srq is also included in the GLOBAL.BAS file.

The GPIB (IEEE-488) address is set to “22” when the meter is shipped from the factory. The examples in this chapter assume a GPIB address of 22. When sending a remote interface command, you append this address to the GPIB interface’s select code (normally 7). Therefore, with an address of 22 and a select code of 7, the combination is “722”.

```
Sub Command1_Click ()
Rem Set up interface variables
    isc& = 7
    device& = isc& * 100 + 22
    swap% = 2
    max% = 4000 * swap%
    act% = 0
    max1% = 50
    TimeVal# = 10#
Rem Clear out text boxes
    Text1.Text = ""
    Text2.Text = ""
    Text3.Text = ""
    Text4.Text = ""
Rem Set up the meter
    errnum% = GpibOpen(isc&, hGpib%)
    Call check_srq
    errnum% = GpibReset(hGpib%, isc&)
    Call check_srq
    errnum% = GpibTimeout(hGpib%, isc&, TimeVal#)
    Call check_srq
    rst$ = "*RST; *CLS; *SRE 32; *ESE 60"
    errnum% = GpibOutputS(hGpib%, device&, rst$, Len(rst$))
    Call check_srq
```

... continued

```
Rem Channel 1 measurement
    conf1$ = "CONF:VOLT:DC 1,0.01,(@FRON1)"
    errnum% = GpibOutputS(hGpib%, device&, conf1$, Len(conf1$))
    Call check_srq
    conf2$ = "CONF:VOLT:DC 0.001,MAX,(@FRON2); "
    errnum% = GpibOutputS(hGpib%, device&, conf2$, Len(conf2$))
    Call check_srq
    route$ = "ROUT:TERM FRONT1"
    errnum% = GpibOutputS(hGpib%, device&, route$, Len(route$))
    ’ Take the reading and check for errors.
    reed$ = "READ?"
    errnum% = GpibOutputS(hGpib%, device&, reed$, Len(reed$))
    errnum% = GpibEnter(hGpib%, device&, reading1!)
    Text1.Text = Str$(reading1!)
    Call check_srq
Rem Channel 2 measurement
    route$ = "ROUT:TERM FRONT2"
    errnum% = GpibOutputS(hGpib%, device&, route$, Len(route$))
    Call check_srq
’ Take the reading and check for errors.
    errnum% = GpibOutputS(hGpib%, device&, reed$, Len(reed$))
    errnum% = GpibEnter(hGpib%, device&, reading2!)
    Text2.Text = Str$(reading2!)
    Call check_srq
Rem Nulled Channel 1 Measurement
    null_stmt$ = "SENS1:VOLT:DC:NULL ON"
    errnum% = GpibOutputS(hGpib%, device&, null_stmt$, Len(null_stmt$))
    Call check_srq
    null_stmt$ = "SENS1:VOLT:DC:NULL " + Str$(reading1!)
    errnum% = GpibOutputS(hGpib%, device&, null_stmt$, Len(null_stmt$))
    Call check_srq
    errnum% = GpibOutputS(hGpib%, device&, reed$, Len(reed$))
    errnum% = GpibEnter(hGpib%, device&, reading!)
    Text3.Text = Str$(reading!)
    Call check_srq
```

... continued

```
Rem Nulled Channel 2 Measurement
    null_stmt$ = "SENS2:VOLT:DC:NULL ON"
    errnum% = GpibOutputS(hGpib%, device&, null_stmt$, Len(null_stmt$))
    Call check_srq
    null_stmt$ = "SENS2:VOLT:DC:NULL " + Str$(reading2!)
    errnum% = GpibOutputS(hGpib%, device&, null_stmt$, Len(null_stmt$))
    Call check_srq
    errnum% = GpibOutputS(hGpib%, device&, reed$, Len(reed$))
    errnum% = GpibEnter(hGpib%, device&, reading!)
    Text4.Text = Str$(reading!)
    Call check_srq
Rem Close the instrument
    errnum% = GpibClose(hGpib%)
End Sub
Sub Command2_Click ()
    End
End Sub
Sub Command3_Click ()
Rem Set up interface variables
    isc& = 7
    device& = isc& * 100 + 22
    swap% = 2
    max% = 4000 * swap%
    act% = 0
    max1% = 50
    TimeVal# = 10#
Rem Set up Measurement variables
    Funct$ = "SCAL"
    Trig_src$ = "EXT"
    range% = 10
    nplc% = 100
    Gain% = .001
    Offset% = .5
    samp_count% = 100
    ReDim Readings!(samp_count%)
Rem Clear out text boxes
    Text1.Text = ""
    Text2.Text = ""
    Text3.Text = ""
    Text4.Text = ""
```

... continued

```
Rem Set up the meter
    errnum% = GpibOpen(isc&, hGpib%)
    Call check_srq
    errnum% = GpibReset(hGpib%, isc&)
    Call check_srq
    errnum% = GpibTimeout(hGpib%, isc&, TimeVal#)
    Call check_srq
    rst$ = "*RST; *CLS; *SRE 32; *ESE 61"
    errnum% = GpibOutputS(hGpib%, device&, rst$, Len(rst$))
    Call check_srq
Rem Measurement Setup
    quote$ = Chr$(34)
    cmd$ = "SENS:FUNC " + quote$ + "VOLT:DC" + quote$
    errnum% = GpibOutputS(hGpib%, device&, cmd$, Len(cmd$))
    Call check_srq
    cmd$ = "SENS:VOLT:DC:RANG " + Str$(range%)
    errnum% = GpibOutputS(hGpib%, device&, cmd$, Len(cmd$))
    Call check_srq
    cmd$ = "SENS:VOLT:DC:NPLC " + Str$(nplc%)
    errnum% = GpibOutputS(hGpib%, device&, cmd$, Len(cmd$))
    Call check_srq
    cmd$ = "CALC:FUNC " + Funct$
    errnum% = GpibOutputS(hGpib%, device&, cmd$, Len(cmd$))
    Call check_srq
    cmd$ = "CALC:STAT ON"
    errnum% = GpibOutputS(hGpib%, device&, cmd$, Len(cmd$))
    Call check_srq
    cmd$ = "CALC:SCAL:GAIN " + Str$(Gain%)
    errnum% = GpibOutputS(hGpib%, device&, cmd$, Len(cmd$))
    Call check_srq
    cmd$ = "CALC:SCAL:OFFS " + Str$(Offset%)
    errnum% = GpibOutputS(hGpib%, device&, cmd$, Len(cmd$))
    Call check_srq
    cmd$ = "SAMP:COUN " + Str$(samp_count%)
    errnum% = GpibOutputS(hGpib%, device&, cmd$, Len(cmd$))
    Call check_srq
    cmd$ = "TRIG:SOUR " + Trig_src$
    errnum% = GpibOutputS(hGpib%, device&, cmd$, Len(cmd$))
    Call check_srq
```

... continued

```
    Rem Need to set timeout so that sampling and averaging can complete.
    Rem This assumes triggers occur at a rate high enough to be a
    Rem negligible factor.
    errnum% = GpibTimeout(hGpib%, isc&, samp_count% * nplc% / 20)
    Call check_srq
    Rem Use "INIT" and "FETCH" to initiate measurement and read back
    Rem results into an array.
    cmd$ = "INIT"
    errnum% = GpibOutputS(hGpib%, device&, cmd$, Len(cmd$))
    Call check_srq
    reed$ = "FETC?"
    errnum% = GpibOutputS(hGpib%, device&, reed$, Len(reed$))
    errnum% = GpibEnterA(hGpib%, device&, Readings!(0), samp_count%)
    Text1.Text = Str$(Readings!(0))
    Call check_srq
Rem Close the instrument
    errnum% = GpibClose(hGpib%)
End Sub
Sub Command5_Click ()
    form1.PrintForm
End Sub
```

... continued The following subroutine may be included in the GLOBAL.BAS file.

```
’ GPIB.DLL Access
’ Put into your GLOBAL.BAS file
    .
    .
    .
Global isc As Long
Global device As Long
Global hGpib As Integer
Global errnum As Integer

Sub check_srq ()

    condition = 1

     ’ Display any errors caused by the return of an GPIB command.

     If errnum% <> NOERROR Then
         MsgBox ("Error = " + GpibErrStr$(errnum%))
     End If

     ’ Wait for the completion of the current command.

     errnum% = GpibOutPutS(hGpib%, device&, "*OPC?", 5)
     errnum% = GpibEnter(hGpib%, device&, done!)

     ’ If there is an error bit set, obtain the error.

     errnum% = GpibSpoll(hGpib%, device&, response%)

     If response% And 32 Then
         syst_err$ = "SYST:ERR?"
         errnum% = GpibOutPutS(hGpib%, device&, syst_err$, Len(syst_err$))
         errnum% = GpibEnter(hGpib%, device&, errx!)
         actual% = 80
         MsgBox ("Error=" + Str$(errx!))
         errnum% = GpibOutPutS(hGpib%, device&, "*CLS", 4)
     End If

     ’ Obtain other GPIB errors.

     errnum% = GpibStatus(hGpib%, isc&, condition, Status%)

     If errnum% <> NOERROR Then
         MsgBox ("Error = " + GpibErrStr$(errnum%))
     End If

     If Status% = 1 Then
         errnum% = GpibSpoll(hGpib%, device&, Status%)
         If errnum% <> NOERROR Then
              MsgBox ("Error = " + GpibErrStr$(errnum%))
         End If
     End If

End Sub
```

... continued The following function is used by the check_srq subroutine.

```
Function GpibErrStr$ (errorcode As Integer)
Select Case errorcode
    Case NOERR
         Msg$ = " No error "
    Case EUNKNOWN
         Msg$ = " Unknown error "
    Case ESEL
         Msg$ = " Invalid select code or device address "
    Case ERANGE
         Msg$ = " Value out of range "
    Case ETIME
         Msg$ = " Timeout "
    Case ECTRL
         Msg$ = " GPIB must be controller "
    Case EPASS
         Msg$ = " Pass control not permitted "
    Case EFILE
         Msg$ = " File I/O Error "
    Case ENUMB
         Msg$ = " Invalid Number "
    Case EADDR
         Msg$ = " Improper addressing "
    Case EOPEN
         Msg$ = " Cannot Open This Card "
    Case ENOOPEN
         Msg$ = " Card is not open "
    Case ECLOSE
         Msg$ = " Cannot close card "
    Case EHANDLE
         Msg$ = " Invalid handle for this ISC "
    Case Else
         Msg$ = " What? "
End Select
GpibErrStr$ = Msg$
End Function
```

C Language Programs

All of the C language example programs in this chapter are written for the 82335 GPIB Interface Card using the GPIB Command Library for C. Unless otherwise noted, the library functions used in the example programs are compatible with the ANSI C standard.

