CALIBRATION PROCEDURE

NI 4071 7½-Digit FlexDMM™

This document contains instructions for writing an external calibration procedure for the National Instruments PXI-4071 (NI 4071) 7½-digit FlexDMM and 1.8 MS/s isolated digitizer. For more information about calibration, visit `ni.com/calibration`.

Contents

NATIONAL INSTRUMENTS™

Conventions

The following conventions are used in this manual:

» The » symbol leads you through nested menu items and dialog box options to a final action. The sequence **Options»Settings»General** directs you to pull down the **Options** menu, select the **Settings** item, and select **General** from the last dialog box.

This icon denotes a note, which alerts you to important information.

This icon denotes a caution, which advises you of precautions to take to avoid injury, data loss, or a system crash. When this symbol is marked on a product, refer to the *Read Me First: Safety and Electromagnetic Compatibility* document included with the device for information about precautions to take.

bold Bold text denotes items that you must select or click in the software, such as menu items and dialog box options. Bold text also denotes parameter names.

italic Italic text denotes variables, emphasis, a cross-reference, hardware labels, or an introduction to a key concept. Italic text also denotes text that is a placeholder for a word or value that you must supply.

`monospace` Text in this font denotes text or characters that you should enter from the keyboard, sections of code, programming examples, and syntax examples. This font is also used for the proper names of disk drives, paths, directories, programs, subprograms, subroutines, device names, functions, operations, variables, filenames, and extensions.

Software Requirements

NI-DMM supports a number of programming languages including LabVIEW, LabWindows™/CVI™, Microsoft Visual C++, and Microsoft Visual Basic. When you install NI-DMM, you need to install support for only the language you intend to use to write your calibration utility. The procedures in this document are described using LabVIEW VIs and C function calls.

Note You must use NI-DMM version 3.0.2 or later with this calibration procedure.

Documentation Requirements

In addition to this calibration document, you may find the following references helpful in writing your calibration utility. All these documents are installed on your computer when you install NI-DMM. To locate them, select **Start»All Programs»National Instruments»NI-DMM» Documentation**.

- *NI Digital Multimeters Help*
- *NI Digital Multimeters Getting Started Guide*

NI recommends referring to the *NI 4071 Specifications* online at `ni.com/manuals` to ensure you are using the latest NI 4071 specifications.

Calibration Function Reference

For detailed information about the NI-DMM calibration VIs and functions in this procedure, refer to the LabVIEW reference or the C/CVI/VB reference sections of the *NI Digital Multimeters Help*, located at **Start»All Programs»National Instruments»NI-DMM»Documentation**. Refer to Figure 7 for the procedural flow for verification. Refer to Figure 8 for the procedural flow for adjustment.

Password

The password is required to open an external calibration session. If the password has not been changed since manufacturing, the password is `NI`.

Calibration Interval

The accuracy requirements of your measurement application determine how often you should calibrate the NI 4071. NI recommends performing a complete calibration at least once every two years. NI does not guarantee the absolute accuracy of the NI 4071 beyond this two-year calibration interval. You can shorten the calibration interval based on the demands of your application. Refer to *Appendix A: Calibration Options* for more information.

Test Equipment

Table 1 lists the equipment required for calibrating the NI 4071. If you do not have the recommended instruments, use these specifications to select a substitute calibration standard.

Table 1. Required Test Equipment

Required Equipment	Recommended Models
Multifunction calibrator	Fluke 5720A* (calibrated within the last 90 days)
Two sets of low thermal electromotive force (EMF) copper cables	Two sets of Fluke 5440 cables
A means of creating a short (≤100 mΩ) with low thermal EMF (≤150 nV) across the HI and LO input banana plug connectors on the NI 4071	Pomona 5145 insulated double banana plug shorting bar
Two sets of banana-to-banana cables with length not to exceed 4 in.	Two Pomona B-4 banana-to-banana patch cords (cables)
Double banana plug with binding posts	Pomona 5405 Binding Post
Insulated low thermal electromotive force (EMF) spade lugs	Two Pomona 2305 lugs
Chassis	National Instruments PXI chassis and controller
A device capable of generating pulse trains at the frequencies listed in Table 27	NI PXI-6608 counter/timer module or Agilent 33250a function/arbitrary waveform generator
Any additional equipment needed to connect the external frequency source to the NI 4071	NI TB-2715 terminal block
Shielded cable	NI SH68-68-D1 shielded cable
Double banana plug with strain relief	Pomona MDP 4892 double banana plug with strain relief
Coaxial cable	RG178
1 GΩ reference standard resistor	IET Labs SRL-1G

* The 90-day DC current uncertainty of the Fluke 5720A is not adequate to calibrate the 100 µA and 1 mA DC current ranges on the NI 4071. See Table 13 in the *Verifying DC Current* section for information on the required uncertainty.

Test Conditions

Follow these guidelines to optimize the connections and the environment during calibration:

- Ensure that the PXI chassis fan speed is set to *HI* and that the fan filters are clean.
- Use PXI filler panels in all vacant slots to allow proper cooling.
- Plug the PXI chassis and the calibrator into the same power strip to avoid ground loops.
- Power on and warm up both the calibrator and the NI 4071 for at least 60 minutes before beginning this calibration procedure.
- Maintain an ambient temperature of 23 ±1 °C.
- Maintain an ambient relative humidity of less than 60%.
- Allow the calibrator to settle fully before taking any measurements. Consult the Fluke 5720A user documentation for instructions.
- Allow the thermal EMF enough time to stabilize when you change connections to the calibrator or the NI 4071. The suggested time periods are stated where necessary throughout this document.
- Keep a shorting bar connected between the *V GUARD* and *GROUND* binding posts of the calibrator at all times.
- Clean any oxidation from the banana plugs on the Fluke 5440 cables before plugging them into the binding posts of the calibrator or the banana plug connectors of the NI 4071. Oxidation tarnishes the copper banana plugs so that they appear dull rather than shiny and leads to greater thermal EMF.
- Keep the blue banana plugs on the Fluke 5440 cables connected to the *V GUARD* binding post of the calibrator at all times.
- Prevent the cables from moving or vibrating by taping or strapping them to a nonvibrating surface. Movement or vibration causes triboelectric effects that can result in measurement errors.

Calibration Procedures

The calibration process includes the following steps:

1. *Initial Setup*—Set up the test equipment.
2. *Verification Procedures*—Verify the existing operation of the device. This step confirms whether the device is operating within its specified range prior to calibration. Figure 7 in Appendix A shows the procedural flow for verification.
3. *Adjustment Procedures*—Perform an external adjustment of the device that adjusts the calibration constants with respect to standards of known values. Figure 8 in Appendix A shows the procedural flow for adjustment.
4. Reverification—Repeat the verification procedure to ensure that the device is operating within its specifications after adjustment.

These steps are described in more detail in the following sections.

Note In some cases, the complete calibration procedure may not be required. Refer to *Appendix A: Calibration Options* for more information.

Throughout the procedure, refer to the C/C++ function call parameters for the LabVIEW input values.

Initial Setup

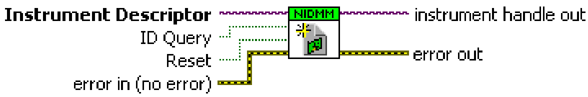
Note This section is necessary only for pre-adjustment verifications. If you are performing a post-adjustment verification, skip the setup and go directly to the *Verifying DC Voltage* section.

To set up the test equipment, complete the following steps:

1. Remove all connections from the four input banana plug connectors on the NI 4071.
2. Verify that the calibrator has been calibrated within the time limits specified in the *Test Equipment* section, and that DC zeros calibration has been performed within the last 30 days. Consult the Fluke 5720A user documentation for instructions for calibrating these devices.

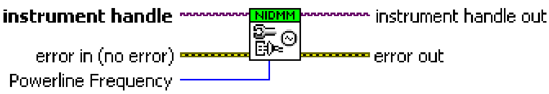
Note Ensure that both the calibrator and the NI 4071 (installed in a powered-on PXI chassis) are warmed up for at least 60 minutes before you begin this procedure.

3. Call the niDMM Initialize VI with the Instrument Descriptor of the device to create an instrument session.

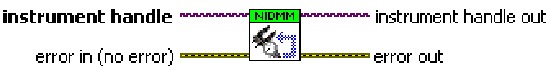
LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_init` with the following parameters: **Instrument_Descriptor**: The name of the device to calibrate. You can find this name under Devices and Interfaces in Measurement & Automation Explorer (MAX) **ID_Query**: `VI_FALSE` **Reset**: `VI_FALSE`

Note You will use this session in all subsequent VI and function calls throughout the verification procedures. For more information about using the niDMM Initialize VI or the `niDMM_init` function, refer to the *NI Digital Multimeters Help*.

4. Call the niDMM Configure Powerline Frequency VI.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_ConfigurePowerLineFrequency` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **PowerLine Frequency**: Set this parameter to `50` or `60`, depending on the powerline frequency (in hertz) powering your instruments; select `50` for 400 Hz powerline frequencies

5. Call the niDMM Self Cal VI.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_SelfCal` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init`

Verification Procedures

You can use the verification procedures described in this section for both pre-adjustment and post-adjustment verification. The verification procedures and the steps within them must be performed in the order listed; however, you can opt to omit entire sections (for example, the entire *Verifying AC Current* section).

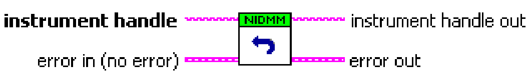
The parameters **Range**, **Resolution**, and **Sample Interval** used in VI and function calls throughout this section have floating point values. For example, if **Range** = `1`, the floating point value is `1.0`. The parameters **Trigger Count**, **Sample Count**, **Array Size**, and **ParamValue** have integer values. Refer to the *NI Digital Multimeters Help* for more information about parameter values.

Note Many of the parameter values listed in this document are expressed in scientific notation. Some programming languages do not support the direct entry of numbers in this format. Be sure to properly enter these values with the appropriate number of zeros. For example, you must enter the scientific notation number 10e–6 as `0.00001` and the number 100e3 as `100000`. If your programming language supports numeric entries in scientific notation, NI recommends that you use this feature to minimize possible data entry errors.

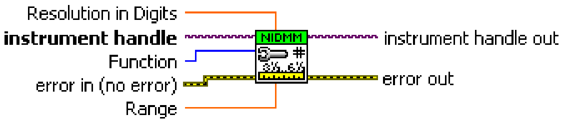
Verifying DC Voltage

To verify DC voltage of the NI 4071, complete the following steps:

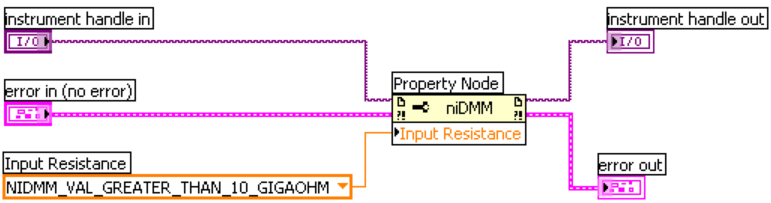
1. Plug in the insulated banana plug shorting bar across the *HI* and *LO* banana plug connectors on the NI 4071.
2. Wait one minute for the thermal EMF to stabilize.
3. Call the niDMM Reset VI.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_reset` with the following parameter: **Instrument_Handle**: The instrument handle from `niDMM_init`

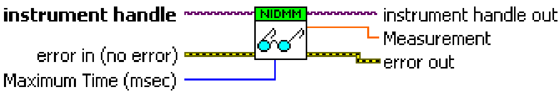
4. Call the niDMM Config Measurement VI and select the **Resolution in Digits** instance.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_ConfigureMeasurementDigits` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Resolution_Digits**: `7.5` **Measurement_Function**: `NIDMM_VAL_DC_VOLTS` **Range**: `100 mV`

5. Use a writable niDMM property node to set the input resistance of the NI 4071 to >10 GΩ.

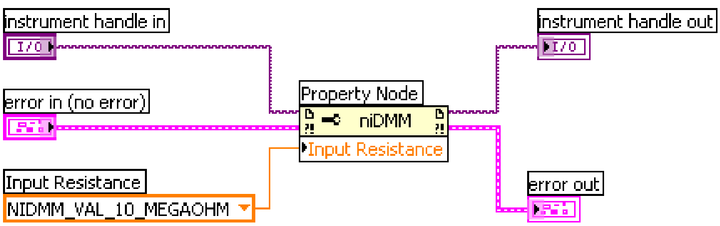
LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM SetAttributeViReal64` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Attribute_ID**: `NIDMM_ATTR_INPUT_RESISTANCE` **Attribute_Value**: `NIDMM_VAL_GREATER_THAN_10_GIGAOHM`

6. Call the niDMM Read VI. Store the reading or measurement value as the 100 mV >10 GΩ mode offset.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_read` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Maximum_Time**: `-1` **Reading**: The measurement returned by the function. Store the reading or measurement value as the 100 mV >10 GΩ mode offset.

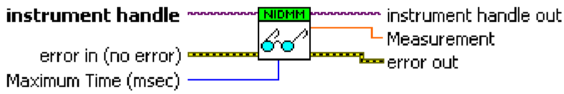
Note The **Measurement** output of the LabVIEW block diagram equates to the **Reading** function of C++ function call.

7. Use a writable niDMM property node to set the input resistance of the NI 4071 to 10 MΩ.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM SetAttributeViReal64` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Attribute_ID**: `NIDMM_ATTR_INPUT_RESISTANCE` **Attribute_Value**: `NIDMM_VAL_10_MEGAOHM`

8. Call the niDMM Read VI.

<table>
<tr><th>LabVIEW Block Diagram</th><th>C/C++ Function Call</th></tr>
<tr><td>



</td><td>

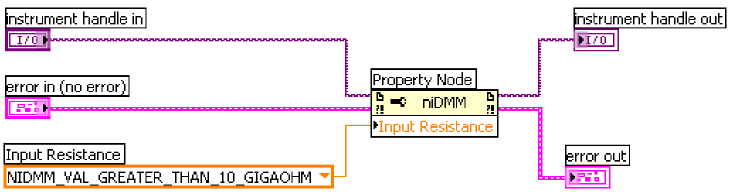
Call `niDMM_read` with the following parameters:

Instrument_Handle: The instrument handle from `niDMM_init`
Reading: The measurement returned by the function. Store the reading or measurement value as the 100 mV 10 MΩ mode offset.
Maximum_Time: `-1`

</td></tr>
</table>

9. Use a writable niDMM property node to set the input resistance of the NI 4071 to >10 GΩ.

<table>
<tr><th>LabVIEW Block Diagram</th><th>C/C++ Function Call</th></tr>
<tr><td>



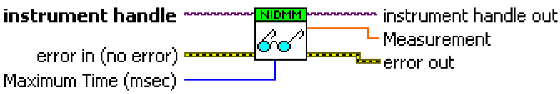
</td><td>

Call `niDMM SetAttributeViReal64` with the following parameters:

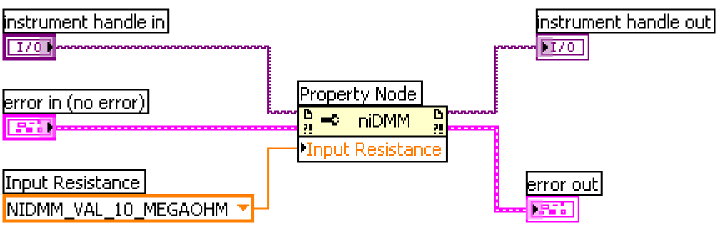
Instrument_Handle: The instrument handle from `niDMM_init`
Attribute_ID: `NIDMM_ATTR_INPUT_RESISTANCE`
Attribute_Value: `NIDMM_VAL_GREATER_THAN_10_GIGAOHM`