All of the C programs were compiled and tested used the following compilers:

- Microsoft® QuickC® Version 2.0
- Borland® Turbo C®++ Version 1.0

Compiling and Linking a C Program

To run a program, first compile and link the program to make an executable file. To compile and link a program:

- Make sure the necessary PATH statements have been added to your AUTOEXEC.BAT file to enable the compiler to find the library and header files (see your C language manual to specify the proper paths).
- Link the appropriate GPIB C library file for your compiler (the file is located on the GPIB Command Library disk that came with the GPIB Interface Card). Use the appropriate library file:

```
clgpib.lib      for Microsoft® QuickC®
tchgpib.lib     for Borland® Turbo C®++
```

- If you are not compiling in the “large” or “huge” memory model, include the “CFUNC.H” header file in your program (the file is located on the GPIB Command Library disk that came with your GPIB Interface Card).

– To compile Microsoft® QuickC® programs from the DOS command line (using the “large” memory model), execute the following:

```
qcl /AL <path...\program.C> <path... \clgpib.lib>
```

For example,

```
qcl /AL b:\meter.c c:\qc2\lib\clgpib.lib
```

Change the “/AL” parameter to the appropriate setting when compiling in the smaller memory models (see your C Language manual for more information). For some programs, you may also have to change the stack size using the “/F” parameter.

– To compile Borland® Turbo C®++ programs from the DOS command line (using the “large” memory model), execute the following:

```
tcc -ml <path...\program.C> <path... \tchgpib.lib>
```

For example

```
tcc -ml b:\meter.c c:\tc\lib\tchgpib.lib
```

Change the “-ml” parameter to the appropriate setting when compiling in smaller memory models (see you C Language manual for more information).

– Once compiled and linked, and executable file (.EXE) and object file (.OBJ) are created in the current directory. run the program by typing the file name with the .EXE extension.

C/GPIB Example 1

This example demonstrates the general sequence of programming and the use of the MEASure? command to set and obtain a 4-wire Ohms measurement.

```
/**********************************************************************
    This program configures the meter for a 4-wire ohms measurement on
    Channel 1 using the MEAS command.
**********************************************************************/
#include <stdio.h>     /* Used for printf() */
#include <stdlib.h>    /* Used for atoi() */
#include <string.h>    /* Used for strlen() */
#include <cfunc.h>     /* Header file from GPIB Command Library */

#define ADDR 722L      /* Set GPIB address for meter */

/* Function Prototypes */

void rst_clear(void);
void meter_meas(void);
void check_error(char *func_name);

/**********************************************************************/

void main(void)        /* Start of main() */
{
    rst_clear();       /* Reset the instrument and clear error queue */
    meter_meas();      /* Set up meter for measurement */
}
/**********************************************************************/

void rst_clear(void)
{
    /* Reset the meter, clear the error queue, and wait for commands to
       complete. A "1" is sent to the output buffer from the *OPC? command
       when *RST and *CLS are completed. */

    float value;

    IOOUTPUTS(ADDR, "*RST;*CLS;*OPC?", 15);
    IOENTER(ADDR, &value);
}
```

... continued

```
/*************************************************************************/
void meter_meas(void)
{
    /* Configure the meter for 4-wire ohms measurements on channel 1
       (default channel). The measurement is made on the 10 ohm range with
       the least resolution. */

    float ohm_rdg;

    IOOUTPUTS(ADDR, "MEAS:FRES? 10, MAX", 18);   /* Configure channel 1 */
    IOENTER(ADDR, &ohm_rdg);                     /* Enter reading */

    printf("Reading: %f\n\n", ohm_rdg);          /* Display reading */

    /* Call the function to check for errors */

    check_error("meter_meas");

}
/*************************************************************************/
void check_error(char *func_name)
{
    /* Read error queue to determine if errors have occurred */

    char message[80];
    int length = 80;

    IOOUTPUTS(ADDR, "SYST:ERR?", 9);       /* Read the error queue */
    IOENTERS(ADDR, message, &length);      /* Enter error string */

    while (atoi(message) != 0)             /* Loop until all errors are read */
    {
         printf("Error %s in function %s\n\n", message, func_name);
         IOOUTPUTS(ADDR, "SYST:ERR?", 9);
         IOENTERS(ADDR, message, &length);
    }
}
```

C/GPIB Example 2

This program example demonstrates two input channels can be independently configured and used.

```
/**************************************************************************
    This program configures the meter for voltage measurements on two
    input channels. The channels are set to different ranges and resolutions
    to show the independence of the settings for each channel.
**************************************************************************/

#include <stdio.h>        /* Used for printf() */
#include <stdlib.h>       /* Used for atoi() */
#include <string.h>       /* Used for strlen() */
#include <cfunc.h>        /* Header file from GPIB Command Library */

#define ISC 7L            /* Assign GPIB select code */
#define ADDR 722L         /* Set GPIB address for meter */

/* Function Prototypes */

void rst_clear(void);
void meter_meas(void);
void command_exe(char *commands[], int length);
void check_error(char *func_name);

/**************************************************************************/

void main(void)           /* Start of main() */
{
    rst_clear();          /* Reset the instrument and clear error queue */
    meter_meas();         /* Set up meter for measurement */
}

/**************************************************************************/

void rst_clear(void)
{
    /* Reset the meter, clear the error queue, and wait for commands to
       complete. A "1" is sent to the output buffer from the *OPC? command
       when *RST and *CLS are completed. */

    float value;

    IOOUTPUTS(ADDR, "*RST;*CLS;*OPC?", 15);
    IOENTER(ADDR, &value);
}
```

... continued

```
/*************************************************************************/
void meter_meas(void)
{
    /* Configure channel 1 for dc voltage measurements on the 1 volt range
       with 10 mV resolution. Configure channel 2 for dc voltage
       measurements on the 1 mV range with the least resolution. */

    float rdg_ch1, rdg_ch2, null_ch1, null_ch2;

    static char state[2] = {13,10};     /* ASCII 13 = carriage return,
                                           ASCII 10 = line feed */

    /* Configure measurements on both channels */

    IOOUTPUTS(ADDR, "CONF:VOLT:DC 1,0.01,(@FRON1);:CONFVOLT:DC 0.001,MAX,(@FRON2)",
61);
    /* Channel 1: initiate measurement, transfer reading to the output buffer */
    IOOUTPUTS(ADDR, "ROUT:TERM FRON1", 15);
    IOOUTPUTS(ADDR, "READ?", 5);
    IOENTER(ADDR, &rdg_ch1);

    /* Channel 1: Turn null on */

    IOOUTPUTS(ADDR, "SENS1:VOLT:DC:NULL ON", 21);

    /* Place the null reading into the Null Register. To do this,
       suppress the carriage return (CR) and line feed (LF) before
       sending the null reading. */

    /* First, disable EOI (End-or-Identify) and EOL (End-of-Line) */

    IOEOI(ISC, 0);IOEOL(ISC, " ", 0);

    /* Send "SENS" header and suppress CR/LF */

    IOOUTPUTS(ADDR, "SENS1:VOLT:DC:NULL:VAL ", 23);

    /* Re-enable EOI and EOL for normal GPIB operation and send data */

    IOEOI(ISC, 1);IOEOL(ISC, state, 2);
    IOOUTPUT(ADDR, rdg_ch1);

    /* Channel 2: Initiate measurement, transfer reading to output buffer,
      and store reading in variable */

    IOOUTPUTS(ADDR, "ROUT:TERM FRON2", 15);
    IOOUTPUTS(ADDR, "READ?", 5);
    IOENTER(ADDR, &rdg_ch2);
```

... continued

```
    /* Channel 2: Turn null on */
    IOOUTPUTS(ADDR, "SENS2:VOLT:DC:NULL ON", 21);
    /* Place the null reading into the Null Register. To do this,
       suppress the carriage return (CR) and line feed (LF) before
       sending the null reading. */
    /* First, disable EOI (End-or-Identify) and EOL (End-of-Line) */
    IOEOI(ISC, 0);IOEOL(ISC, " ", 0);
    /* Send "SENS" header and suppress CR/LF */
    IOOUTPUTS(ADDR, "SENS2:VOLT:DC:NULL:VAL ", 23);
    /* Re-enable EOI and EOL for normal GPIB operation and send data */
    IOEOI(ISC,1);IOEOL(ISC, state, 2);
    IOOUTPUT(ADDR, rdg_ch2);
    /* Channel 1: Take nulled reading, transfer reading to output buffer,
       and print nulled reading */
    IOOUTPUTS(ADDR, "ROUT:TERM FRON1", 15);
    IOOUTPUTS(ADDR, "READ?", 5);
    IOENTER(ADDR, &null_ch1);
    printf("Nulled Reading (Chan 1): %f\n", null_ch1);
    /* Channel 2: Take nulled reading, transfer reading to output buffer,
       and print nulled reading */
    IOOUTPUTS(ADDR, "ROUT:TERM FRON2", 15);
    IOOUTPUTS(ADDR, "READ?", 5);
    IOENTER(ADDR, &null_ch2);
    printf("Nulled Reading (Chan 2): %f\n", null_ch2);
    /* Call the function to check for errors */
    check_error("meter_meas");
}
```

... continued

```
/*************************************************************************/
void command_exe(char *commands[], int length)
{
    /* Execute one command string at a time using a loop */
    int loop;
    for (loop = 0; loop length; loop++)
    {
         IOOUTPUTS(ADDR, commands[loop], strlen(commands[loop]));
    }
}
/*************************************************************************/
void check_error(char *func_name)
{
    /* Read error queue to determine if errors have occurred */
    char message[80];
    int length = 80;
    IOOUTPUTS(ADDR, "SYST:ERR?", 9);       /* Read the error queue */
    IOENTERS(ADDR, message, &length);      /* Enter error string */
    while (atoi(message) != 0)             /* Loop until all errors are read */
    {
         printf("Error %s in function %s\n\n", message, func_name);
         IOOUTPUTS(ADDR, "SYST:ERR?", 9);
         IOENTERS(ADDR, message, &length);
    }
}
/*************************************************************************/
```

C/GPIB Program Example 3

This program example demonstrates the use of an SRQ to indicate when measurements are complete. The meter is set to use external triggering. Math scaling is also demonstrated.

```
/**************************************************************************
    This program configures the meter to make multiple dc voltage
    measurements using an external trigger. The "Operation Complete" bit
    in the Standard Event Register is used to determine when the trigger
    is received and generates an SRQ. This program also shows the use of
    math scaling.
**************************************************************************/

#include <stdio.h>           /* Used for printf() */
#include <stdlib.h>          /* Used for atoi() */
#include <string.h>          /* Used for strlen() */
#include <cfunc.h>           /* Header file from GPIB Command Library */

#define ISC 7L               /* Assign GPIB select code */
#define ADDR 722L         /* Set GPIB address for meter */

/* Function Prototypes */

void rst_clear(void);
void setup_status(void);
void meter_meas(void);
void command_exe(char *commands[], int length);
void check_error(char *func_name);

/**************************************************************************/

void main(void)              /* Start of main() */
{
    rst_clear();             /* Reset the instrument and clear error queue */
    setup_status();          /* Set up 34420A status registers */
    meter_meas();               /* Set up meter for measurement */
}
```

... continued

```
/*************************************************************************/
void rst_clear(void)
{
    /* Reset the meter, clear the error queue, and wait for commands to
       complete. A "1" is sent to the output buffer from the *OPC? command
       when *RST and *CLS are completed. */

    float value;

    IOOUTPUTS(ADDR, "*RST;*CLS;*OPC?", 15);
    IOENTER(ADDR, &value);
}
/*************************************************************************/
void setup_status(void)
{
    /* An interrupt is to be generated when the external trigger is received
       and readings are available in internal memory. To enable the status
       registers to generate an interrupt, you must enable bits in the
       Status Byte and Standard Event Register. */

    /* Enable bit 5 "Standard Event" in the Status Byte to recognize
       activity from the Standard Event Register (2^5 = 32) */

    IOOUTPUTS(ADDR, "*SRE 32", 7);

    /* Enable bit 0 in the Standard Event Register to interrupt the
       Status Byte (2^0 = 1) */

    IOOUTPUTS(ADDR, "*ESE 1", 6);

    /* Call the function to check for errors */

    check_error("setup_status");
}
```

... continued

```
/*************************************************************************/
void meter_meas(void)
{
    /* Configure the meter for dc voltage measurements on channel 1
       (default channel). The measurement is set up on the 10 volt range
       with an integration time of 10 PLCs. */
    float readings[5];
    int stat_cond, i, numvalues = 5;
    static char *cmd_string[]=
{
    "SENS:FUNC 'VOLT:DC'",         /* Select dc volts on channel 1 */
    "SENS:VOLT:DC:RANG 10",        /* Select 10 volt range */
    "SENS:VOLT:DC:NPLC 10",        /* Select 10 power line cycles */
    "CALC:FUNC SCAL",              /* Select math scaling function */
    "CALC:STAT ON",                /* Turn math scaling on */
    "CALC:SCAL:GAIN 0.001",        /* Set the gain value (1 mV) */
    "CALC:SCAL:OFFS 0.5",          /* Set the offset value (50 mV) */
    "SAMP:COUN 5",                 /* Take 5 readings per trigger */
    "TRIG:SOUR EXT"                /* Select external trigger source */
};
/* Call the function to execute the command strings shown above */
command_exe(cmd_string, sizeof(cmd_string)/sizeof(char*));
/* Initiate the measurement, wait for the trigger, and then set bit 0
   "Operation Complete" in the Standard Event Register when complete. */
IOOUTPUTS(ADDR, "INIT;*OPC", 9);
/* Loop until the SRQ line is asserted when operation complete */
do
    IOSTATUS(ISC, 1, &stat_cond);
while (!stat_cond);
/* Transfer readings from internal memory to the output buffer */
IOOUTPUTS(ADDR, "FETC?", 5);
/* Enter readings and print */
IOENTERA(ADDR, readings, &numvalues);
for (i = 0; i numvalues; i++)
    printf("%f \n", readings[i]);
```

... continued

```
    /* Use Serial Poll to read the Status Byte and clear all bits */
    IOSPOLL(ADDR, &stat_cond);
    /* Call the function to check for errors */
    check_error("meter_meas");
}
/*************************************************************************/
void command_exe(char *commands[], int length)
{
    /* Execute one command string at a time using a loop */
    int loop;
    for (loop = 0; loop length; loop++)
    {
         IOOUTPUTS(ADDR, commands[loop], strlen(commands[loop]));
    }
}
/*************************************************************************/
void check_error(char *func_name)
{
    /* Read error queue to determine if errors have occurred */
    char message[80];
    int length = 80;
    IOOUTPUTS(ADDR, "SYST:ERR?", 9);       /* Read the error queue */
    IOENTERS(ADDR, message, &length);      /* Enter error string */
    while (atoi(message) != 0)             /* Loop until all errors are read */
    {
         printf("Error %s in function %s\n\n", message, func_name);
         IOOUTPUTS(ADDR, "SYST:ERR?", 9);
         IOENTERS(ADDR, message, &length);
    }
}
/*************************************************************************/
```

Microsoft Excel Macro Example

You can use Windows™ Dynamic Data Exchange (DDE) to send commands to the meter and return measurement results directly into a Microsoft® Excel spreadsheet.

This example uses the Keysight "Instrument Tools for Windows" and the 82335B GPIB Interface. The "Instrument Tools for Windows" contains two programs to support GPIB operations, an *Interactive* GPIB program and a GPIB *DDE Server*.

To use the "Instrument Tools for Windows" with Microsoft® Excel, follow this general procedure:

1. Install Tools for Windows (if not already installed).
2. Start the *Interactive* GPIB application and create and save a configuration file (.IBC file) that describes the interface with the meter. Be sure EOI is disabled and EOL is set to include the <lf> character. Refer to the "Instrument Tools for Windows" User's Manual for procedures and descriptions of creating a configuration (.IBC) file.

NOTE **You can test your interface connection in the Interactive GPIB program by sending an output string of *IDN? and entering the results. A typical response is shown in the example in cell F4.**

3. Start the GPIB DDE server. This application starts and minimizes to an icon.
4. Run the macro shown in the example.

   In the macro, the following commands are part of the GPIB DDE program and are documented in the "Instrument Tools for Windows" User's Manual. All other commands are Microsoft® Excel commands and are documented in the Excel function reference.

   - [Abort]
   - [clear(device)]
   - EnterS
   - EOL
   - OutputS

Macro Comments

- For simplicity, this macro is designed to output and enter data directly onto the macro sheet. You may wish to output and enter data on a worksheet.
- The macro sheet is divided into the following columns:

A	contains the variable names used in the macro
B	contains the macro commands
C	contains comments about the macro commands
D	place holder
E	contains the SCPI commands to send
F, G, ...	contains the returned data

- The macro takes temperature measurements every 15 minutes for a 12 hour duration, and enters the resulting statistics. Individual temperature readings are not stored.
- This macro uses the DATA:FEED command to overcome the 1024 memory reading maximum.
- The macro uses the CALCulate:STATe OFF and CALCulate:STATE ON commands to reset the stats for each 15 minute interval. Without these commands in the while loop, the stats at each interval would be an accumulation of readings for all measurements.