</td></tr>
</table>

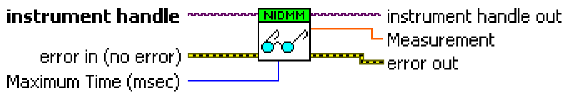
10. Call the niDMM Read VI.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_read` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Reading**: The measurement returned by the function. Subtract the previously stored 100 mV >10 GΩ mode offset value from this measurement. Verify that this measurement falls between the limits listed in Table 20. **Maximum_Time**: `-1`

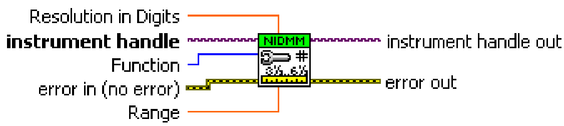
11. Use a writable niDMM property node to set the input resistance of the NI 4071 to 10 MΩ.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM SetAttributeViReal64` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Attribute_ID**: `NIDMM_ATTR_INPUT_ RESISTANCE` **Attribute_Value**: `NIDMM_VAL_10_MEGAOHM`

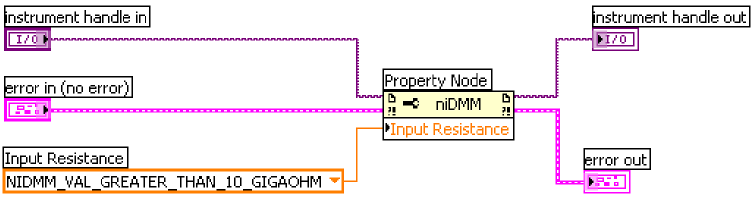
12. Call the niDMM Read VI.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_read` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Reading**: The measurement returned by the function. Subtract the previously stored 100 mV 10 MΩ mode offset value from this measurement.Verify that this measurement falls between the limits listed in Table 20. **Maximum_Time**: `-1`

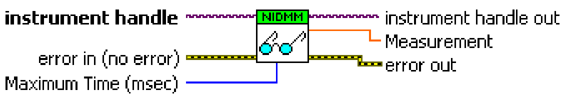
13. Call the niDMM Config Measurement VI and select the **Resolution in Digits** instance.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_ConfigureMeasurementDigits` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Resolution_Digits**: `7.5` **Measurement_Function**: `NIDMM_VAL_DC_VOLTS` **Range**: `1`

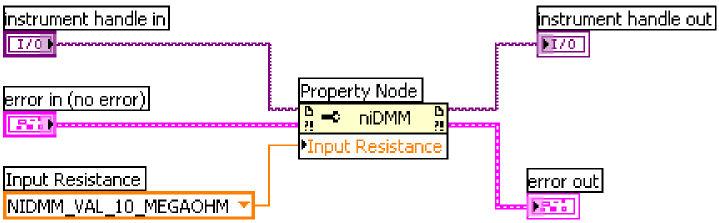
14. Use a writable niDMM property node to set the input resistance of the NI 4071 to >10 GΩ.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM SetAttributeViReal64` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Attribute_ID**: `NIDMM_ATTR_INPUT_RESISTANCE` **Attribute_Value**: `NIDMM_VAL_GREATER_THAN_10_GIGAOHM`

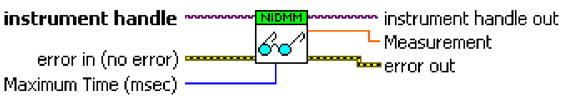
15. Call the niDMM Read VI.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_read` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Reading**: The measurement returned by the function. Verify that this measurement falls between the limits listed in Table 20. **Maximum_Time**: `-1`

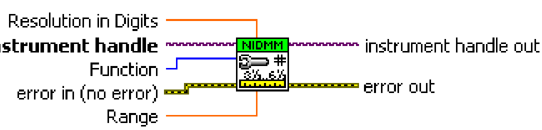
16. Use a writable niDMM property node to set the input resistance of the NI 4071 to 10 MΩ.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM SetAttributeViReal64` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Attribute_ID**: `NIDMM_ATTR_INPUT_ RESISTANCE` **Attribute_Value**: `NIDMM_VAL_10_MEGAOHM`

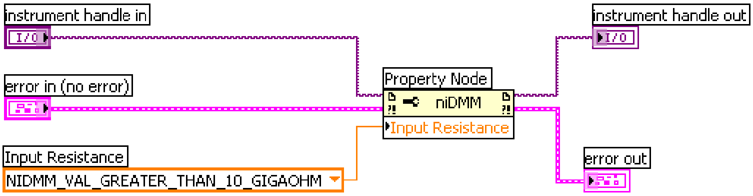
17. Call the niDMM Read VI.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_read` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Reading**: The measurement returned by the function. Verify that this measurement falls between the limits listed in Table 20. **Maximum_Time**: `-1`

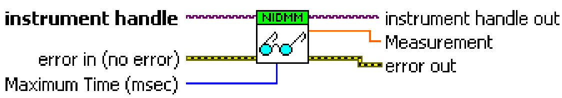
18. Call the niDMM Config Measurement VI and select the **Resolution in Digits** instance.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_ConfigureMeasurement Digits` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Resolution_Digits**: `7.5` **Measurement_Function**: `NIDMM_VAL_DC_VOLTS` **Range**: `10`

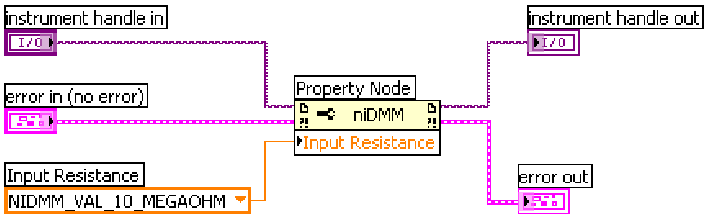
19. Use a writable niDMM property node to set the input resistance of the NI 4071 to >10 GΩ.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM SetAttributeViReal64` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Attribute_ID**: `NIDMM_ATTR_INPUT_RESISTANCE` **Attribute_Value**: `NIDMM_VAL_GREATER_THAN_10_GIGAOHM`

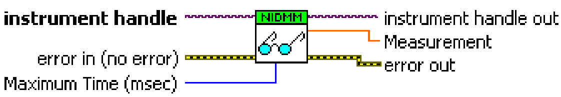
20. Call the niDMM Read VI.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_read` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Reading**: The measurement returned by the function. Verify that this measurement falls between the limits listed in Table 20. **Maximum_Time**: `-1`

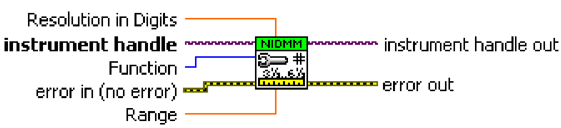
21. Use a writable niDMM property node to set the input resistance of the NI 4071 to 10 MΩ.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM SetAttributeViReal64` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Attribute_ID**: `NIDMM_ATTR_INPUT_ RESISTANCE` **Attribute_Value**: `NIDMM_VAL_10_MEGAOHM`

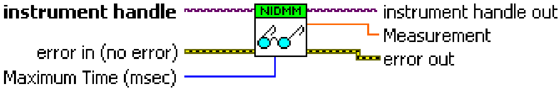
22. Call the niDMM Read VI.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_read` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Reading**: The measurement returned by the function. Verify that this measurement falls between the limits listed in Table 20. **Maximum_Time**: `-1`

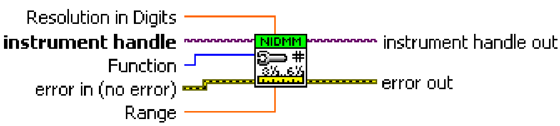
23. Call the niDMM Config Measurement VI and select the **Resolution in Digits** instance.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_ConfigureMeasurement Digits` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Resolution_Digits**: `7.5` **Measurement_Function**: `NIDMM_VAL_DC_VOLTS` **Range**: `100`

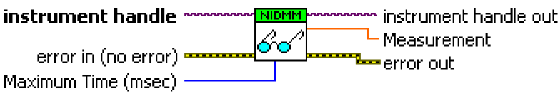
24. Call the niDMM Read VI.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_read` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Reading**: The measurement returned by the function. Verify that this measurement falls between the limits listed in Table 20. **Maximum_Time**: `-1`

25. Call the niDMM Config Measurement VI and select the **Resolution in Digits** instance.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_ConfigureMeasurementDigits` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Resolution_Digits**: `7.5` **Measurement_Function**: `NIDMM_VAL_DC_VOLTS` **Range**: `1000`

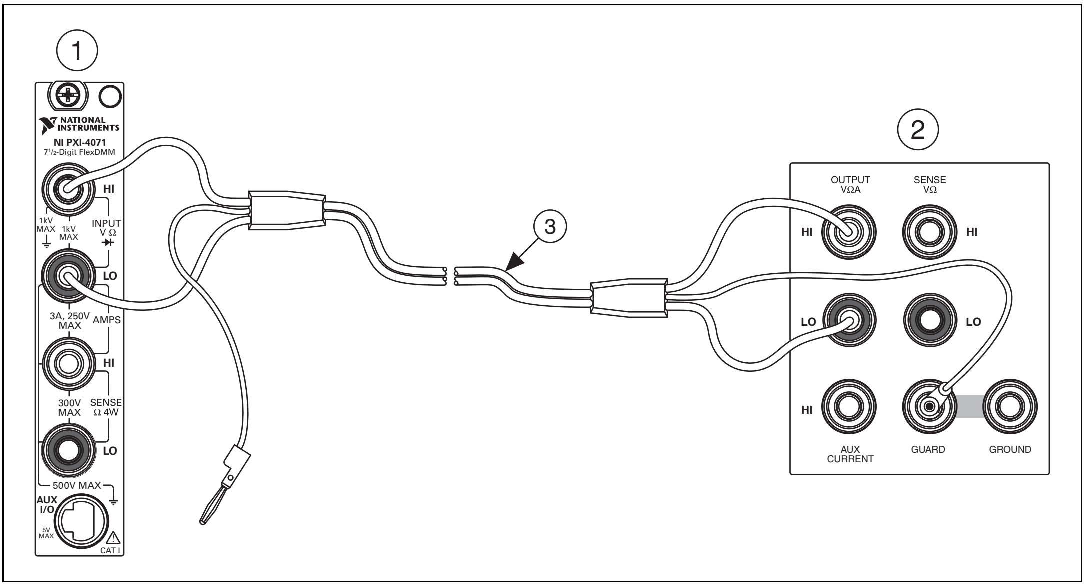
26. Call the niDMM Read VI.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_read` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Reading**: The measurement returned by the function. Verify that this measurement falls between the limits listed in Table 20. **Maximum_Time**: `-1`

27. Remove the shorting bar from the NI 4071.

28. Reset the calibrator.

29. Connect the NI 4071 and the Fluke 5720A calibrator using the Fluke 5440 cable, as shown in Figure 1. Table 2 lists the cable connections.



1	NI 4071	2	Fluke 5720A Calibrator	3	Fluke 5440 Cable

Figure 1. Cable Connections for Voltage and 2-Wire Resistance

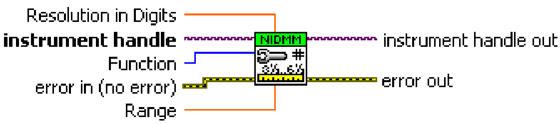
Table 2. Fluke 5440 Cable Connections

Banana Plug Connector (NI 4071)	Banana Plug Color (Fluke 5440 Cable)	Binding Post Label (Fluke 5720A Calibrator)
HI	Red	OUTPUT HI
LO	Black	OUTPUT LO
(No connection)	Blue	V GUARD

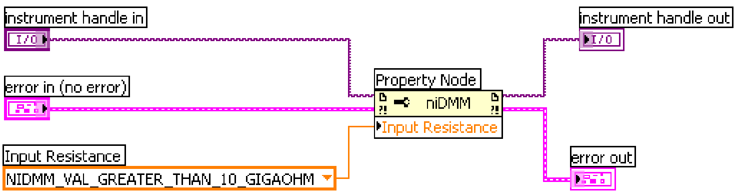
30. Wait two minutes for the thermal EMF to stabilize.
31. Generate 90 mV on the calibrator with the range locked to 2.2 V.

Note This calibrator range prevents a 50 Ω output resistance from creating a voltage divider with the internal resistance of the NI 4071.

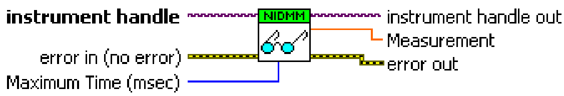
32. Call the niDMM Config Measurement VI and select the **Resolution in Digits** instance.

LabVIEW Block Diagram	**C/C++ Function Call**
	Call `niDMM_ConfigureMeasurementDigits` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Resolution_Digits**: `7.5` **Measurement_Function**: `NIDMM_VAL_DC_VOLTS` **Range**: `0.1`

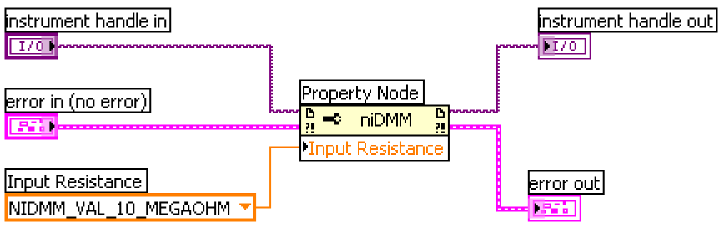
33. Use a writable niDMM property node to set the input resistance of the NI 4071 to >10 GΩ.

LabVIEW Block Diagram	**C/C++ Function Call**
	Call `niDMM SetAttributeViReal64` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Attribute_ID**: `NIDMM_ATTR_INPUT_RESISTANCE` **Attribute_Value**: `NIDMM_VAL_GREATER_THAN_10_GIGAOHM`

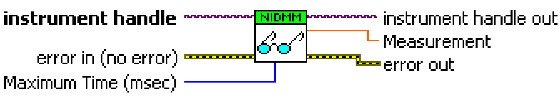
34. Call the niDMM Read VI.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_read` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Reading**: The measurement returned by the function. Subtract the previously stored 100 mV >10 GΩ mode offset value from this measurement. Verify the reading or measurement value falls between the limits listed in Table 20. **Maximum_Time**: `-1`

35. Use a writable niDMM property node to set the input resistance of the NI 4071 to 10 MΩ.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM SetAttributeViReal64` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Attribute_ID**: `NIDMM_ATTR_INPUT_RESISTANCE` **Attribute_Value**: `NIDMM_VAL_10_MEGAOHM`

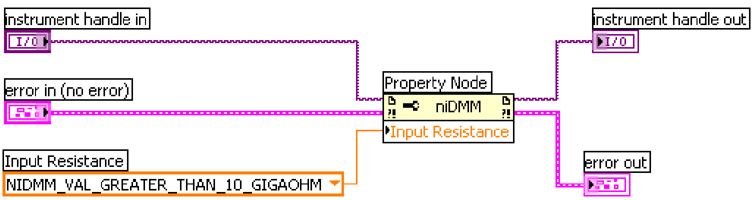
36. Call the niDMM Read VI.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_read` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Reading**: The measurement returned by the function. Subtract the previously stored 100 mV 10 MΩ mode offset value from this measurement. Verify the reading or measurement value falls between the limits listed in Table 20. **Maximum_Time**: `-1`

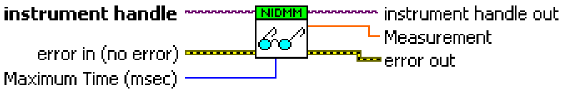
37. Generate –90 mV on the calibrator with the range locked to 2.2 V.