	A	B	C	D
1	*ExcelExample*	**Excel Example**	*Comments*	
2		=ECHO(FALSE)	*Turn off screen updates to increase speed*	
3	*MainLink*	=INITIATE("HPIBDDE","MAIN")	*Open a channel with the MAIN topic in HPIBDDE*	
4			*Returns a handle used in subsequent commands*	
5		=EXECUTE(MainLink,"[OPENCONFIG(XLSAMPLE)]")	*Opens a configuration file named XLSAMPLE.IO*	
6			*(created using interactive HP-IB environment)*	
7		=TERMINATE(MainLink)	*Close channel MAIN*	
8				
9	*MeterLink*	=INITIATE("HPIBDDE","METER")	*Open a channel to the meter*	
10				
11		=EXECUTE(MeterLink,"[abort]")	*Stop any interface activity*	
12		=EXECUTE(MeterLink,"[clear(device)]")	*Send device clear*	
13		=POKE(MeterLink,"OutputS(40)",E3)	*Send reset*	
14		=POKE(MeterLink,"OutputS(40)",E4)	*Send ID query*	
15		=SET.VALUE(F4,REQUEST(MeterLink,"EnterS(40)"))	*Enter result of query*	
16		=POKE(MeterLink,"OutputS(40)",E5)	*Send thermistor configuration command*	
17		=POKE(MeterLink,"OutputS(40)",E6)	*Send temperature units command*	
18		=POKE(MeterLink,"OutputS(40)",E7)	*Send trigger command*	
19		=POKE(MeterLink,"OutputS(40)",E8)	*Send stats command*	
20		=POKE(MeterLink,"OutputS(40)",E9)	*Send discard readings command*	
21		=POKE(MeterLink,"OutputS(40)",E10)	*Send stats enable command*	
22				
23	*Endtime*	=NOW()+TIMEVALUE("12:00:00")	*Run for 12 hours --- TEST DURATION*	
24	*Columncount*	=6	*Hour 1 in column 6*	
25				
26		=WHILE(NOW() < Endtime)		
27		= POKE(MeterLink,"OutputS(40)",E12)	*Start measurements*	
28				
29		= WAIT(NOW()+"00:15:00")	*Wait for 15 minutes -- TEST FREQUENCY*	
30				
31		= EXECUTE(MeterLink,"[clear(device)]")	*Stop measurements with device clear*	
32		= POKE(MeterLink,"OutputS(40)",E13)	*Turn off calculate to reset stats*	
33		= FORMULA("=TEXT(NOW(),""hh:mm:ss"")","R14C"&Columncount)	*Time stamp the data*	
34		= POKE(MeterLink,"OutputS(40)",E15)	*Get the minimum*	
35		= FORMULA(VALUE(REQUEST(MeterLink,"EnterS(40)")),"R15C"&Columnc	*Enter value*	
36		= POKE(MeterLink,"OutputS(40)",E16)	*Get the maximum*	
37		= FORMULA(REQUEST(MeterLink,"EnterS(40)"),"R16C"&Columncount)	*Enter value*	
38		= POKE(MeterLink,"OutputS(40)",E17)	*Get the average*	
39		= FORMULA(REQUEST(MeterLink,"EnterS(40)"),"R17C"&Columncount)	*Enter value*	
40		= POKE(MeterLink,"OutputS(40)",E18)	*Get the total number of readings*	
41		= FORMULA(REQUEST(MeterLink,"EnterS(40)"),"R18C"&Columncount)	*Enter value*	
42		= POKE(MeterLink,"OutputS(40)",E19)	*Get the peak-to-peak value*	
43		= FORMULA(REQUEST(MeterLink,"EnterS(40)"),"R19C"&Columncount)	*Enter value*	
44		= POKE(MeterLink,"OutputS(40)",E20)	*Get the standard deviation*	
45		= FORMULA(REQUEST(MeterLink,"EnterS(40)"),"R20C"&Columncount)	*Enter value*	
46		= SELECT("R14C"&Columncount&":R20C"&Columncount)	*Convert formula to numbers*	
47		= COPY()		
48		= PASTE.SPECIAL(3,1,FALSE,FALSE)		
49		= SET.VALUE(Columncount,Columncount+1)	*Increment column for next set of data*	
50		= POKE(MeterLink,"OutputS(40)",E21)	*Turn on CALCULATE system*	
51		=NEXT()		
52		=TERMINATE(MeterLink)	*Close the channel*	
53		=RETURN()		

	D	E	F	G	H	I	J	K
1		**SCPI Commands To Send**	**Data Returned by Queries**	---------->				
2								
3		*RST						
4		*IDN?	HEWLETT-PACKARD,34420A,0					
5		CONF:TEMPerature THER, DEF, 1, 0.000						
6		UNIT:TEMPERATURE F						
7		TRIGGER:COUNT INFinite						
8		CALCULATE:FUNCTION AVERAGE						
9		DATA:FEED RDG_STORE, ""						
10		CALCULATE:STATE ON						
11								
12		INIT						
13		CALCULATE:STATE OFF						
14			15:17:13	15:32:22	15:47:34	16:02:47	16:18:01	16:33:13
15		CALCULATE:AVERAGE:MINIMUM?	64.688	62.7854	62.798	63.7682	63.9248	64.0238
16		CALCULATE:AVERAGE:MAXIMUM?	66.1568	64.6718	63.842	64.0148	64.1156	64.2038
17		CALCULATE:AVERAGE:AVERAGE?	65.1605140112	63.8633350873	63.5882911315	63.8787225688	64.0190004415	64.1290021367
18		CALCULATE:AVERAGE:COUNT?	55884	55798	55680	55776	55723	55786
19		CALCULATE:AVERAGE:PTPEAK?	1.4688	1.8864	1.044	0.2466	0.1908	0.18
20		CALCULATE:AVERAGE:SDEVIATION?	0.397842973471	0.508862435818	0.242776229978	0.0617566369474	0.0423553138971	0.03534732759
21		CALCULATE:STATE ON						
22								
23								
24								

THIS PAGE HAS BEEN INTENTIONALLY LEFT BLANK.

7 Measurement Tutorial

Measurement Tutorial 272
Measurement Techniques and Sources of Error 273
Cleaning the Connector Contacts 297
Input Connector 299
Obtaining Parts For the Input Connector 300

Measurement Tutorial

The 34420A is capable of making highly accurate voltage, resistance or temperature measurements. In order to achieve the greatest accuracy, you must take the necessary steps to eliminate potential measurement errors.

This chapter is divided into two sections. The first section discusses the causes of measurement errors. The second section describes specific ways to use the meter to minimize measurement errors.

This chapter is divided into the following sections:

- "Measurement Techniques and Sources of Error" on page 273
  - "Voltage Measurements" on page 273
  - "Resistance Measurements" on page 283
  - "Temperature Measurements" on page 288
- "Cleaning the Connector Contacts" on page 297
- "Input Connector" on page 299
- "Obtaining Parts For the Input Connector" on page 300

Measurement Techniques and Sources of Error

Voltage Measurements

The following table shows voltage measurement error sources. Each of these sources of error is described in more detail in the sections that follow.

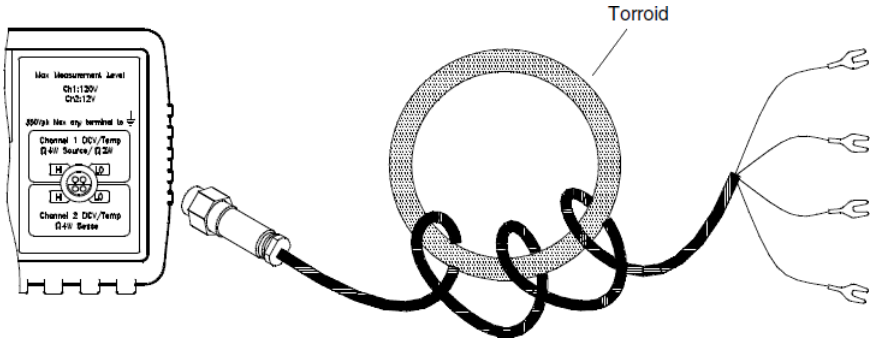
	Error Source	Correction	Page
Cabling and connection related	Radio Frequency Interference (RFI)	Special circuits and shields are used to minimize the effects of RFI in the measurement. Low emission of RFI also allows the use of the meter with sensitive devices such as Josephson junctions where high frequency signals may interfere with proper operation of the device.	275
	Thermal EMF Errors	The meter has 99% copper terminals with soldered connections. The test leads are copper wire with crimped copper terminals. These test connectors and leads are recommended to help minimize the effects of thermal EMF errors.	276
	Magnetic Loops	Use twisted-pair connections to the meter to reduce the noise pickup loop area, or dress the test leads as close together as possible. Loose or vibrating test leads will also induce error voltages. Make sure your test leads are tied down securely when operating near magnetic fields. Whenever possible, use magnetic shielding materials or physical separation to reduce problem magnetic field sources. Use NPLC ≥ 1 if the pickup is power line related.	276
	Power Line Rejection (NMR)	Set the integration time to 1 or more PLCs.	277
	Ground Loops	The best way to eliminate ground loops is to maintain the meter's isolation from earth; do not connect the input terminals to ground. If the meter must be earth-referenced, be sure to connect it, and the device-under-test, to the same common ground point. This will reduce or eliminate any voltage difference between the devices. Also make sure the meter and device-under-test are connected to the same electrical outlet whenever possible.	278

	Error Source	Correction	Page
Cabling and connection related	Common Mode Rejection	Reduce the series resistance or common mode voltage.	279
Meter related	Loading Errors	The meter's input resistance is > 10 GΩ for 1 mV to 10 V ranges, 10 MΩ for 100 volt range.	280
	Input Bias Current (Leakage Current)	The meter's measuring circuitry exhibits approximately 30pA of input bias current for ambient temperatures from 0°C to 30°C. Bias current will change at temperatures different than those at which the meter was calibrated.	281
	Injected Current	A circuit may or may not be sensitive to injected current depending on the topology of circuit being measured by the voltmeter, the impedance level of the circuit, and the common mode or normal mode rejection of the meter.	281

Radio Frequency Interference

Most voltage measuring instruments can generate false readings in the presence of large high frequency fields. Possible sources of such fields could include radio, television, communication transmitters, and portable telephones. The high frequency energy can couple into the meter on the input leads or cables connected to the rear of the instrument. Keeping the test leads and rear panel cables as far removed as possible from high frequency sources will minimize the RFI interference.

RFI emissions from the meter meet international standards. If your application is extremely sensitive to RFI radiated from the meter, use a common mode choke in the input cable as shown.



Thermal EMF Errors

Thermoelectric voltages are the most common source of error in low-level dc voltage measurements. Thermoelectric voltages are generated when you make circuit connections using dissimilar metals at different temperatures. Each metal-to-metal junction forms a *thermocouple*, which generates a voltage proportional to the junction temperature. You should take the necessary precautions to minimize thermocouple voltages and temperature variations in low-level voltage measurements. The best connections are formed using copper-to-copper crimped connections. The table below shows common thermoelectric voltages for connections between dissimilar metals.

Copper-to-	Approx. μV/ °C
Copper	<0.3
Gold	0.5
Silver	0.5
Brass	3
Beryllium Copper	5
Aluminum	5
Kovar or Alloy 42	40
Silicon	500
Copper-Oxide	1000
Cadmium-Tin Solder	0.2
Tin-Lead Solder	5

Noise Caused by Magnetic Fields

If you are making measurements near magnetic fields, you should take the necessary precautions to avoid inducing voltages in the measurement connections. You should be especially careful when working near conductors carrying large currents.

Rejecting Power-Line Noise Voltages

A desirable characteristic of integrating analog-to-digital (A/D) converters is their ability to reject spurious signals. Integrating techniques reject power-line related noise present with dc signals on the input. This is called *normal mode rejection* or NMR. Normal mode noise rejection is achieved when the meter measures the average of the input by "integrating" it over a fixed period. If you set the integration time to a whole number of power line cycles (PLCs) of the spurious input, these errors (and their harmonics) will average out to approximately zero.

When you apply power to the meter, it measures the power-line frequency (50 Hz or 60 Hz), and uses this measurement to determine the integration time. The table below shows the noise rejection achieved with various configurations. For better resolution and increased noise rejection, select a longer integration time.

NPLC's	Integration Time		NMR
	60 Hz	(50 Hz)	
0.02	334 ms	(400 ms)	none
.2	3 ms	(4 ms)	none
1	16.7 ms	(20 ms)	60 dB
2	33.4 ms	(40 ms)	90 dB
10	167 ms	(200 ms)	95 dB
20	334 ms	(400 ms)	100 dB
100	1.67 s	(2 s)	105 dB
200	3.34 s	(4 s)	110 dB

Noise Caused by Ground Loops

When measuring voltages in circuits where the meter and the device-under-test are both referenced to a common earth ground, a “ground loop” is formed. As shown below, any voltage difference between the two ground reference points (V_{ground}) causes a current to flow through the **LO** measurement lead. This causes an error voltage, V_L, which is added to the measured voltage.

R_L = *Lead Resistance*

R_i = *Multimeter Isolation Resistance*

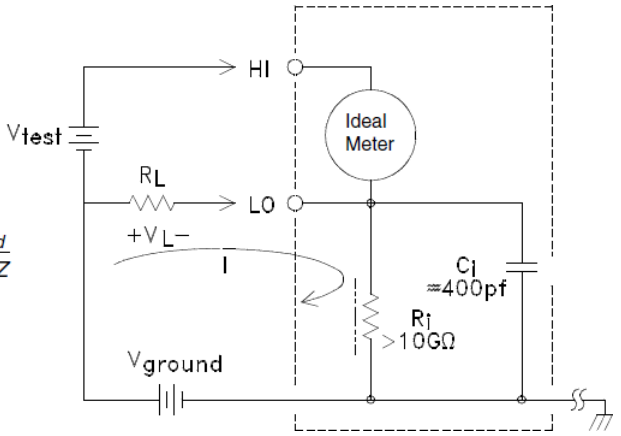
C_i = *Multimeter Isolation Capacitance*

V_{ground} = *Ground Noise Voltage*

$$I = V_{ground} \text{ caused current flow} = \frac{V_{ground}}{R_L + Z}$$

$$Z = \frac{R_i * Z_{C_i}}{R_i + Z_{C_i}}$$

$$V_L = I * R_L$$



To minimize ground loop errors:

- If V_{ground} is a *dc voltage*, keep R_L small compared to R_i.
- If V_{ground} is an *ac voltage*, keep R_L small compared to Z, and set the meter’s integration time (NPLC) to 1 or greater. *See page 71*.

Common Mode Rejection (CMR)

Ideally, a meter is completely isolated from earth-referenced circuits. However, there is finite resistance and capacitance between the meter's input LO terminal and earth ground. If the input terminals are both driven by an earth referenced signal, V_f, then a current will flow through R_S and create a voltage drop V_L as shown below.

V_f = *Common Mode Float voltage*

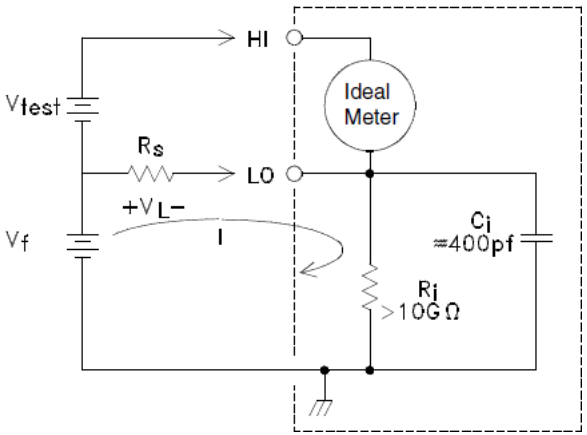
R_S = *LO Lead Resistance*

R_i = *Meter Isolation Resistance*

C_i = *Meter Isolation Capacitance*

Z_i = *Parallel Impedance of* $R_i + C_i$

$$\text{Error } (V_L) = \frac{V_f \times R_S}{R_S + Z}$$



Any resulting voltage, V_L, will appear as an input to the meter. As the value of R_S approaches 0, so does the error. Additionally, if V_f is at the power line frequency (50 Hz or 60 Hz), the noise can be greatly reduced by setting the meter's integration time (NPLC) to 1 or greater. *See page 277*.

Loading Errors Due to Input Resistance (dc volts)

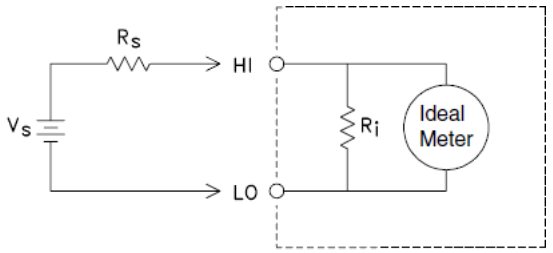
Measurement loading errors occur when the resistance of the device-under- test (DUT) is an appreciable percentage of the meter's own input resistance. The diagram below shows this error source.

V_s = *Ideal DUT voltage*

R_S = *DUT source resistance*

R_i = *Meter input resistance* (10 MΩ or > 10 GΩ)

$$Error\ (\%) = \frac{100 \times R_S}{R_S + R_i}$$



Loading Errors Due to Input Bias Current (Leakage Current)

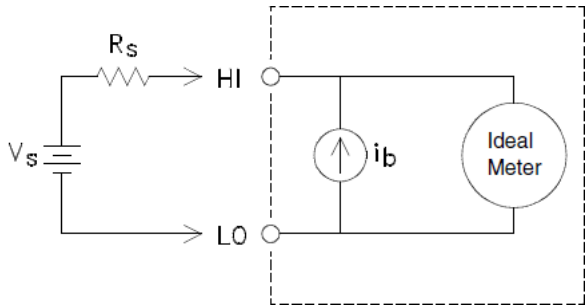
The semiconductor devices used in the input circuits of the meter have slight leakage currents, called bias currents. The effect of the input bias current is an loading error at the meter's input terminals.

i_b = meter bias current

R_S = DUT source resistance

C_i = meter input capacitance

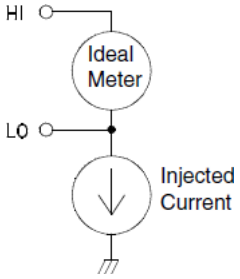
Error (V) ≈ i_b x R_S



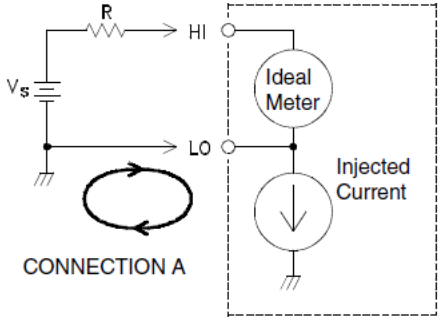
The leakage current will approximately double for every 10 °C temperature rise, making the problem much more apparent at higher temperatures. You can adjust the input bias current to remove these errors. The bias current is automatically minimized during the zero calibration procedure (see Chapter 4 in the *34420A Service Guide*). You should perform the zero calibration procedure if you are operating the meter at a temperature more than 5 °C from the calibration temperature.