Note This calibrator range prevents the 50 Ω output resistance of the 220 mV range from creating a voltage divider with the internal resistance of the NI 4071.

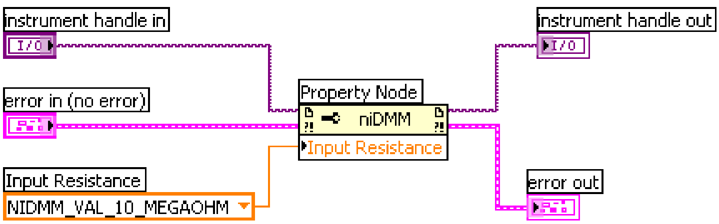
38. Use a writable niDMM property node to set the input resistance of the NI 4071 to >10 GΩ.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM SetAttributeViReal64` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Attribute_ID**: `NIDMM_ATTR_INPUT_RESISTANCE` **Attribute_Value**: `NIDMM_VAL_GREATER_THAN_10_GIGAOHM`

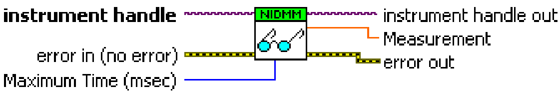
39. Call the niDMM Read VI.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_read` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Reading**: The measurement returned by the function. Subtract the previously stored 100 mV >10 GΩ mode offset value from this measurement. Verify the reading or measurement value falls between the limits listed in Table 20. **Maximum_Time**: `-1`

40. Use a writable niDMM property node to set the input resistance of the NI 4071 to 10 MΩ.

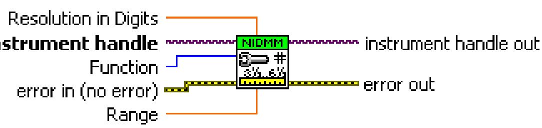
LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM SetAttributeViReal64` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Attribute_ID**: `NIDMM_ATTR_INPUT_ RESISTANCE` **Attribute_Value**: `NIDMM_VAL_10_MEGAOHM`

41. Call the niDMM Read VI.

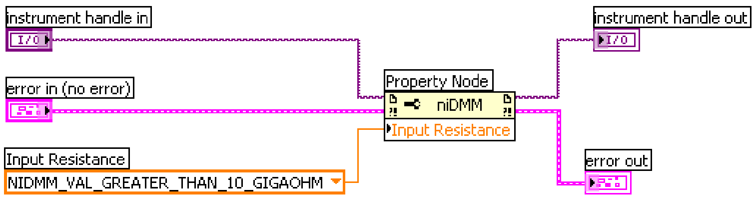
LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_read` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Reading**: The measurement returned by the function. Subtract the previously stored 100 mV 10 MΩ mode offset value from this measurement. Verify that the reading or measurement value falls between the limits listed in Table 20. **Maximum_Time**: `-1`

42. Generate 0.9 V on the calibrator.

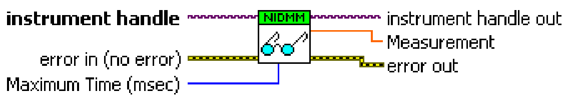
43. Call the niDMM Config Measurement VI and select the **Resolution in Digits** instance.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_ConfigureMeasurementDigits` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Resolution_Digits**: `7.5` **Measurement_Function**: `NIDMM_VAL_DC_VOLTS` **Range**: `1`

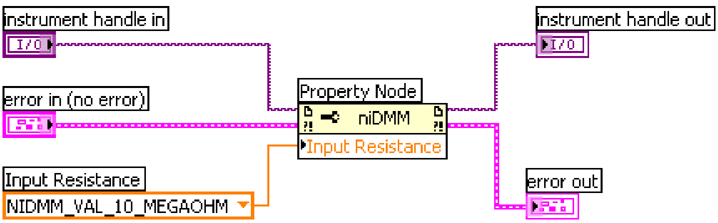
44. Use a writable niDMM property node to set the input resistance of the NI 4071 to >10 GΩ.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM SetAttributeViReal64` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Attribute_ID**: `NIDMM_ATTR_INPUT_RESISTANCE` **Attribute_Value**: `NIDMM_VAL_GREATER_THAN_10_GIGAOHM`

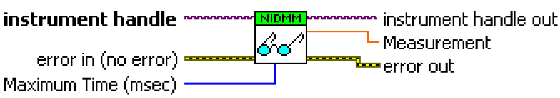
45. Call the niDMM Read VI.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_read` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Reading**: The measurement returned by the function. Verify that this measurement falls between the limits listed in Table 20. **Maximum_Time**: `-1`

46. Use a writable niDMM property node to set the input resistance of the NI 4071 to 10 MΩ.

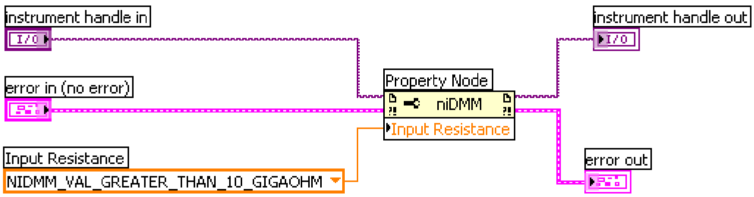
LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM SetAttributeViReal64` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Attribute_ID**: `NIDMM_ATTR_INPUT_RESISTANCE` **Attribute_Value**: `NIDMM_VAL_10_MEGAOHM`

47. Call the niDMM Read VI.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_read` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Reading**: The measurement returned by the function. Verify that this measurement falls between the limits listed in Table 20. **Maximum_Time**: `-1`

48. Generate –0.9 V on the calibrator.

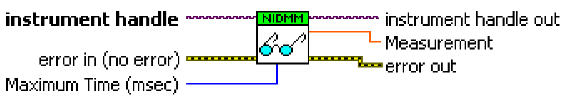
49. Use a writable niDMM property node to set the input resistance of the NI 4071 to >10 GΩ.

<table>
<tr><th>LabVIEW Block Diagram</th><th>C/C++ Function Call</th></tr>
<tr><td>


</td><td>
Call <code>niDMM SetAttributeViReal64</code> with the following parameters:

Instrument_Handle: The instrument handle from <code>niDMM_init</code>
Attribute_ID: <code>NIDMM_ATTR_INPUT_RESISTANCE</code>
Attribute_Value: <code>NIDMM_VAL_GREATER_THAN_10_GIGAOHM</code>
</td></tr>
</table>

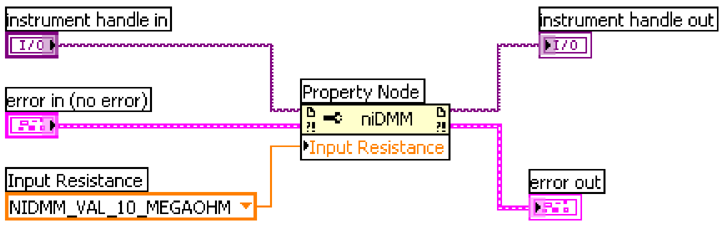
50. Call the niDMM Read VI.

<table>
<tr><th>LabVIEW Block Diagram</th><th>C/C++ Function Call</th></tr>
<tr><td>


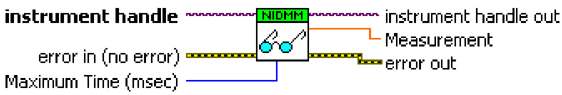
</td><td>
Call <code>niDMM_read</code> with the following parameters:

Instrument_Handle: The instrument handle from <code>niDMM_init</code>
Reading: The measurement returned by the function. Verify that this measurement falls between the limits listed in Table 20.
Maximum_Time: <code>-1</code>
</td></tr>
</table>

51. Use a writable niDMM property node to set the input resistance of the NI 4071 to 10 MΩ.

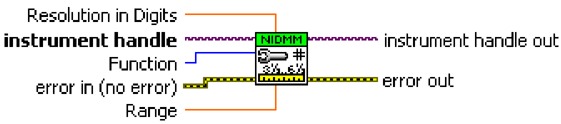
LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM SetAttributeViReal64` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Attribute_ID**: `NIDMM_ATTR_INPUT_ RESISTANCE` **Attribute_Value**: `NIDMM_VAL_10_MEGAOHM`

52. Call the niDMM Read VI.

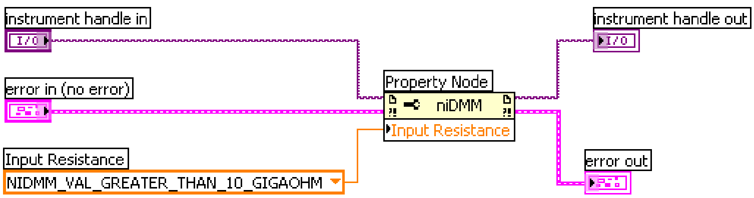
LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_read` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Reading**: The measurement returned by the function. Verify that this measurement falls between the limits listed in Table 20. **Maximum_Time**: `-1`

53. Generate 9 V on the calibrator.

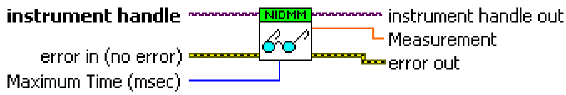
54. Call the niDMM Config Measurement VI and select the **Resolution in Digits** instance.Call the niDMM Config Measurement VI and select the **Resolution in Digits** instance.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_ConfigureMeasurementDigits` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Resolution_Digits**: `7.5` **Measurement_Function**: `NIDMM_VAL_DC_VOLTS` **Range**: `10`

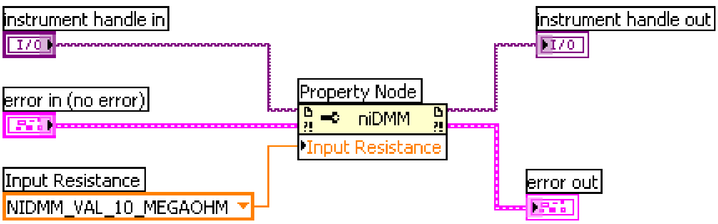
55. Use a writable niDMM property node to set the input resistance of the NI 4071 to >10 GΩ.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM SetAttributeViReal64` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Attribute_ID**: `NIDMM_ATTR_INPUT_RESISTANCE` **Attribute_Value**: `NIDMM_VAL_GREATER_THAN_10_GIGAOHM`

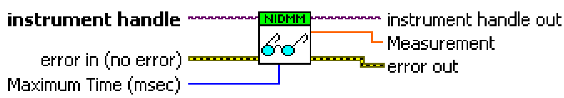
56. Call the niDMM Read VI.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_read` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Reading**: The measurement returned by the function. Verify that this measurement falls between the limits listed in Table 20. **Maximum_Time**: `-1`

57. Use a writable niDMM property node to set the input resistance of the NI 4071 to 10 MΩ.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM SetAttributeViReal64` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Attribute_ID**: `NIDMM_ATTR_INPUT_ RESISTANCE` **Attribute_Value**: `NIDMM_VAL_10_MEGAOHM`

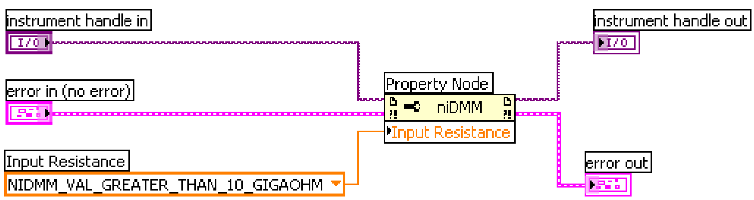
58. Call the niDMM Read VI.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_read` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Reading**: The measurement returned by the function. Verify that this measurement falls between the limits listed in Table 20. **Maximum_Time**: `-1`

59. Refer to Table 3 for the appropriate calibrator outputs and parameter values as you complete the following steps:

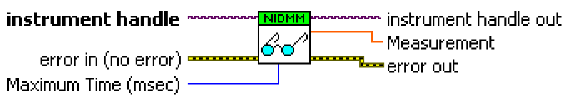
    a. On the calibrator, generate the value listed under Calibrator Output in Table 3 for the current iteration.

    b. Use a writable niDMM property node to set the input resistance of the NI 4071 to the value shown in Table 3 for the current iteration.

<table>
<tr><th>LabVIEW Block Diagram</th><th>C/C++ Function Call</th></tr>
<tr><td>
</td><td>Call <code>niDMM SetAttributeViReal64</code> with the following parameters:

Instrument_Handle: The instrument handle from <code>niDMM_init</code>
Attribute_ID: <code>NIDMM_ATTR_INPUT_RESISTANCE</code>
Attribute_Value: <code>NIDMM_VAL_GREATER_THAN_10_GIGAOHM</code></td></tr>
</table>

    c. Call the niDMM Read VI.

<table>
<tr><th>LabVIEW Block Diagram</th><th>C/C++ Function Call</th></tr>
<tr><td>
</td><td>Call <code>niDMM_read</code> with the following parameters:

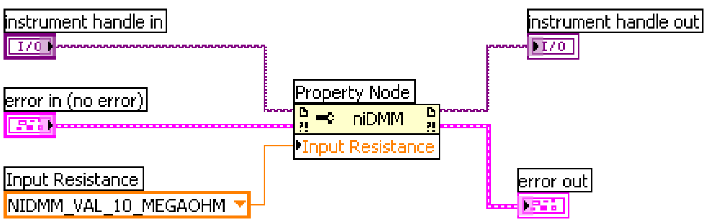
Instrument_Handle: The instrument handle from <code>niDMM_init</code>
Reading: The measurement returned by the function. Verify that this measurement falls between the limits listed in Table 20.
Maximum_Time: <code>-1</code></td></tr>
</table>

60. Repeat step 59 for each of the remaining iterations shown in Table 3.

Table 3. niDMM 10 V Linearity Settings

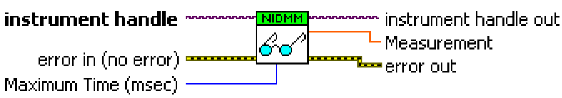
Iteration	Calibrator Output	Input Resistance
1	5 V	>10 GΩ
2	2.5 V	>10 GΩ
3	–2.5 V	>10 GΩ
4	–5 V	>10 GΩ
5	–9 V	>10 GΩ

61. With the calibrator still set at –9 V, use a writable niDMM property node to set the input resistance of the NI 4071 to 10 MΩ.

<table>
<tr><th>LabVIEW Block Diagram</th><th>C/C++ Function Call</th></tr>
<tr><td>
</td><td>Call <code>niDMM SetAttributeViReal64</code> with the following parameters:

Instrument_Handle: The instrument handle from <code>niDMM_init</code>
Attribute_ID: <code>NIDMM_ATTR_INPUT_ RESISTANCE</code>
Attribute_Value: <code>NIDMM_VAL_10_MEGAOHM</code></td></tr>
</table>

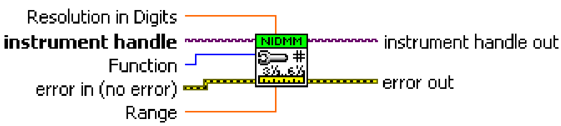
62. Call the niDMM Read VI.