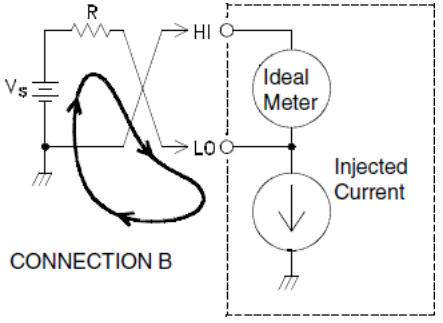
Noise Caused by Injected Current

Residual capacitances in the power transformer cause small currents to flow from the LO of the meter to earth ground. The frequency of the “injected current” is the power line frequency or possibly power line frequency harmonics. The injected current is dependent upon the power line configuration and frequency. You will need to recalibrate the injected current if you operate the meter with a different power line than the power line used to calibrate the meter. A simplified circuit is shown.



In the circuit below, with connection A the injected current flows from the earth connection provided by the circuit, to the LO terminal of the voltmeter, and no noise is added to the measurement. However with connection B, the injected current flows through the resistor R adding noise to the measurement. With connection B, larger values of R will make the problem worse.



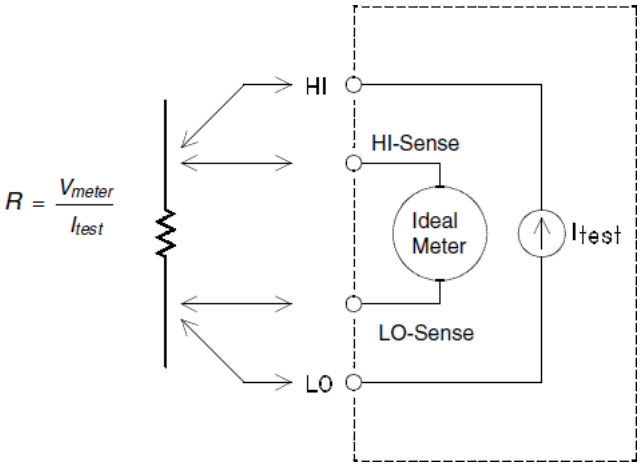


The noise caused by injected current can be significantly reduced by setting the integration time of the meter (NPLC) to 1 or greater. *See* *page 277*.

Resistance Measurements

Specific resistance measurement techniques are discussed in the following sections. You should also consider the techniques and errors discussed earlier in this chapter.

The 4-wire ohms method provides the most accurate way to measure small resistances. Test lead resistances and contact resistances are automatically reduced using this method. The connections for resistance measurements are shown below. *See also* *"Resistance Measurement Configuration"* on page 81.



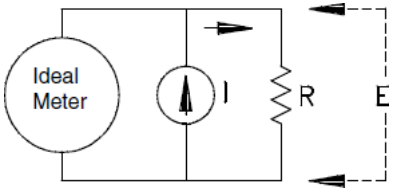
Removing 2-wire Ohm Test Lead Resistance

To eliminate offset errors associated with the test lead resistance in 2-wire ohms measurements, follow the steps below.

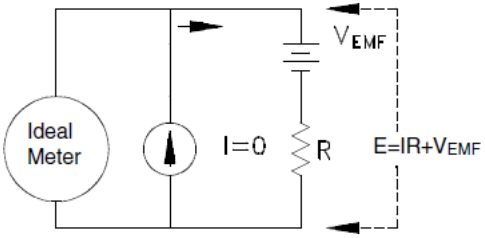
1. Short the ends of the test leads together. The meter displays the test lead resistance.
2. Press Null from the front panel. The meter displays "0" ohms with the leads shorted together.

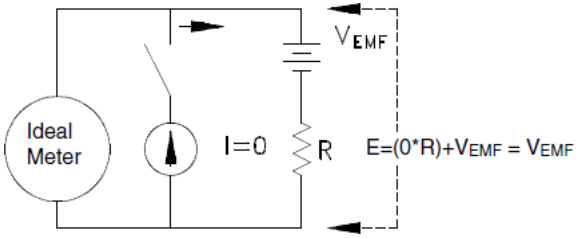
Offset Compensation

A resistance measurement involves measuring a voltage (E) induced across the resistance by a known current source.



Thermal EMF caused by dissimilar metals can create a parasitic voltage in the measurement circuit (V_{EMF}). (*See page 276 for a description of thermal EMF*). The thermal EMF can be caused by the input lead connections or internally in resistor R. In general, this voltage will not change with the current applied to the resistor.





The voltage measured, and so the resistance calculated, is in error by V_{EMF}. Using *offset compensation* can reduce the errors caused by V_{EMF}. To make an offset compensated measurement, the meter makes two voltage measurements, one with the current source on and one with the current source off, and subtracts the two measurements. The actual voltage drop across the resistor, and the calculated resistance are obtained by:

$$\textit{First Reading - Second Reading} = (I * R + V_{EMF}) - V_{EMF} = I * R$$

Offset compensation can be used in 2-Wire or 4-Wire ohms measurements.

In the power-on state of the meter, offset compensation is enabled. In some circumstances it may be desirable to turn offset compensation off. For example if the resistor being measured does not respond quickly to changes in current, offset compensation will not produce an accurate measurement. Resistors with very large inductances or resistors with large values of parallel capacitance would fall into this category. In these cases the DELAY parameter can be increased to allow more settling time after the current source is switched on or off, or the offset compensation function can be turned off.

Power Dissipation Effects

When measuring resistors designed for temperature measurements (or other resistive devices with large temperature coefficients), be aware that the meter will dissipate some power in the device-under-test. The effects of this power dissipation can affect the measurement accuracy.

If power dissipation is a problem, there are two actions you can take: you can select a higher measurement range to reduce the errors to acceptable levels or you can set the meter to the low power mode to limit the test current (*see page 82*). The following table shows several examples.

	Normal		Low Power	
Range	Test Current	DUT Power (at Full Scale)	Test Current	DUT Power (at Full Scale)
1 Ω	10 mA	100 µW	10 mA	100 µW
10 Ω	10 mA	1 mW	10 mA	1 mW
100 Ω	10 mA	10 mW	1 mA	100 µW
1 kΩ	1 mA	1 mW	100 µA	10 µW
10 kΩ	100 µA	100 µW	10 µA	1 µW
100 kΩ	10 µA	10 µW	5 µA	2.5 µW
1 MΩ	5 µA	25 µW	5 µA	25 µW

Output Clamping (Dry Circuit Testing)

Resistance measurements on certain types of contacts may require a limitation on the voltage levels used to make the measurement. Both the open circuit and the measurement voltage should be considered. The need for voltage limitation arises from the possibility that oxidation on the contact surfaces may increase the resistance reading. If the voltage is too high, the oxide layer may be punctured resulting in a lower resistance reading.

The meter provides a programmable level of open circuit clamping, called *voltage limited* measurements (*see page 83*). This feature is available on the 10 and 100 ohm ranges. The open circuit voltage can be clamped at one of three levels, 20 mV, 100 mV, or 500 mV.

The table below shows the current and voltage levels used.

Range	Test Current	Measurement Voltage (at Full Scale)	Open Circuit Voltage (MV)
10 Ω	1 mA	10 mV	20, 100, or 500 mV
100 Ω	0.1 mA	10 mV	20, 100, or 500 mV

Settling Time Effects

The meter has the ability to insert automatic measurement settling delays. These delays are adequate for resistance measurements with less than 200 pF of combined cable and device capacitance. This is particularly important if you are measuring resistances above 100 kΩ. Settling due to RC time constant effects can be quite long. Some precision resistors and multi-function calibrators use large parallel capacitors (1000 pF to 0.1 μF) with high resistor values to filter out noise currents injected by their internal circuitry. Non-ideal capacitances due to dielectric absorption (soak) effects in cables and other devices may have much longer settling times than expected just by RC time constants. Errors will be measured when settling after the initial connection and after a range change. You may need to increase the delay time before a measurement in these situations (*see page 100*).

Errors in High Resistance Measurements

When you are measuring large resistances, significant errors can occur due to insulation resistance and surface cleanliness. You should take the necessary precautions to maintain a “clean” high-resistance system. Test leads and fixtures are susceptible to leakage due to moisture absorption in insulating materials and “dirty” surface films. Nylon and PVC are relatively poor insulators (10^9 ohms) when compared to PTFE insulators (10^{13}Ω). Leakage from nylon or PVC insulators can easily contribute a 0.1% error when measuring a 1 MΩ resistance in humid conditions.

Temperature Measurements

A temperature measurement is either a resistance measurement or a voltage measurement, converted to a temperature mathematically inside the meter. The mathematical conversions requires the knowledge of specific properties of specific transducer types. The conversion routines used are compatible with ITS-90. The conversion accuracy (not including the transducer accuracy) for each transducer type is:

RTD	0.05 °C
Thermistor	0.1 °C
Thermocouple	0.2 °C

Errors associated with temperature measurements include all those listed for dc Voltage and Resistance Measurements. The largest source of error in temperature measurements is the transducer itself.

Your measurement requirements determine which temperature transducer type to use. Each transducer type has a particular temperature range, accuracy, and cost. Each type of transducer is described in more detail in the following subsections. The table below summarizes some general specifications for each type of transducer. Use the table to help select the general type of transducer to use; transducer manufacturers can provide you with exact specifications for a particular transducer.

	RTD	Thermistor	Thermocouple
Temperature Range	-200 to 850 °C	-80 to 150 °C	-210 to 1820 °C
Measurement Type	4-wire Resistance	2-wire Resistance	Voltage
Transducer Sensitivity	≈R_0 X 0.004 °C	≈400 Ω /°C	6 to 60 μV/°C
Accuracy	0.01 to 0.1 °C	0.1 to 1 °C	0.5 to 5 °C
Cost (approx)	$20.00 to $100.00 ea[a]	$10.00 to $100.00 ea[a]	$1.00/foot[a]
Durability	Fragile	Fragile	Rugged

[a] Estimated cost in U.S. Dollars

RTDs

An RTD uses a metal (typically platinum) that changes resistance with a change in temperature. Your meter can measure this resistance and by knowing the characteristics of the RTD, calculate the temperature.

RTDs have the highest stability of the temperature transducers. Their output is also very linear. This makes the RTD the transducer of choice for high accuracy, long-term measurements. RTDs come in two main types: α = 0.00385 (DIN/IEC 751) and α = 0.00391.

The RTD resistance at 0 °C is referred to as nominal or R_0. The meter can use RTD's with R_0 values from 4.9 Ω to 2.1 kΩ.

The meter always makes a 4-wire measurement of RTDs to preserve their accuracy.

Thermistors

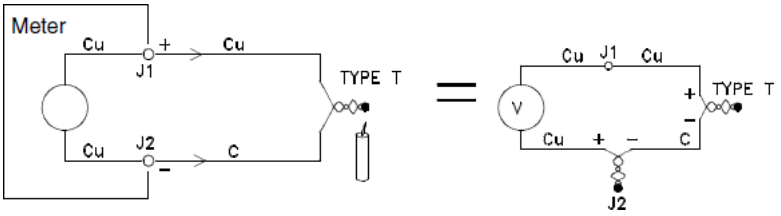
Thermistors use materials that change resistance with changes in temperature. Your meter can measure this resistance and by knowing the characteristics of the thermistor, calculates the temperature.

Thermistors have a higher sensitivity than thermocouples or RTDs. This makes thermistors the transducer to use when measuring very small changes in temperature. Thermistors are, however, very non-linear, especially at high temperatures and function best below 100 °C.

Because of their high resistance, thermistors can be measured using two wire measurement techniques. The meter supports 5000 Ω thermistors.

Thermocouples

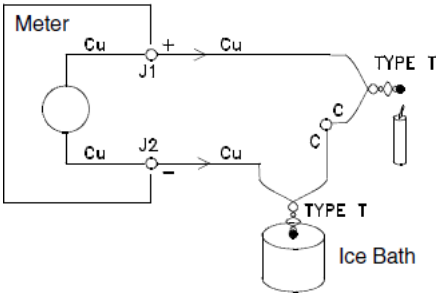
A thermocouple converts temperature to voltage. When two wires, composed of dissimilar metals, are joined, an open circuit voltage is generated. The voltage is a function of the junction temperature and the types of metals in the wire. Since the temperature characteristics of special dissimilar metals is well known, a conversion from the voltage generated to the temperature of the junction can be made. For example, a voltage measurement of a type T thermocouple (made up of copper and constantan wire) might look like this:



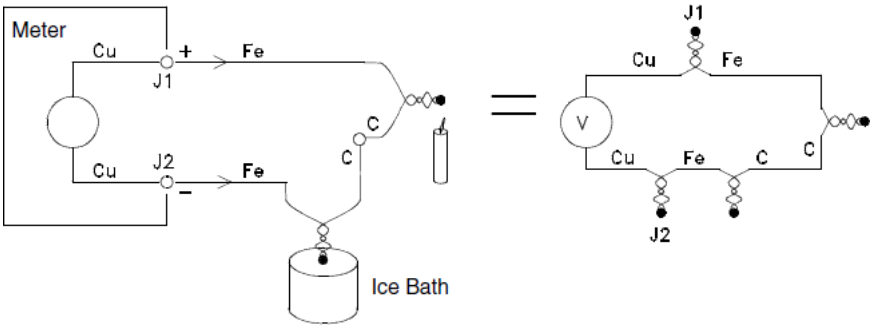
Notice, however, that the connections made between the thermocouple wire and the meter make another thermocouple where the thermocouple's constantan (C) lead connects to the meter's copper (Cu) input terminal. The voltage generated by this second thermocouple affects the voltage measurement of the type T thermocouple.

If the temperature of the thermocouple created at J2 (the LO input terminal) is known, the temperature of the type T thermocouple can be calculated. One way to do this is to connect two type T thermocouples together to create only copper to copper connections at the meter's input terminals, and to hold the second thermocouple at a known temperature.

An ice bath is used to create a known reference temperature (0 °C). Once the reference temperature and thermocouple type are known, the temperature of the measurement thermocouple can be calculated.

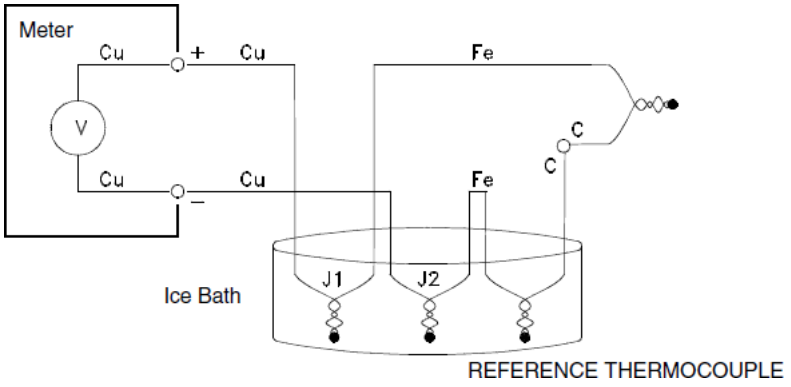


The type T thermocouple is a unique case since one of the conductors (copper) is the same metal as the meter's input terminal. If another type of thermocouple is used, two additional thermocouples are created. For example, the connections with a type J thermocouple (iron and constantan):

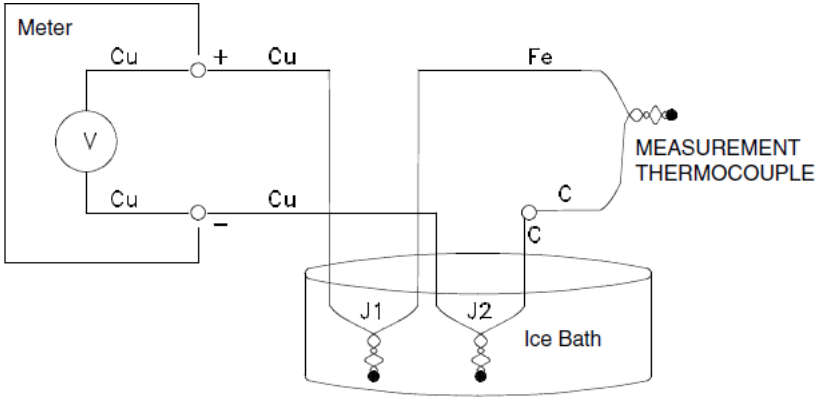


Two additional thermocouples have been created where the iron lead connects to the meter's copper input terminals. Since these two junctions will generate opposing voltages, their effect will be to cancel each other. However, if the input terminals are not at the same temperature, then an error will be created in the measurement.

To make a more accurate measurement, the copper test leads of the meter should be extended closer to the measurement and the connections to the thermocouple be held at the same temperature.



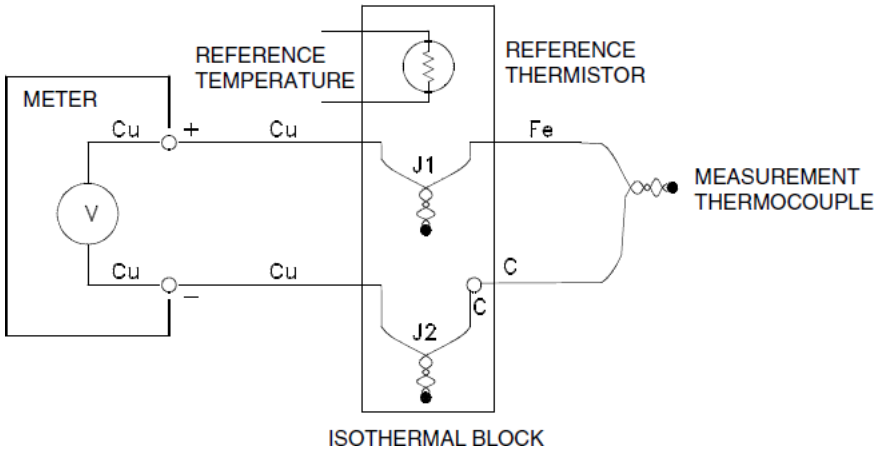
This circuit will give accurate temperature measurements. However, it is not very convenient to make two thermocouple connections and keep all connections at a known temperature. The “Law of Intermediate Metals” eliminates the need for the extra connection. This empirical law states that a third metal (iron (Fe) in the example) inserted between two dissimilar metals will have no effect upon the output voltage provided the junctions formed are at the same temperature. Removing the reference thermocouple makes the connections much easier.