<table>
<tr><th>LabVIEW Block Diagram</th><th>C/C++ Function Call</th></tr>
<tr><td>
</td><td>Call <code>niDMM_read</code> with the following parameters:

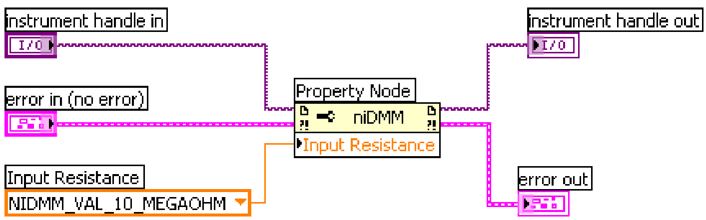
Instrument_Handle: The instrument handle from <code>niDMM_init</code>
Reading: The measurement returned by the function. Verify that this measurement falls between the limits listed in Table 20.
Maximum_Time: <code>-1</code></td></tr>
</table>

Caution Avoid touching the connections when generating a high voltage from the calibrator.

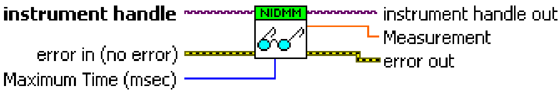
63. Generate 90 V on the calibrator.
64. Call the niDMM Config Measurement VI and select the **Resolution in Digits** instance.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_ConfigureMeasurementDigits` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Resolution_Digits**: `7.5` **Measurement_Function**: `NIDMM_VAL_DC_VOLTS` **Range**: `100`

65. Use a writable niDMM property node to set the input resistance of the NI 4071 to 10 MΩ.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM SetAttributeViReal64` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Attribute_ID**: `NIDMM_ATTR_INPUT_RESISTANCE` **Attribute_Value**: `NIDMM_VAL_10_MEGAOHM`

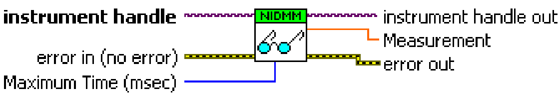
66. Call the niDMM Read VI.

<table>
<tr><th>LabVIEW Block Diagram</th><th>C/C++ Function Call</th></tr>
<tr><td>
</td><td>Call <code>niDMM_read</code> with the following parameters:

Instrument_Handle: The instrument handle from <code>niDMM_init</code>
Reading: The measurement returned by the function. Verify that this measurement falls between the limits listed in Table 20.
Maximum_Time: <code>-1</code></td></tr>
</table>

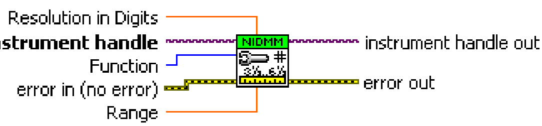
67. Output –90 V on the calibrator.

68. Call the niDMM Read VI.

<table>
<tr><th>LabVIEW Block Diagram</th><th>C/C++ Function Call</th></tr>
<tr><td>
</td><td>Call <code>niDMM_read</code> with the following parameters:

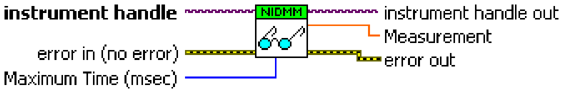
Instrument_Handle: The instrument handle from <code>niDMM_init</code>
Reading: The measurement returned by the function. Verify that this measurement falls between the limits listed in Table 20.
Maximum_Time: <code>-1</code></td></tr>
</table>

69. Call the niDMM Config Measurement VI and select the **Resolution in Digits** instance.

<table>
<tr><th>LabVIEW Block Diagram</th><th>C/C++ Function Call</th></tr>
<tr><td>
</td><td>Call <code>niDMM_ConfigureMeasurementDigits</code> with the following parameters:

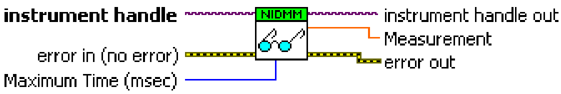
Instrument_Handle: The instrument handle from <code>niDMM_init</code>
Resolution_Digits: <code>7.5</code>
Measurement_Function: <code>NIDMM_VAL_DC_VOLTS</code>
Range: <code>1000</code></td></tr>
</table>

70. Call the niDMM Read VI.

<table>
<tr><th>LabVIEW Block Diagram</th><th>C/C++ Function Call</th></tr>
<tr><td></td><td>Call <code>niDMM_read</code> with the following parameters:

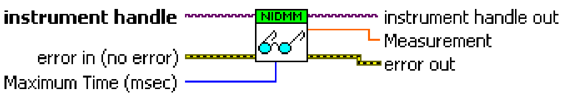
Instrument_Handle: The instrument handle from <code>niDMM_init</code>
Reading: The measurement returned by the function. The DMM must be in the 1000 V range before you apply the voltage.
Maximum_Time: <code>-1</code></td></tr>
</table>

71. Generate 1000 V on the calibrator.
72. Call the niDMM Read VI.

<table>
<tr><th>LabVIEW Block Diagram</th><th>C/C++ Function Call</th></tr>
<tr><td></td><td>Call <code>niDMM_read</code> with the following parameters:

Instrument_Handle: The instrument handle from <code>niDMM_init</code>
Reading: The measurement returned by the function. Verify that this measurement falls between the limits listed in Table 20.
Maximum_Time: <code>-1</code></td></tr>
</table>

73. Generate –1000 V on the calibrator.
74. Call the niDMM Read VI.

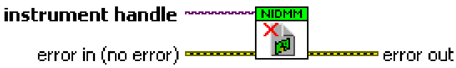
<table>
<tr><th>LabVIEW Block Diagram</th><th>C/C++ Function Call</th></tr>
<tr><td></td><td>Call <code>niDMM_read</code> with the following parameters:

Instrument_Handle: The instrument handle from <code>niDMM_init</code>
Reading: The measurement returned by the function. Verify that this measurement falls between the limits listed in Table 20.
Maximum_Time: <code>-1</code></td></tr>
</table>

75. Reset the calibrator for safety reasons.

You have completed verifying the DC voltage of the NI 4071. Select one of the following options:

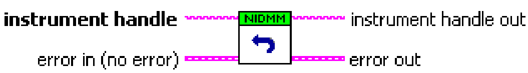
- If you want to continue verifying other modes, go to the *Verifying AC Voltage* section.
- If you do *not* want to verify other modes *and* you are performing a *post-adjustment* verification, go to the *Completing the Adjustment Procedures* section.
- If you do *not* want to verify any additional modes *and* you are performing a *pre-adjustment* verification, call the niDMM Close VI to close the session.

LabVIEW Block Diagram	**C/C++ Function Call**
	Call `niDMM_close` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init`

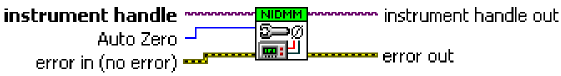
Verifying AC Voltage

To verify the AC voltage of the NI 4071, complete the following steps:

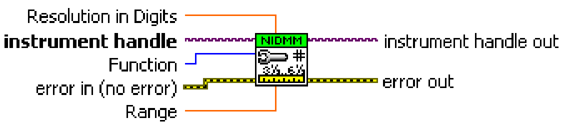
1. Reset the calibrator.
2. Connect the NI 4071 and the Fluke 5720A calibrator using the Fluke 5440 cable, as shown in Figure 1. Table 2 lists the cable connections.
3. Generate 4.5 mV at 1 kHz on the calibrator.
4. Call the niDMM Reset VI to reset the NI 4071 to a known state.

LabVIEW Block Diagram	**C/C++ Function Call**
	Call `niDMM_reset` with the following parameter: **Instrument_Handle**: The instrument handle from `niDMM_init`

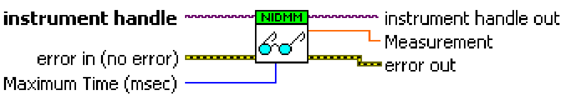
5. Call the niDMM Configure Auto Zero VI.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_ConfigureAutoZeroMode` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Auto_Zero_Mode**: `NIDMM_VAL_AUTO_ZERO_ON`

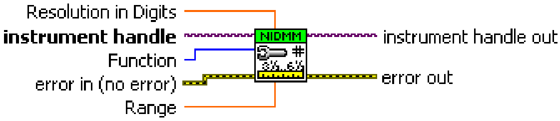
6. Call the niDMM Config Measurement VI and select the **Resolution in Digits** instance.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_ConfigureMeasurementDigits` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Resolution_Digits**: `6.5` **Measurement_Function**: `NIDMM_VAL_AC_VOLTS` **Range**: `0.05`

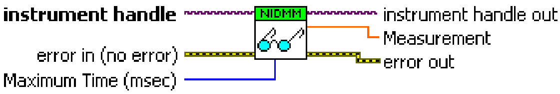
7. Call the niDMM Read VI.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_read` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Reading**: The measurement returned by the function. Verify that this measurement falls between the limits listed in Table 21. **Maximum_Time**: `-1`

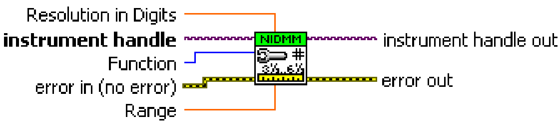
8. Call the niDMM Config Measurement VI and select the **Resolution in Digits** instance.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_ConfigureMeasurementDigits` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Resolution_Digits**: `6.5` **Measurement_Function**: `NIDMM_VAL_AC_VOLTS_DCCOUPLED` **Range**: `0.05`

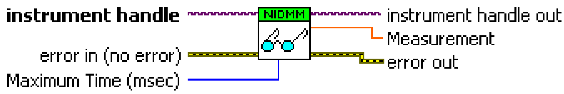
9. Call the niDMM Read VI.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_read` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Reading**: The measurement returned by the function. Verify that this measurement falls between the limits listed in Table 21. **Maximum_Time**: `-1`

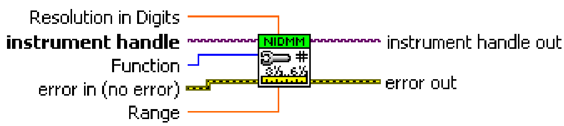
10. Generate 45 mV at 30 Hz on the calibrator.
11. Call the niDMM Config Measurement VI and select the **Resolution in Digits** instance.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_ConfigureMeasurementDigits` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Resolution_Digits**: `6.5` **Measurement_Function**: `NIDMM_VAL_AC_VOLTS_DCCOUPLED` **Range**: `0.05`

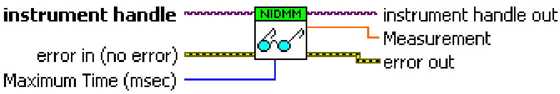
12. Call the niDMM Read VI.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_read` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Reading**: The measurement returned by the function. Verify that this measurement falls between the limits listed in Table 21. **Maximum_Time**: `-1`

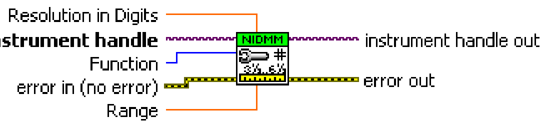
13. Refer to Table 4 for the appropriate calibrator output and measurement parameter values as you complete the following steps:
    a. On the calibrator, generate the value listed under Calibrator Output in Table 4 for the current iteration.
    b. Call the niDMM Config Measurement VI with **Function** set to `NIDMM_VAL_AC_VOLTS`, and set the remaining parameters as shown in Table 4 for the current iteration. Call the niDMM Config Measurement VI and select the **Resolution in Digits** instance.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_ConfigureMeasurement Digits` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Resolution_Digits**: Set as shown in Table 4 for the current iteration **Measurement_Function**: `NIDMM_VAL_AC_VOLTS` **Range**: Set as shown in Table 4 for the current iteration

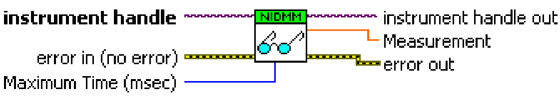
c. Call the niDMM Read VI.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_read` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Reading**: The measurement returned by the function. Verify that this measurement falls between the limits listed in Table 21. **Maximum_Time**: `-1`

d. Call the niDMM Config Measurement VI again, changing **Function** to `NIDMM_VAL_AC_VOLTS_DCCOUPLED`, and select the **Resolution in Digits** instance.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_ConfigureMeasurementDigits` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Resolution_Digits**: Set as shown in Table 4 for the current iteration **Measurement_Function**: `NIDMM_VAL_AC_VOLTS_DCCOUPLED` **Range**: Set as shown in Table 4 for the current iteration

e. Call the niDMM Read VI.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_read` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Reading**: The measurement returned by the function. Verify that this measurement falls between the limits listed in Table 21. **Maximum_Time**: `-1`

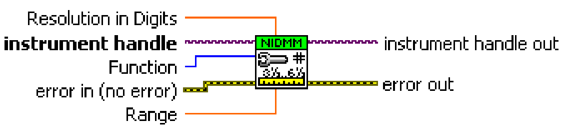
14. Repeat step 13 for each of the remaining iterations shown in Table 4.

Table 4. niDMM Config Measurement Parameters

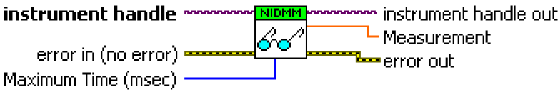
Iteration	Calibrator Output Signal Parameters		niDMM Config Measurement Parameters		
	Amplitude	**Frequency**	**Function**	**Range**	**Resolution**
1	45 mV	50 Hz	`NIDMM_VAL_AC_VOLTS`	`0.05`	`6.5`
	45 mV	50 Hz	`NIDMM_VAL_AC_VOLTS_DCCOUPLED`	`0.05`	`6.5`
2	45 mV	1 kHz	`NIDMM_VAL_AC_VOLTS`	`0.05`	`6.5`
	45 mV	1 kHz	`NIDMM_VAL_AC_VOLTS_DCCOUPLED`	`0.05`	`6.5`
3	45 mV	1 kHz	`NIDMM_VAL_AC_VOLTS`	`0.5`	`6.5`
	45 mV	1 kHz	`NIDMM_VAL_AC_VOLTS_DCCOUPLED`	`0.5`	`6.5`
4	45 mV	20 kHz	`NIDMM_VAL_AC_VOLTS`	`0.05`	`6.5`
	45 mV	20 kHz	`NIDMM_VAL_AC_VOLTS_DCCOUPLED`	`0.05`	`6.5`
5	45 mV	50 kHz	`NIDMM_VAL_AC_VOLTS`	`0.05`	`6.5`
	45 mV	50 kHz	`NIDMM_VAL_AC_VOLTS_DCCOUPLED`	`0.05`	`6.5`
6	45 mV	100 kHz	`NIDMM_VAL_AC_VOLTS`	`0.05`	`6.5`
	45 mV	100 kHz	`NIDMM_VAL_AC_VOLTS_DCCOUPLED`	`0.05`	`6.5`
7	45 mV	300 kHz	`NIDMM_VAL_AC_VOLTS`	`0.05`	`6.5`
	45 mV	300 kHz	`NIDMM_VAL_AC_VOLTS_DCCOUPLED`	`0.05`	`6.5`

15. Generate 450 mV at 30 Hz on the calibrator.

16. Call the niDMM Config Measurement VI and select the **Resolution in Digits** instance.

LabVIEW Block Diagram	C/C++ Function Call
 	Call `niDMM_ConfigureMeasurementDigits` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Resolution_Digits**: `6.5` **Measurement_Function**: `NIDMM_VAL_AC_VOLTS_DCCOUPLED` **Range**: `0.5`

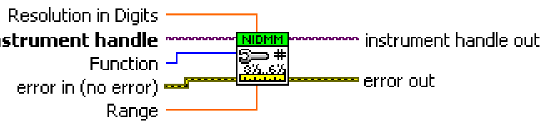
17. Call the niDMM Read VI.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_read` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Reading**: The measurement returned by the function. Verify that this measurement falls between the limits listed in Table 21. **Maximum_Time**: `-1`

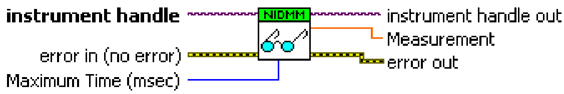
18. Refer to Table 5 for the appropriate calibrator output and measurement parameter values as you complete the following steps:

    a. On the calibrator, generate the value listed under Calibrator Output in Table 5 for the current iteration.