This circuit is the best solution for accurate thermocouple connections.

In some measurement situations, however, it would be nice to remove the need for an ice bath (or other fixed external reference). To do this, an *isothermal block* is used to make the connections. An isothermal block is an electrical insulator, but a good heat conductor. The additional thermocouples created at J1 and J2 now held at the same temperature by the isothermal block.

Once the temperature of the isothermal block is known, accurate temperature measurements can be made. Typically, a thermistor is mounted to the isothermal block and used to measure the temperature.



Thermocouples come in a variety of types. The type is specified by a single letter. The following table shows the most commonly used thermocouple types and some key characteristics of each.

Type	+ lead	- lead	Temperature Range	Probe Accuracy	Comments
B	**Platinum – 30% Rhodium**	**Platinum– 60% Rhodium**	250 to 1820 °C	± 0.5 °C	High Temperature. Beware of contamination. Do not insert in metal tubes.
U.S.	Gray	Red			
British	N/A	N/A			
DIN	Red	Gray			
Japanese	Red	Gray			
French	N/A	N/A			
J	**Iron**	**Constantan**	-210 to 1200 °C	± 1.1 to 2.2 °C	For vacuum, inert environments. Least expensive. Not recommended for low temperature.
U.S.	White	Red			
British	Yellow	Blue			
DIN	Red	Blue			
Japanese	Red	White			
French	Yellow	Black			
K	**Nickel – Chromium**	**Nickel – Aluminum**	-200 to 1370 °C	± 1.1 to 2.2 °C	For oxidizing environments. Good linearity above 8 °C.
U.S.	Yellow	Red			
British	Brown	Blue			
DIN	Red	Green			
Japanese	Red	White			
French	Yellow	Purple			
T	**Copper**	**Constantan**	-200 to 400 °C	± 0.5 to 1 °C	Withstands moisture. Has a copper lead. Low temperature applications.
U.S.	Blue	Red			
British	White	Blue			
DIN	Red	Brown			
Japanese	Red	White			
French	Yellow	Blue			
E	**Nickel – Chromium**	**Constantan**	-200 to 1000 °C	± 1 to 1.7 °C	Highest output voltage. Highest resolution.
U.S.	Purple	Red			
British	Brown	Blue			
DIN	Red	Black			
Japanese	Red	White			
French	Yellow	Blue			

Type	+ lead	- lead	Temperature Range	Probe Accuracy	Comments
N	**Nicrosil**	**Nisil**			
U.S.	Orange	Red			
British	N/A	N/A	-200 to 1300 °C	± 1.1 to 2.2 °C	Superior stability to type K at high temperatures.
DIN	N/A	N/A			
Japanese	N/A	N/A			
French	N/A	N/A			
R	**Platinum – 13% Rhodium**	**Rhodium**			
U.S.	Black	Red			
British	White	Blue	-50 to 1760 °C	± 0.6 to 1.5 °C	High Temperature. Beware of contamination. Do not insert in metal tubes.
DIN	Red	White			
Japanese	Red	White			
French	Yellow	Green			
S	**Platinum – 10% Rhodium**	**Platinum**			
U.S.	Black	Red			
British	White	Blue	-50 to 1760 °C	± 0.6 to 1.5 °C	Low error, good stability. High Temperature. Beware of contamination. Do not insert in metal tubes.
DIN	Red	White			
Japanese	Red	White			
French	Yellow	Green			

Constantan = Copper–Nickel, Nicrosil = Nickel–Chromium–Silicon, Nisil = Nickel–Silicone–Magnesium, N/A = Not Available

Thermocouple Measurement Errors

The meter allows you to use one of three types of temperature reference for thermocouple measurements:

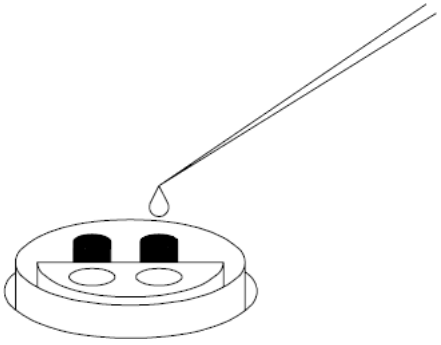
- **Fixed Reference value (entered).** The fixed reference is the most accurate, and is the type of thermocouple measurement described earlier using isothermal blocks and an external known reference temperature.
- **Measured Reference value (external thermistor).** A thermistor is connected to channel 1 input of the meter. The thermistor is then mounted to the isothermal block. The meter measures the temperature of the isothermal block and uses the measurement in the temperature calculation.
- **Internal Reference value (internal thermistor).** The meter contains an internal thermistor, mounted just behind the input connector. The internal thermistor is the least accurate way to measure temperature. If you use the internal thermistor, you should connect the thermocouple wire directly to the input connector (do not use the copper test leads or an isothermal block). *See page 299*.

Cleaning the Connector Contacts

Because the meter uses connector contacts that are almost pure copper, they are subject to oxidation. Oxidation can cause measurement errors. *See page 276*.

To help prevent the formation of oxides on the connector contacts, keep the connector engaged whenever possible. After extended periods of time, it may be necessary to treat the connector contacts. The recommended treatment uses DeoxIT™[1] and a small bottle of DeoxIT was included with the 34420A. Use the following procedure to apply DeoxIT to the connector terminals.

1. Remove the power cord from the meter.
2. Remove the input connector.
3. Stand the meter on a smooth level surface so that the front panel is up.
4. Apply one drop of DeoxIT to each connector pin in the input connector. A convenient method to apply a single drop is to use a clean piece of wire to transfer the liquid from the container provided to the connector.



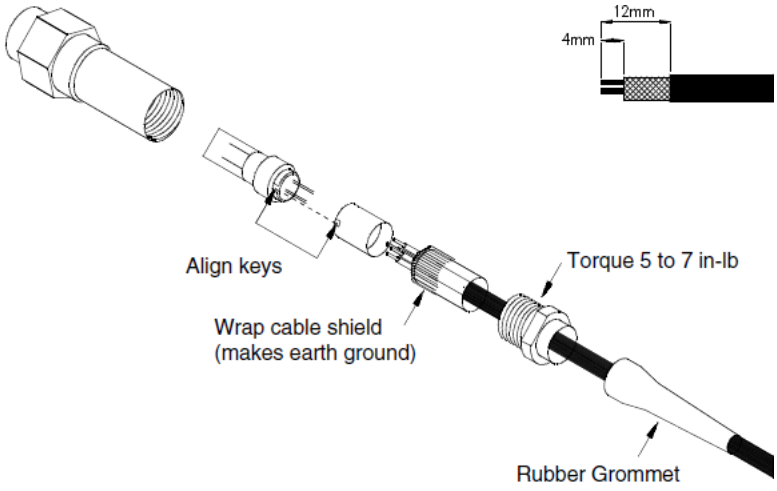
5. Remove any excess DeoxIT from the connector by wiping with a clean soft cloth.
6. Engage and disengage the input cable several times to distribute the DeoxIT evenly over both halves of the connector.

[1] DeoxIT™ is a trademark of CAIG Laboratories, Inc., San Diego, California.

NOTE The input connector may cause measurement problems if the coupling nut is not tightened properly, the connector keys are not aligned properly, or the clamping nut is not tightened properly. See the next page.

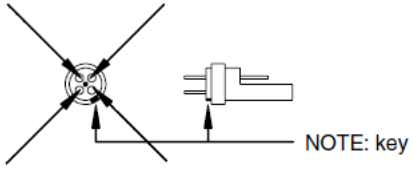
Input Connector

WIRE: 22 - 28 guage copper



Channel 1

	HI	LO
DCV	+	–
4-WireΩ	Source +	Source –
2-WireΩ	Source and Sense +	Source and Sense –
Thermistor	Source and Sense +	Source and Sense –



Channel 2

	HI	LO
DCV	+	–
4-WireΩ	Sense +	Sense –
Thermocouple	+	–

Obtaining Parts For the Input Connector

Use the following Keysight Technologies part numbers to obtain replacement input connectors and cables. Contact your nearest Keysight Technologies Sales Office for replacement.

Part Number	Description
34104A	Low thermal input connector (blank)
34103A	Low thermal 4-wire shorting plug
34102A	Low thermal input connector and 4 foot cable with spade lugs.

You can obtain additional DeoxIT™, part number D100L2, from:

CAIG Laboratories, Inc.
16744 West Bernardo Drive
San Diego, CA 92172-1904

8 Specifications

For the specifications and characteristics of the 34420A Nano Volt/Micro Ohm Meter, refer to the datasheet at http://literature.cdn.keysight.com/litweb/pdf/5968-0161EN.pdf

THIS PAGE HAS BEEN INTENTIONALLY LEFT BLANK.

This information is subject to change without notice. Always refer to the Keysight website for the latest revision.

© Keysight Technologies 1994 - 2021
Edition 5, January 2021

Printed in Malaysia

34420-90001
www.keysight.com