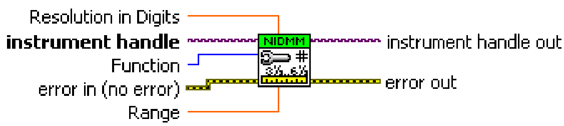
    b. Call the niDMM Config Measurement VI with **Function** set to `NIDMM_VAL_AC_VOLTS`, and set the remaining parameters as shown in Table 5 for the current iteration. Select the **Resolution in Digits** instance of the niDMM Config Measurement VI.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_ConfigureMeasurementDigits` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Resolution_Digits**: Set as shown in Table 5 for the current iteration **Measurement_Function**: `NIDMM_VAL_AC_VOLTS` **Range**: Set as shown in Table 5 for the current iteration

c. Call the niDMM Read VI.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_read` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Reading**: The measurement returned by the function. Verify that this measurement falls between the limits listed in Table 21. **Maximum_Time**: `-1`

d. Call the niDMM Config Measurement VI again, changing **Function** to `NIDMM_VAL_AC_VOLTS_DCCOUPLED`, and select the **Resolution in Digits** instance.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_ConfigureMeasurementDigits` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Resolution_Digits**: Set as shown in Table 5 for the current iteration **Measurement_Function**: `NIDMM_VAL_AC_VOLTS_DCCOUPLED` **Range**: Set as shown in Table 5 for the current iteration

e. Call the niDMM Read VI.

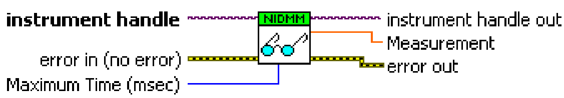
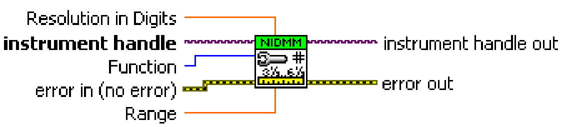
LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_read` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Reading**: The measurement returned by the function. Verify that this measurement falls between the limits listed in Table 21. **Maximum_Time**: `-1`

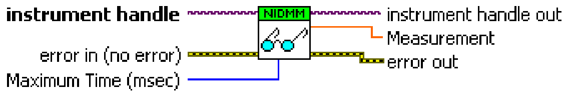
Table 5. niDMM Config Measurement Parameters

Iteration	Calibrator Output Signal Parameters		niDMM Config Measurement Parameters		
	Amplitude	Frequency	Function	Range	Resolution
1	450 mV	50 Hz	`NIDMM_VAL_AC_VOLTS`	`0.5`	`6.5`
	450 mV	50 Hz	`NIDMM_VAL_AC_VOLTS_DCCOUPLED`	`0.5`	`6.5`
2	450 mV	1 kHz	`NIDMM_VAL_AC_VOLTS`	`0.5`	`6.5`
	450 mV	1 kHz	`NIDMM_VAL_AC_VOLTS_DCCOUPLED`	`0.5`	`6.5`
3	450 mV	1 kHz	`NIDMM_VAL_AC_VOLTS`	`5`	`6.5`
	450 mV	1 kHz	`NIDMM_VAL_AC_VOLTS_DCCOUPLED`	`5`	`6.5`
4	450 mV	20 kHz	`NIDMM_VAL_AC_VOLTS`	`0.5`	`6.5`
	450 mV	20 kHz	`NIDMM_VAL_AC_VOLTS_DCCOUPLED`	`0.5`	`6.5`
5	450 mV	50 kHz	`NIDMM_VAL_AC_VOLTS`	`0.5`	`6.5`
	450 mV	50 kHz	`NIDMM_VAL_AC_VOLTS_DCCOUPLED`	`0.5`	`6.5`
6	450 mV	100 kHz	`NIDMM_VAL_AC_VOLTS`	`0.5`	`6.5`
	450 mV	100 kHz	`NIDMM_VAL_AC_VOLTS_DCCOUPLED`	`0.5`	`6.5`
7	450 mV	300 kHz	`NIDMM_VAL_AC_VOLTS`	`0.5`	`6.5`
	450 mV	300 kHz	`NIDMM_VAL_AC_VOLTS_DCCOUPLED`	`0.5`	`6.5`

19. Generate 4.5 V at 30 Hz on the calibrator.
20. Call the niDMM Config Measurement VI and select the **Resolution in Digits** instance.

LabVIEW Block Diagram	C/C++ Function Call
 	Call `niDMM_ConfigureMeasurementDigits` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Resolution_Digits**: `6.5` **Measurement_Function**: `NIDMM_VAL_AC_VOLTS_DCCOUPLED` **Range**: `5`

21. Call the niDMM Read VI.

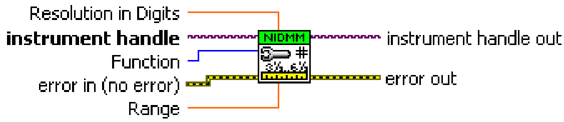
<table>
<tr><th>LabVIEW Block Diagram</th><th>C/C++ Function Call</th></tr>
<tr><td>
</td><td>Call <code>niDMM_read</code> with the following parameters:

Instrument_Handle: The instrument handle from <code>niDMM_init</code>
Reading: The measurement returned by the function. Verify that this measurement falls between the limits listed in Table 21.
Maximum_Time: <code>-1</code></td></tr>
</table>

22. Refer to Table 6 for the appropriate calibrator outputs and parameter values as you complete the following steps:

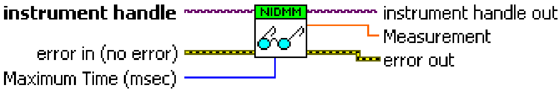
   a. On the calibrator, generate the value listed under Calibrator Output in Table 6 for the current iteration.

   b. Call the niDMM Config Measurement VI with **Function** set to `NIDMM_VAL_AC_VOLTS`, and set the remaining parameters as shown in Table 6 for the current iteration. Select the **Resolution in Digits** instance of the niDMM Config Measurement VI.

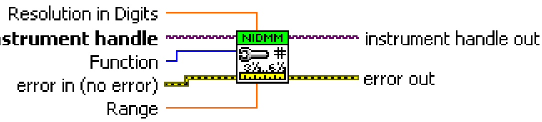
<table>
<tr><th>LabVIEW Block Diagram</th><th>C/C++ Function Call</th></tr>
<tr><td>
</td><td>Call <code>niDMM_ConfigureMeasurement Digits</code> with the following parameters:

Instrument_Handle: The instrument handle from <code>niDMM_init</code>
Resolution_Digits: Set as shown in Table 6 for the current iteration
Measurement_Function: <code>NIDMM_VAL_AC_VOLTS</code>
Range: Set as shown in Table 6 for the current iteration</td></tr>
</table>

c. Call the niDMM Read VI.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_read` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Reading**: The measurement returned by the function. Verify that this measurement falls between the limits listed in Table 21. **Maximum_Time**: `-1`

d. Call the niDMM Config Measurement VI again, changing **Function** to `NIDMM_VAL_AC_VOLTS_DCCOUPLED`, and select the **Resolution in Digits** instance.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_ConfigureMeasurement Digits` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Resolution_Digits**: Set as shown in Table 6 for the current iteration **Measurement_Function**: `NIDMM_VAL_AC_VOLTS_DCCOUPLED` **Range**: Set as shown in Table 6 for the current iteration

e. Call the niDMM Read VI.

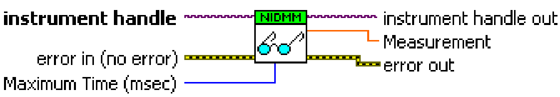
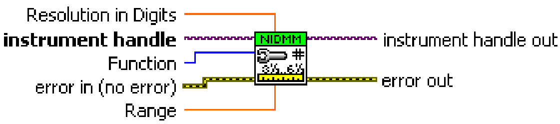
LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_read` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Reading**: The measurement returned by the function. Verify that this measurement falls between the limits listed in Table 21. **Maximum_Time**: `-1`

Table 6. niDMM Config Measurement Parameters

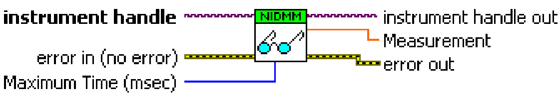
Iteration	Calibrator Output Signal Parameters		niDMM Config Measurement Parameters		
	Amplitude	Frequency	Function	Range	Resolution
1	4.5 V	50 Hz	`NIDMM_VAL_AC_VOLTS`	`5`	`6.5`
	4.5 V	50 Hz	`NIDMM_VAL_AC_VOLTS_DCCOUPLED`	`5`	`6.5`
2	1 V	1 kHz	`NIDMM_VAL_AC_VOLTS`	`5`	`6.5`
	1 V	1 kHz	`NIDMM_VAL_AC_VOLTS_DCCOUPLED`	`5`	`6.5`
3	2 V	1 kHz	`NIDMM_VAL_AC_VOLTS`	`5`	`6.5`
	2 V	1 kHz	`NIDMM_VAL_AC_VOLTS_DCCOUPLED`	`5`	`6.5`
4	3 V	1 kHz	`NIDMM_VAL_AC_VOLTS`	`5`	`6.5`
	3 V	1 kHz	`NIDMM_VAL_AC_VOLTS_DCCOUPLED`	`5`	`6.5`
5	4.5 V	1 kHz	`NIDMM_VAL_AC_VOLTS`	`5`	`6.5`
	4.5 V	1 kHz	`NIDMM_VAL_AC_VOLTS_DCCOUPLED`	`5`	`6.5`
6	4.5 V	1 kHz	`NIDMM_VAL_AC_VOLTS`	`50`	`6.5`
	4.5 V	1 kHz	`NIDMM_VAL_AC_VOLTS_DCCOUPLED`	`50`	`6.5`
7	4.5 V	20 kHz	`NIDMM_VAL_AC_VOLTS`	`5`	`6.5`
	4.5 V	20 kHz	`NIDMM_VAL_AC_VOLTS_DCCOUPLED`	`5`	`6.5`
8	4.5 V	50 kHz	`NIDMM_VAL_AC_VOLTS`	`5`	`6.5`
	4.5 V	50 kHz	`NIDMM_VAL_AC_VOLTS_DCCOUPLED`	`5`	`6.5`
9	4.5 V	100 kHz	`NIDMM_VAL_AC_VOLTS`	`5`	`6.5`
	4.5 V	100 kHz	`NIDMM_VAL_AC_VOLTS_DCCOUPLED`	`5`	`6.5`
10	4.5 V	300 kHz	`NIDMM_VAL_AC_VOLTS`	`5`	`6.5`
	4.5 V	300 kHz	`NIDMM_VAL_AC_VOLTS_DCCOUPLED`	`5`	`6.5`

23. Generate 45 V at 30 Hz on the calibrator.

24. Call the niDMM Config Measurement VI and select the **Resolution in Digits** instance.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_ConfigureMeasurement Digits` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Resolution_Digits**: `6.5` **Measurement_Function**: `NIDMM_VAL_AC_VOLTS_DCCOUPLED` **Range**: `50`

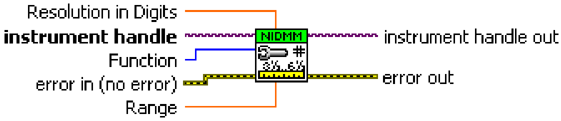
25. Call the niDMM Read VI.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_read` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Reading**: The measurement returned by the function. Verify that this measurement falls between the limits listed in Table 21. **Maximum_Time**: `-1`

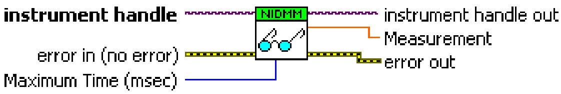
26. Refer to Table 7 for the appropriate calibrator outputs and parameter values as you complete the following steps:

    a. On the calibrator, generate the value listed under Calibrator Output in Table 7 for the current iteration.

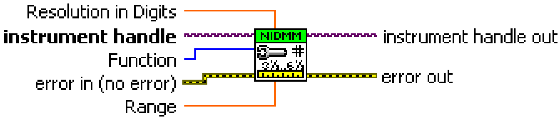
    b. Call the niDMM Config Measurement VI with **Function** set to `NIDMM_VAL_AC_VOLTS`, and set the remaining parameters as shown in Table 7 for the current iteration. Select the **Resolution in Digits** instance of the niDMM Config Measurement VI.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_ConfigureMeasurementDigits` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Resolution_Digits**: Set as shown in Table 7 for the current iteration **Measurement_Function**: `NIDMM_VAL_AC_VOLTS` **Range**: Set as shown in Table 7 for the current iteration

    c. Call the niDMM Read VI.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_read` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Reading**: The measurement returned by the function. Verify that this measurement falls between the limits listed in Table 21. **Maximum_Time**: `-1`

d. Call the niDMM Config Measurement VI again, changing **Function** to `NIDMM_VAL_AC_VOLTS_DCCOUPLED`, and select the **Resolution in Digits** instance.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_ConfigureMeasurementDigits` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Resolution_Digits**: Set as shown in Table 7 for the current iteration **Measurement_Function**: `NIDMM_VAL_AC_VOLTS_DCCOUPLED` **Range**: Set as shown in Table 7 for the current iteration

e. Call the niDMM Read VI.

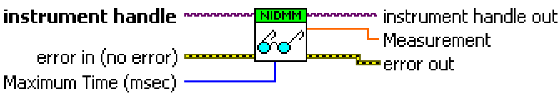
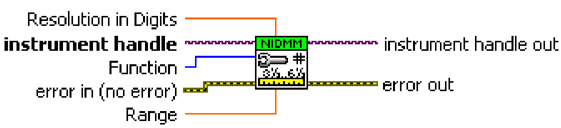
LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_read` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Reading**: The measurement returned by the function. Verify that this measurement falls between the limits listed in Table 21. **Maximum_Time**: `-1`

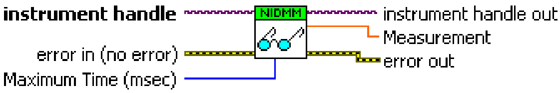
Table 7. niDMM Config Measurement Parameters

Iteration	Calibrator Output Signal Parameters		niDMM Config Measurement Parameters		
	Amplitude	Frequency	Function	Range	Resolution
1	45 V	50 Hz	`NIDMM_VAL_AC_VOLTS`	`50`	`6.5`
	45 V	50 Hz	`NIDMM_VAL_AC_VOLTS_DCCOUPLED`	`50`	`6.5`
2	45 V	1 kHz	`NIDMM_VAL_AC_VOLTS`	`50`	`6.5`
	45 V	1 kHz	`NIDMM_VAL_AC_VOLTS_DCCOUPLED`	`50`	`6.5`
3	45 V	1 kHz	`NIDMM_VAL_AC_VOLTS`	`700`	`6.5`
	45 V	1 kHz	`NIDMM_VAL_AC_VOLTS_DCCOUPLED`	`700`	`6.5`
4	45 V	20 kHz	`NIDMM_VAL_AC_VOLTS`	`50`	`6.5`
	45 V	20 kHz	`NIDMM_VAL_AC_VOLTS_DCCOUPLED`	`50`	`6.5`
5	45 V	50 kHz	`NIDMM_VAL_AC_VOLTS`	`50`	`6.5`
	45 V	50 kHz	`NIDMM_VAL_AC_VOLTS_DCCOUPLED`	`50`	`6.5`
6	45 V	100 kHz	`NIDMM_VAL_AC_VOLTS`	`50`	`6.5`
	45 V	100 kHz	`NIDMM_VAL_AC_VOLTS_DCCOUPLED`	`50`	`6.5`
7	45 V	300 kHz	`NIDMM_VAL_AC_VOLTS`	`50`	`6.5`
	45 V	300 kHz	`NIDMM_VAL_AC_VOLTS_DCCOUPLED`	`50`	`6.5`

27. Call the niDMM Config Measurement VI and select the **Resolution in Digits** instance.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_ConfigureMeasurementDigits` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Resolution_Digits**: `6.5` **Measurement_Function**: `NIDMM_VAL_AC_VOLTS_DCCOUPLED` **Range**: `700`

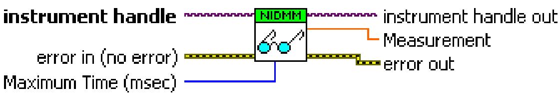
28. Call the niDMM Read VI.

<table>
<tr><th>LabVIEW Block Diagram</th><th>C/C++ Function Call</th></tr>
<tr><td>
</td><td>Call <code>niDMM_read</code> with the following parameters:

Instrument_Handle: The instrument handle from <code>niDMM_init</code>
Reading: The measurement returned by the function. The DMM must be in the 700 V range before you apply the voltage.
Maximum_Time: <code>-1</code></td></tr>
</table>

29. Generate 200 V at 30 Hz on the calibrator.

30. Call the niDMM Read VI.

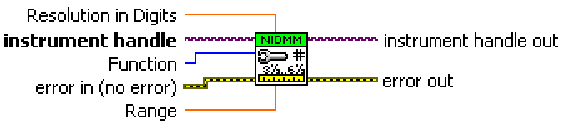
<table>
<tr><th>LabVIEW Block Diagram</th><th>C/C++ Function Call</th></tr>
<tr><td>
</td><td>Call <code>niDMM_read</code> with the following parameters:

Instrument_Handle: The instrument handle from <code>niDMM_init</code>
Reading: The measurement returned by the function. Verify that this measurement falls between the limits listed in Table 21.
Maximum_Time: <code>-1</code></td></tr>
</table>

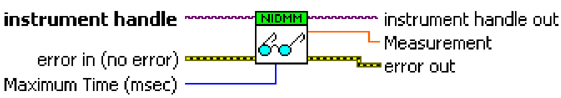
31. Refer to Table 8 for the appropriate calibrator output signal and measurement parameter values as you complete the following steps:

    a. On the calibrator, generate the value listed under Calibrator Output in Table 8 for the current iteration.

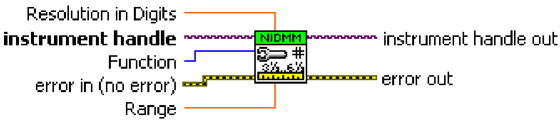
    b. Call the niDMM Config Measurement VI with **Function** set to `NIDMM_VAL_AC_VOLTS`, and set the remaining parameters as shown in Table 8 for the current iteration. Select the **Resolution in Digits** instance of the niDMM Config Measurement VI.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_ConfigureMeasurementDigits` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Resolution_Digits**: Set as shown in Table 8 for the current iteration **Measurement_Function**: `NIDMM_VAL_AC_VOLTS` **Range**: Set as shown in Table 8 for the current iteration

    c. Call the niDMM Read VI.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_read` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Reading**: The measurement returned by the function. Verify that this measurement falls between the limits listed in Table 21. **Maximum_Time**: `-1`

d. Call the niDMM Config Measurement VI again, changing **Function** to `NIDMM_VAL_AC_VOLTS_DCCOUPLED`, and select the **Resolution in Digits** instance.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_ConfigureMeasurementDigits` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Resolution_Digits**: Set as shown in Table 8 for the current iteration **Measurement_Function**: `NIDMM_VAL_AC_VOLTS_DCCOUPLED` **Range**: Set as shown in Table 8 for the current iteration

e. Call the niDMM Read VI.

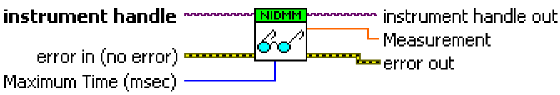
LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_read` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Reading**: The measurement returned by the function. Verify that this measurement falls between the limits listed in Table 21. **Maximum_Time**: `-1`

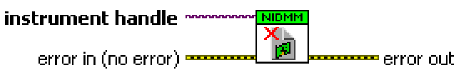
Table 8. niDMM Config Measurement Parameters

Iteration	Calibrator Output Signal Parameters		niDMM Config Measurement Parameters		
	Amplitude	Frequency	Function	Range	Resolution
1	630 V	50 Hz	NIDMM_VAL_AC_VOLTS	700	6.5
	630 V	50 Hz	NIDMM_VAL_AC_VOLTS_DCCOUPLED	700	6.5
2	630 V	1 kHz	NIDMM_VAL_AC_VOLTS	700	6.5
	630 V	1 kHz	NIDMM_VAL_AC_VOLTS_DCCOUPLED	700	6.5
3	200 V	20 kHz	NIDMM_VAL_AC_VOLTS	700	6.5
	200 V	20 kHz	NIDMM_VAL_AC_VOLTS_DCCOUPLED	700	6.5
4	200 V	50 kHz	NIDMM_VAL_AC_VOLTS	700	6.5
	200 V	50 kHz	NIDMM_VAL_AC_VOLTS_DCCOUPLED	700	6.5
5	200 V	100 kHz	NIDMM_VAL_AC_VOLTS	700	6.5
	200 V	100 kHz	NIDMM_VAL_AC_VOLTS_DCCOUPLED	700	6.5
6	70 V	300 kHz	NIDMM_VAL_AC_VOLTS	700	6.5
	70 V	300 kHz	NIDMM_VAL_AC_VOLTS_DCCOUPLED	700	6.5

32. Reset the calibrator for safety reasons.

You have completed verifying the AC voltage of the NI 4071. Select one of the following options:

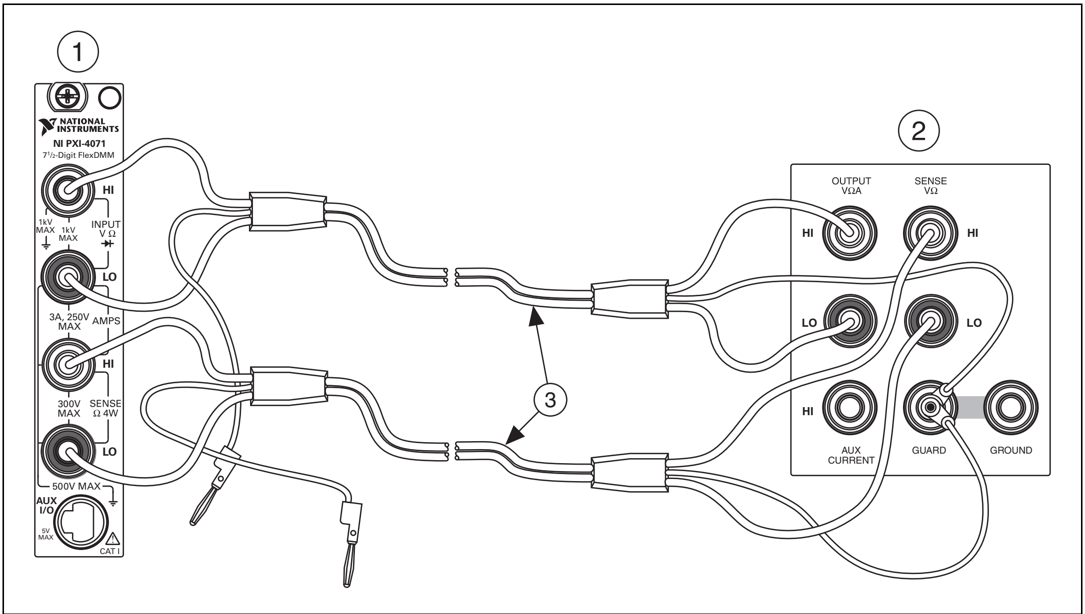
- If you want to continue verifying other modes, go to the *Verifying 4-Wire Resistance* section.
- If you do *not* want to verify other modes *and* you are performing a *post-adjustment* verification, go to the *Completing the Adjustment Procedures* section.
- If you do *not* want to verify any additional modes *and* you are performing a *pre-adjustment* verification, call the niDMM Close VI to close the session.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_close` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init`

Verifying 4-Wire Resistance

To verify the 4-wire resistance of the NI 4071, complete the following steps:

1. Reset the calibrator.
2. Connect the NI 4071 and the Fluke 5720A calibrator using the Fluke 5440 cables, as shown in Figure 2. Table 9 lists the cable connections.



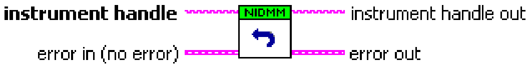
1 NI 4071	2 Fluke 5720A Calibrator	3 Fluke 5440 Cables

Figure 2. Cable Connections for 4-Wire Resistance

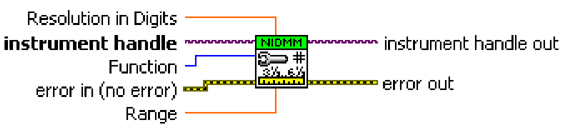
Table 9. Fluke 5440 Cable Connections

Fluke 5440 Cable Identification	Banana Plug Connector (NI 4071)	Banana Plug Color (Fluke 5440 Cable)	Binding Post (Fluke 5720A Calibrator)
First cable	HI	Red	OUTPUT HI
	LO	Black	OUTPUT LO
	(No connection)	Blue	V GUARD
Second cable	HI SENSE	Red	SENSE HI
	LO SENSE	Black	SENSE LO
	(No connection)	Blue	V GUARD

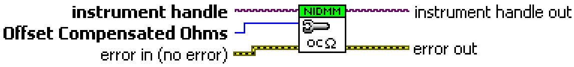
3. Wait two minutes for the thermal EMF to stabilize if the Fluke 5440 cables were not previously connected in this configuration.
4. Call the niDMM Reset VI.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_reset` with the following parameter: **Instrument_Handle**: The instrument handle from `niDMM_init`

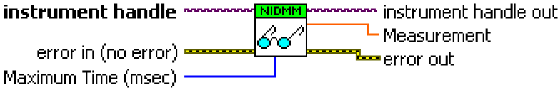
5. Refer to Table 10 for the appropriate calibrator output signal and measurement function parameter values as you complete the following steps:
   a. On the calibrator, generate the value listed in the Calibrator Output column in Table 10 for the current iteration. Make sure that the external sense is turned on, but 2-wire compensation is turned off.
   b. Call the niDMM Config Measurement VI, with the parameters set as shown in Table 10 for the current iteration, and select the **Resolution in Digits** instance.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_ConfigureMeasurementDigits` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Resolution_Digits**: Set as shown in Table 10 for the current iteration **Measurement_Function**: Set as shown in Table 10 for the current iteration **Range**: Set as shown in Table 10 for the current iteration

c. Call the niDMM Configure Offset Comp Ohms VI with **Offset Compensated Ohms** set to either `NIDMM_VAL_OFFSET_COMP_OHMS_ON` or `NIDMM_VAL_OFFSET_COMP_OHMS_OFF` according to Table 10 for the current iteration.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_ConfigureOffsetCompOhms` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **OffsetCompOhms**: Set to either `NIDMM_VAL_OFFSET_COMP_OHMS_ON` or `NIDMM_VAL_OFFSET_COMP_OHMS_OFF` according to Table 10 for the current iteration

d. Call the niDMM Read VI. Verify that this measurement falls between the tolerances listed in Table 23. Tolerances are provided instead of absolute limits, because your calibrator will have different discrete resistance values.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_read` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Reading**: The measurement returned by the function. Verify that this measurement falls between the limits listed in Table 23. **Maximum_Time**: `-1`

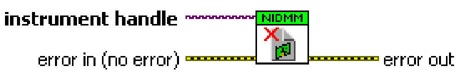
6. Repeat step 5 for each of the remaining iterations listed in Table 10.

Table 10. niDMM Config Measurement Parameters

Iteration	Calibrator Output	niDMM Config Measurement Parameters			
		Function	Range	Resolution	OffsetCompOhms
1	10 MΩ	`NIDMM_VAL_4_WIRE_RES`	`10e6`	`7.5`	OFF
2	1 MΩ	`NIDMM_VAL_4_WIRE_RES`	`1e6`	`7.5`	OFF
3	100 kΩ	`NIDMM_VAL_4_WIRE_RES`	`100e3`	`7.5`	OFF
4	10 kΩ	`NIDMM_VAL_4_WIRE_RES`	`10e3`	`7.5`	ON
5	1 kΩ	`NIDMM_VAL_4_WIRE_RES`	`1e3`	`7.5`	ON
6	100 Ω	`NIDMM_VAL_4_WIRE_RES`	`100`	`7.5`	ON
7	0 Ω	`NIDMM_VAL_4_WIRE_RES`	`10e6`	`7.5`	OFF
8	0 Ω	`NIDMM_VAL_4_WIRE_RES`	`1e6`	`7.5`	OFF
9	0 Ω	`NIDMM_VAL_4_WIRE_RES`	`100e3`	`7.5`	OFF
10	0 Ω	`NIDMM_VAL_4_WIRE_RES`	`10e3`	`7.5`	ON
11	0 Ω	`NIDMM_VAL_4_WIRE_RES`	`1e3`	`7.5`	ON
12	0 Ω	`NIDMM_VAL_4_WIRE_RES`	`100`	`7.5`	ON

You have completed verifying the 4-wire resistance of the NI 4071. Select one of the following options:

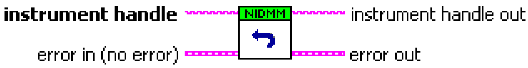
- If you want to continue verifying other modes, go to the *Verifying 2-Wire Resistance* section.
- If you do *not* want to verify other modes *and* you are performing a *post-adjustment* verification, go to the *Completing the Adjustment Procedures* section.
- If you do *not* want to verify any additional modes *and* you are performing a *pre-adjustment* verification, call the niDMM Close VI to close the session.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_close` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init`

Verifying 2-Wire Resistance

To verify the 2-wire resistance of the NI 4071, complete the following steps:

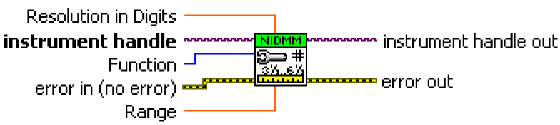
1. Plug in the insulated banana plug shorting bar across the *HI* and *LO* banana plug connectors on the NI 4071.
2. Wait one minute for the thermal EMF to stabilize.
3. Call the niDMM Reset VI.

<table>
<tr><th>LabVIEW Block Diagram</th><th>C/C++ Function Call</th></tr>
<tr><td>


</td><td>
Call <code>niDMM_reset</code> with the following parameter:

Instrument_Handle: The instrument handle from `niDMM_init`
</td></tr>
</table>

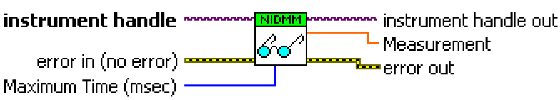
4. Call the niDMM Config Measurement VI and select the **Resolution in Digits** instance.

<table>
<tr><th>LabVIEW Block Diagram</th><th>C/C++ Function Call</th></tr>
<tr><td>


</td><td>
Call <code>niDMM_ConfigureMeasurementDigits</code> with the following parameters:

Instrument_Handle: The instrument handle from `niDMM_init`
Resolution_Digits: `7.5`
Measurement_Function: `NIDMM_VAL_2_WIRE_RES`
Range: `100e6`
</td></tr>
</table>

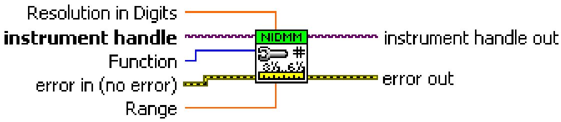
5. Call the niDMM Read VI.

<table>
<tr><th>LabVIEW Block Diagram</th><th>C/C++ Function Call</th></tr>
<tr><td>


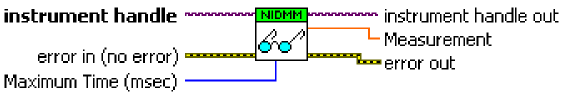
</td><td>
Call <code>niDMM_read</code> with the following parameters:

Instrument_Handle: The instrument handle from `niDMM_init`
Reading: The measurement returned by the function. Verify that this measurement falls between the limits listed in Table 24.
Maximum_Time: `-1`
</td></tr>
</table>

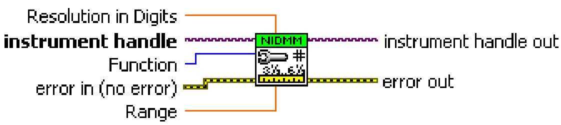
6. Call the niDMM Config Measurement VI and select the **Resolution in Digits** instance.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_ConfigureMeasurementDigits` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Resolution_Digits**: `7.5` **Measurement_Function**: `NIDMM_VAL_2_WIRE_RES` **Range**: `10e6`

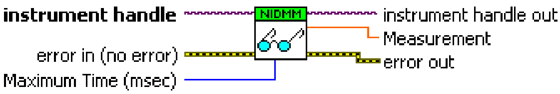
7. Call the niDMM Read VI.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_read` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Reading**: The measurement returned by the function. Verify that this measurement falls between the limits listed in Table 24. **Maximum_Time**: `-1`

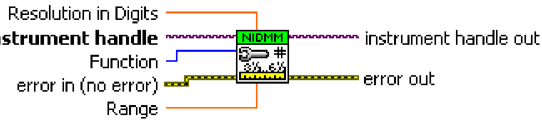
8. Call the niDMM Config Measurement VI and select the **Resolution in Digits** instance.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_ConfigureMeasurementDigits` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Resolution_Digits**: `7.5` **Measurement_Function**: `NIDMM_VAL_2_WIRE_RES` **Range**: `1e6`

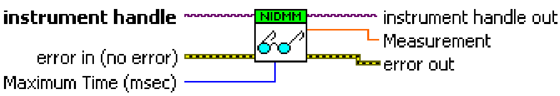
9. Call the niDMM Read VI.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_read` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Reading**: The measurement returned by the function. Verify that this measurement falls between the limits listed in Table 24. **Maximum_Time**: `-1`

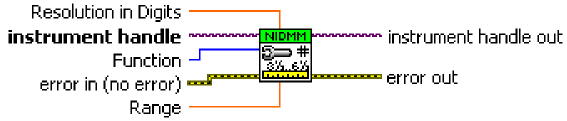
10. Call the niDMM Config Measurement VI and select the **Resolution in Digits** instance.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_ConfigureMeasurementDigits` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Resolution_Digits**: `7.5` **Measurement_Function**: `NIDMM_VAL_2_WIRE_RES` **Range**: `100e3`

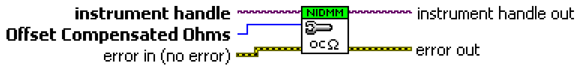
11. Call the niDMM Read VI.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_read` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Reading**: The measurement returned by the function. Verify that this measurement falls between the limits listed in Table 24. **Maximum_Time**: `-1`

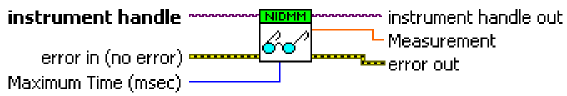
12. Call the niDMM Config Measurement VI and select the **Resolution in Digits** instance.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_ConfigureMeasurementDigits` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Resolution_Digits**: `7.5` **Measurement_Function**: `NIDMM_VAL_2_WIRE_RES` **Range**: `10e3`

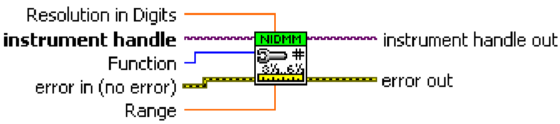
13. Call the niDMM Configure Offset Comp Ohms VI with **Offset Compensated Ohms** set to `NIDMM_VAL_OFFSET_COMP_OHMS_ON`.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_ConfigureOffsetCompOhms` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **OffsetCompOhms**: `NIDMM_VAL_OFFSET_COMP_OHMS_ON`

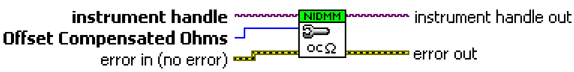
14. Call the niDMM Read VI. Verify that this measurement falls between the tolerances listed in Table 24.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_read` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Reading**: The measurement returned by the function. Verify that this measurement falls between the limits listed in Table 24. **Maximum_Time**: `-1`

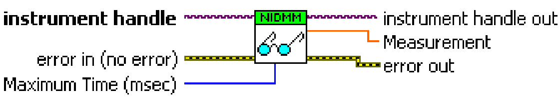
15. Call the niDMM Config Measurement VI and select the **Resolution in Digits** instance.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_ConfigureMeasurementDigits` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Resolution_Digits**: `7.5` **Measurement_Function**: `NIDMM_VAL_2_WIRE_RES` **Range**: `1e3`

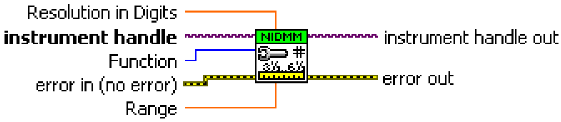
16. Call the niDMM Configure Offset Comp Ohms VI with **Offset Compensated Ohms** set to `NIDMM_VAL_OFFSET_COMP_OHMS_ON`.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_ConfigureOffsetCompOhms` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **OffsetCompOhms**: `NIDMM_VAL_OFFSET_COMP_OHMS_ON`

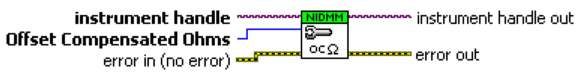
17. Call the niDMM Read VI.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_read` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Reading**: The measurement returned by the function. Verify that this measurement falls between the limits listed in Table 24. **Maximum_Time**: `-1`

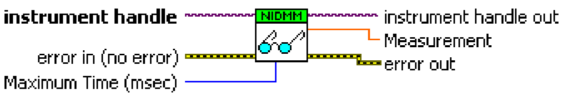
18. Call the niDMM Config Measurement VI and select the **Resolution in Digits** instance.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_ConfigureMeasurementDigits` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Resolution_Digits**: `7.5` **Measurement_Function**: `NIDMM_VAL_2_WIRE_RES` **Range**: `100`

19. Call the niDMM Configure Offset Comp Ohms VI with **Offset Compensated Ohms** set to `NIDMM_VAL_OFFSET_COMP_OHMS_ON`.

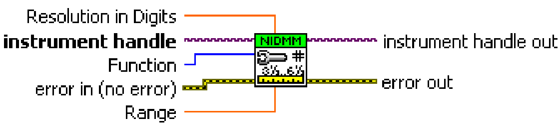
LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_ConfigureOffsetCompOhms` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **OffsetCompOhms**: `NIDMM_VAL_OFFSET_COMP_OHMS_ON`

20. Call the niDMM Read VI.

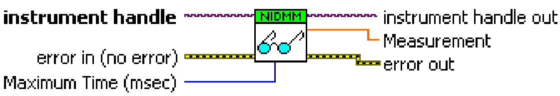
LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_read` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Reading**: The measurement returned by the function. Verify that this measurement falls between the limits listed in Table 24. **Maximum_Time**: `-1`

21. Remove the shorting bar from the NI 4071.
22. Reset the calibrator.

23. Connect the NI 4071 and the Fluke 5720A calibrator using the Fluke 5440 cable, as shown in Figure 1. Table 2 lists the cable connections.
24. Wait two minutes for the thermal EMF to stabilize if the Fluke 5440 cables were not last used in this configuration.
25. Generate 100 MΩ on the calibrator without external sense or 2-wire compensation.
26. Call the niDMM Config Measurement VI and select the **Resolution in Digits** instance.

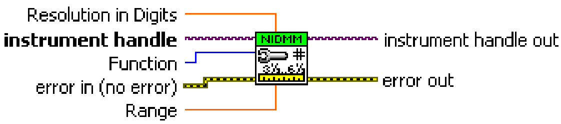
LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_ConfigureMeasurementDigits` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Resolution_Digits**: `7.5` **Measurement_Function**: `NIDMM_VAL_2_WIRE_RES` **Range**: `100e6`

27. Call the niDMM Read VI.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_read` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Reading**: The measurement returned by the function. Verify that this measurement falls between the tolerances listed in Table 24. **Maximum_Time**: `-1`

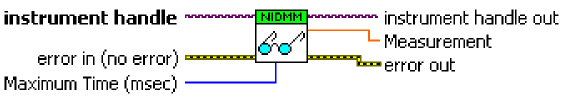
28. Generate 10 MΩ on the calibrator without external sense or 2-wire compensation.

29. Call the niDMM Config Measurement VI and select the **Resolution in Digits** instance.

<table>
<tr><th>LabVIEW Block Diagram</th><th>C/C++ Function Call</th></tr>
<tr><td>
</td><td>Call <code>niDMM_ConfigureMeasurementDigits</code> with the following parameters:

Instrument_Handle: The instrument handle from <code>niDMM_init</code>
Resolution_Digits: <code>7.5</code>
Measurement_Function: <code>NIDMM_VAL_2_WIRE_RES</code>
Range: <code>10e6</code></td></tr>
</table>

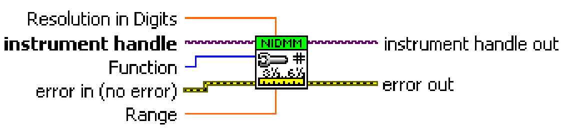
30. Call the niDMM Read VI.

<table>
<tr><th>LabVIEW Block Diagram</th><th>C/C++ Function Call</th></tr>
<tr><td>
</td><td>Call <code>niDMM_read</code> with the following parameters:

Instrument_Handle: The instrument handle from <code>niDMM_init</code>
Reading: The measurement returned by the function. Verify that the reading or measurement value falls between the tolerances listed in Table 24.
Maximum_Time: <code>-1</code></td></tr>
</table>

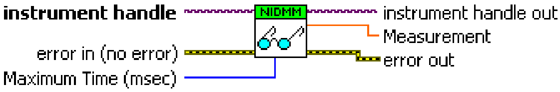
31. Generate 1 MΩ on the calibrator without external sense or 2-wire compensation.

32. Call the niDMM Config Measurement VI and select the **Resolution in Digits** instance.

<table>
<tr><th>LabVIEW Block Diagram</th><th>C/C++ Function Call</th></tr>
<tr><td>
</td><td>Call <code>niDMM_ConfigureMeasurementDigits</code> with the following parameters:

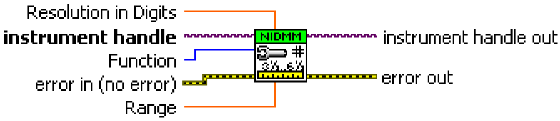
Instrument_Handle: The instrument handle from <code>niDMM_init</code>
Resolution_Digits: <code>7.5</code>
Measurement_Function: <code>NIDMM_VAL_2_WIRE_RES</code>
Range: <code>1e6</code></td></tr>
</table>

33. Call the niDMM Read VI.

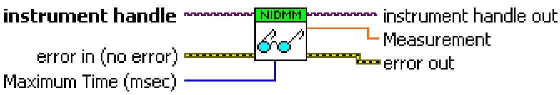
LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_read` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Reading**: The measurement returned by the function. Verify that the reading or measurement value falls between the tolerances listed in Table 24. **Maximum_Time**: `-1`

34. Generate 100 kΩ on the calibrator with external sense turned on, but 2-wire compensation turned off.

35. Call the niDMM Config Measurement VI and select the **Resolution in Digits** instance.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_ConfigureMeasurementDigits` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Resolution_Digits**: `7.5` **Measurement_Function**: `NIDMM_VAL_2_WIRE_RES` **Range**: `100e3`

36. Call the niDMM Read VI.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_read` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Reading**: The measurement returned by the function. Verify that the reading or measurement value falls between the tolerances listed in Table 24. **Maximum_Time**: `-1`

37. Connect the NI 4071 and the Fluke 5720A calibrator using two sets of Fluke 5440 cables, a double banana plug, and two insulated, low EMF spade lugs, as shown in Figure 3. Table 12 lists the cable connections.

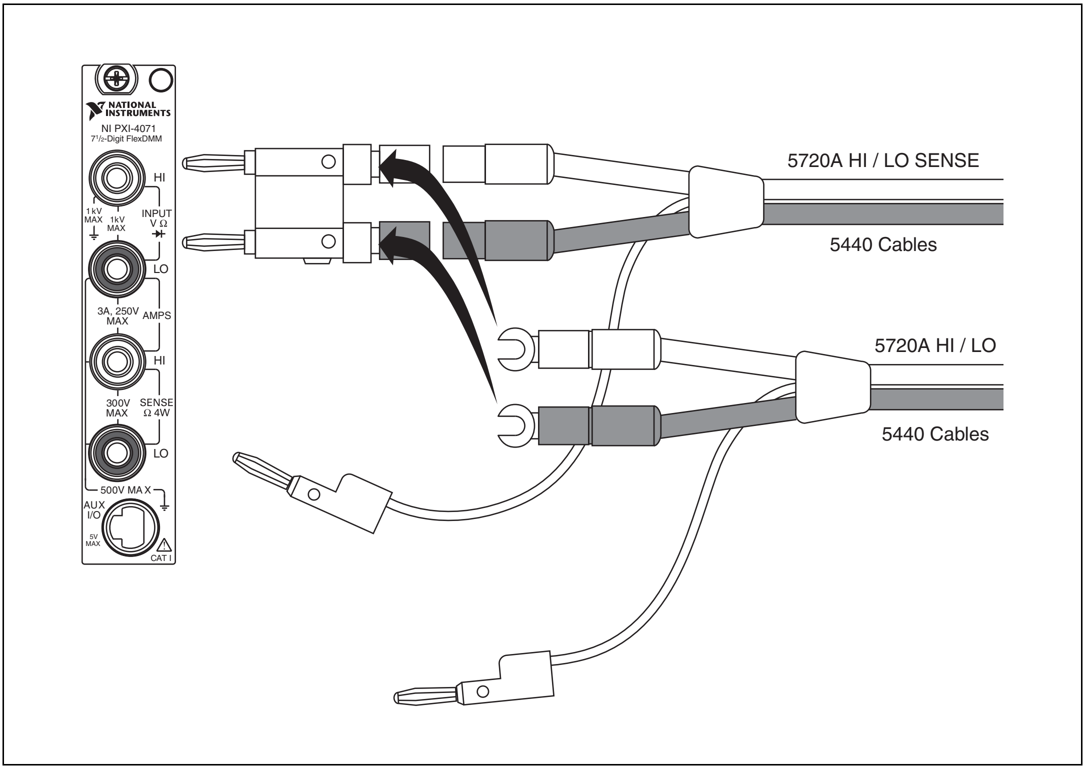


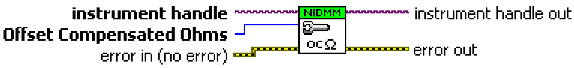
Figure 3. Cable Connections for 2-Wire Resistance

Table 11. Fluke 5440 Cable Connections

Banana Plug Connector (NI 4071)	Banana Plug Color (Fluke 5440 Cable)	Binding Post Label (Fluke 5720A Calibrator)
HI	Red	OUTPUT HI
LO	Black	OUTPUT LO
(No connection)	Blue	V GUARD
HI	Red	OUTPUT HI SENSE
LO	Black	OUTPUT LO SENSE
(No connection)	Blue	V GUARD

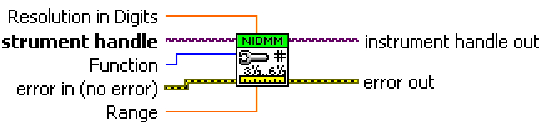
38. Wait two minutes for the thermal EMF to stabilize.

39. Call the niDMM Configure Offset Comp Ohms VI with **Offset Compensated Ohms** set to `NIDMM_VAL_OFFSET_COMP_OHMS_OFF`.

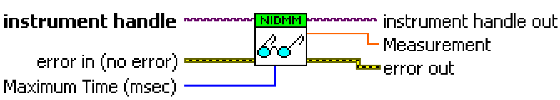
LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_ConfigureOffsetCompOhms` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **OffsetCompOhms**: `NIDMM_VAL_OFFSET_COMP_OHMS_OFF`

40. Generate 10 kΩ on the calibrator with 2-wire compensation and external sense turned on.

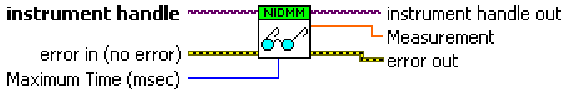
41. Call the niDMM Config Measurement VI and select the **Resolution in Digits** instance.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_ConfigureMeasurementDigits` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Resolution_Digits**: `7.5` **Measurement_Function**: `NIDMM_VAL_2_WIRE_RES` **Range**: `10e3`

42. Call the niDMM Read VI. This allows the calibrator to properly settle for a 10 kΩ compensated measurement at the NI 4071.

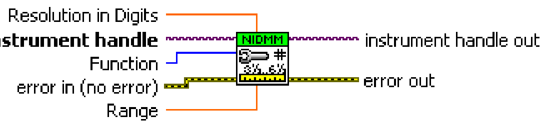
LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_read` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Reading**: The measurement returned by the function. This allows the calibrator to properly settle for a 10 kΩ compensated measurement at the NI 4071. **Maximum_Time**: `-1`

43. Call the niDMM Read VI.

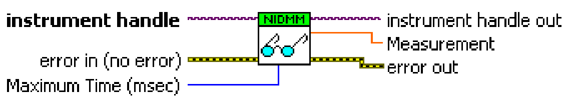
LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_read` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Reading**: The measurement returned by the function. Verify that the reading or measurement value falls between the tolerances listed in Table 24. **Maximum_Time**: `-1`

44. Generate 1 kΩ on the calibrator with 2-wire compensation and external sense turned on.

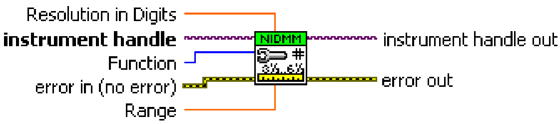
45. Call the niDMM Config Measurement VI and select the **Resolution in Digits** instance.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_ConfigureMeasurementDigits` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Resolution_Digits**: `7.5` **Measurement_Function**: `NIDMM_VAL_2_WIRE_RES` **Range**: `1e3`

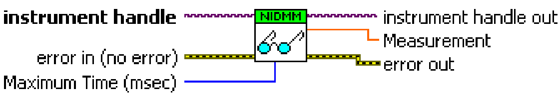
46. Call the niDMM Read VI.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_read` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Reading**: The measurement returned by the function. Verify that the reading or measurement value falls between the tolerances listed in Table 24. **Maximum_Time**: `-1`

47. Generate 100 Ω on the calibrator with 2-wire compensation and external sense turned on.
48. Call the niDMM Config Measurement VI and select the **Resolution in Digits** instance.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_ConfigureMeasurementDigits` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Resolution_Digits**: `7.5` **Measurement_Function**: `NIDMM_VAL_2_WIRE_RES` **Range**: `100`

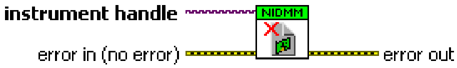
49. Call the niDMM Read VI.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_read` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Reading**: The measurement returned by the function. Verify that the reading or measurement value falls between the tolerances listed in Table 24. **Maximum_Time**: `-1`

You have completed verifying the 2-wire resistance of the NI 4071. Select one of the following options:

- If you want to perform the optional 5 GΩ range verification, go to the *5 GΩ Range Verification* section.
- If you want to continue verifying other modes, go to the *Verifying DC Current* section on page.
- If you do *not* want to verify other modes *and* you are performing a *post-adjustment* verification, go to the *Completing the Adjustment Procedures* section.

- If you do *not* want to verify any additional modes *and* you are performing a *pre-adjustment* verification, call the niDMM Close VI to close the session.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_close` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init`

5 GΩ Range Verification

This verification procedure is optional and requires additional test equipment. If you do *not* want to verify the 5 GΩ range, you have completed verifying the 2-wire resistance of the NI 4071.

1. Connect a 1 GΩ standard resistor to the NI 4071 using the Fluke 5440 cable or applicable low thermal cabling, as shown in Figure 4. Table 12 lists the cable connections.

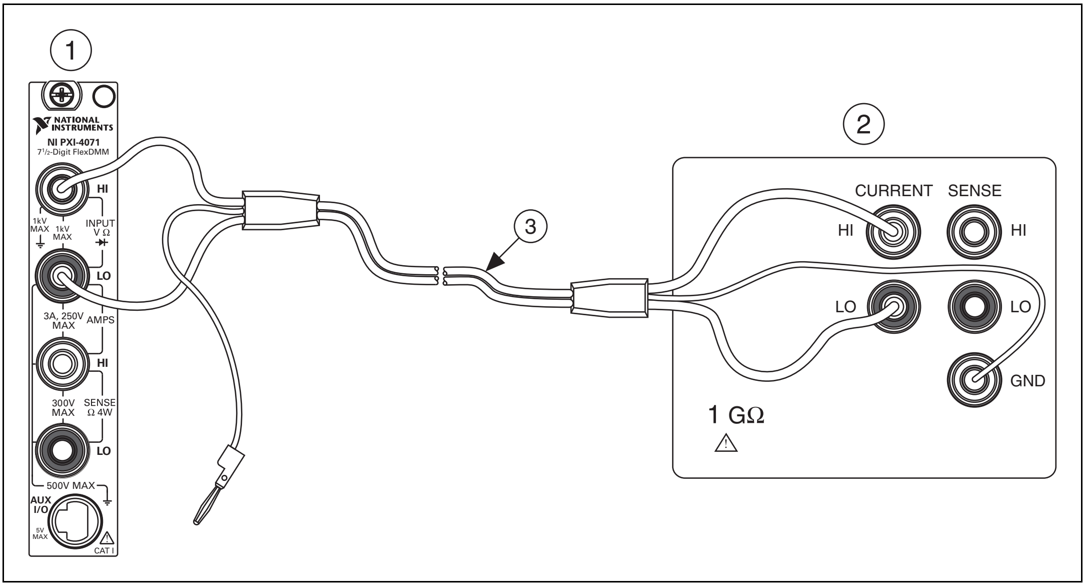
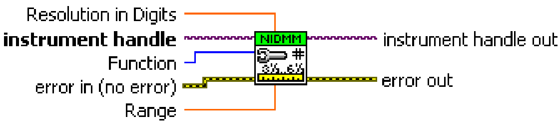


Figure 4. Cable Connections for 1 GΩ Standard Resistor

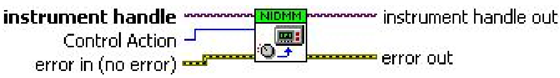
Table 12. Fluke 5440 Cable Connections

Banana Plug Connector (NI 4071)	Banana Plug Color (Fluke 5440 Cable)	Binding Post Label (1 GΩ Standard Resistor)
HI	Red	HI
LO	Black	LO
(No connection)	Blue	GND

2. Call the niDMM Config Measurement VI and select the **Resolution in Digits** instance.

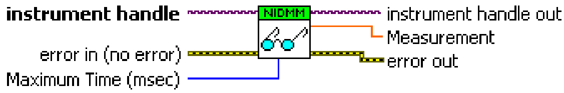
LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_ConfigureMeasurementDigits` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Resolution_Digits**: `7.5` **Measurement_Function**: `NIDMM_VAL_2_WIRE_RES` **Range**: `5e9`

3. Call the niDMM Control VI.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_Control` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Control Action**: `Commit`

4. Wait two minutes for the thermal EMF to stabilize.

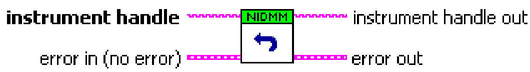
5. Call the niDMM Read VI.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_read` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Reading**: The measurement returned by the function. Verify that this measurement falls between the tolerances listed in Table 24. **Maximum_Time**: `-1`

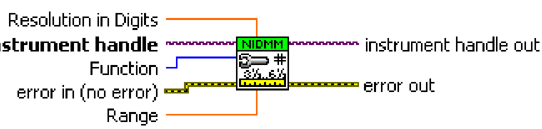
Verifying DC Current

To verify the DC current of the NI 4071, complete the following steps:

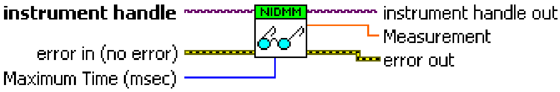
1. Reset the calibrator.
2. Remove all connections from the four input banana plug connectors on the NI 4071.
3. Call the niDMM Reset VI to reset the NI 4071 to a known state.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_reset` with the following parameter: **Instrument_Handle**: The instrument handle from `niDMM_init`

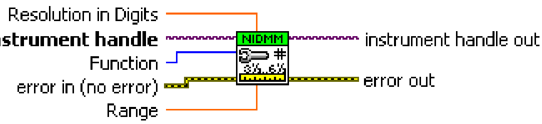
4. Call the niDMM Config Measurement VI and select the **Resolution in Digits** instance.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_ConfigureMeasurementDigits` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Resolution_Digits**: `6.5` **Measurement_Function**: `NIDMM_VAL_DC_CURRENT` **Range**: `100e-6`

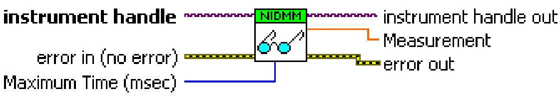
5. Call the niDMM Read VI.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_read` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Reading**: The measurement returned by the function. Verify that this measurement falls between the limits listed in Table 25. **Maximum_Time**: `-1`

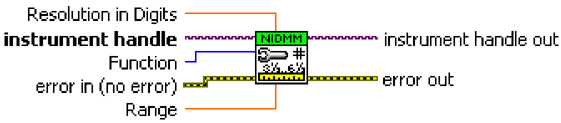
6. Call the niDMM Config Measurement VI and select the **Resolution in Digits** instance.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_ConfigureMeasurementDigits` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Resolution_Digits**: `6.5` **Measurement_Function**: `NIDMM_VAL_DC_CURRENT` **Range**: `1e-3`

7. Call the niDMM Read VI.

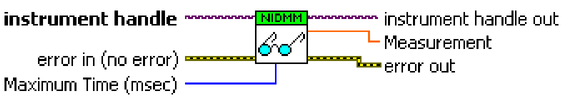
LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_read` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Reading**: The measurement returned by the function. Verify that this measurement falls between the limits listed in Table 25. **Maximum_Time**: `-1`

8. Call the niDMM Config Measurement VI and select the **Resolution in Digits** instance.

<table>
<tr><th>LabVIEW Block Diagram</th><th>C/C++ Function Call</th></tr>
<tr><td>
</td><td>Call <code>niDMM_ConfigureMeasurementDigits</code> with the following parameters:

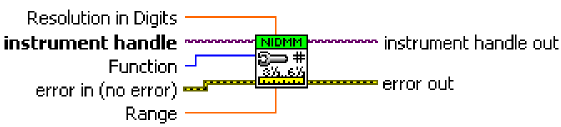
Instrument_Handle: The instrument handle from <code>niDMM_init</code>
Resolution_Digits: <code>6.5</code>
Measurement_Function: <code>NIDMM_VAL_DC_CURRENT</code>
Range: <code>10e-3</code></td></tr>
</table>

9. Call the niDMM Read VI.

<table>
<tr><th>LabVIEW Block Diagram</th><th>C/C++ Function Call</th></tr>
<tr><td>
</td><td>Call <code>niDMM_read</code> with the following parameters:

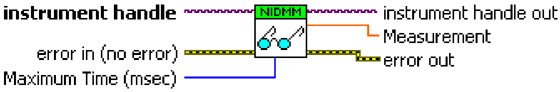
Instrument_Handle: The instrument handle from <code>niDMM_init</code>
Reading: The measurement returned by the function. Verify that this measurement falls between the limits listed in Table 25.
Maximum_Time: <code>-1</code></td></tr>
</table>

10. Call the niDMM Config Measurement VI and select the **Resolution in Digits** instance.

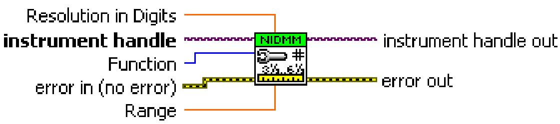
<table>
<tr><th>LabVIEW Block Diagram</th><th>C/C++ Function Call</th></tr>
<tr><td>
</td><td>Call <code>niDMM_ConfigureMeasurementDigits</code> with the following parameters:

Instrument_Handle: The instrument handle from <code>niDMM_init</code>
Resolution_Digits: <code>6.5</code>
Measurement_Function: <code>NIDMM_VAL_DC_CURRENT</code>
Range: <code>100e-3</code></td></tr>
</table>

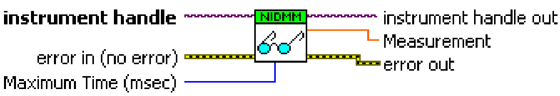
11. Call the niDMM Read VI.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_read` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Reading**: The measurement returned by the function. Verify that this measurement falls between the limits listed in Table 25. **Maximum_Time**: `-1`

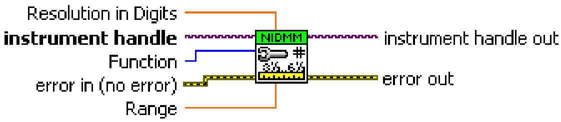
12. Call the niDMM Config Measurement VI and select the **Resolution in Digits** instance.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_ConfigureMeasurementDigits` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Resolution_Digits**: `6.5` **Measurement_Function**: `NIDMM_VAL_DC_CURRENT` **Range**: `1`

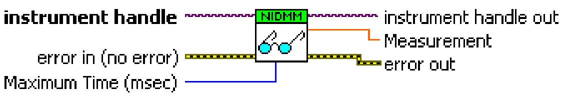
13. Call the niDMM Read VI.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_read` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Reading**: The measurement returned by the function. Verify that this measurement falls between the limits listed in Table 25. **Maximum_Time**: `-1`

14. Call the niDMM Config Measurement VI and select the **Resolution in Digits** instance.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_ConfigureMeasurementDigits` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Resolution_Digits**: `6.5` **Measurement_Function**: `NIDMM_VAL_DC_CURRENT` **Range**: `3`

15. Call the niDMM Read VI.

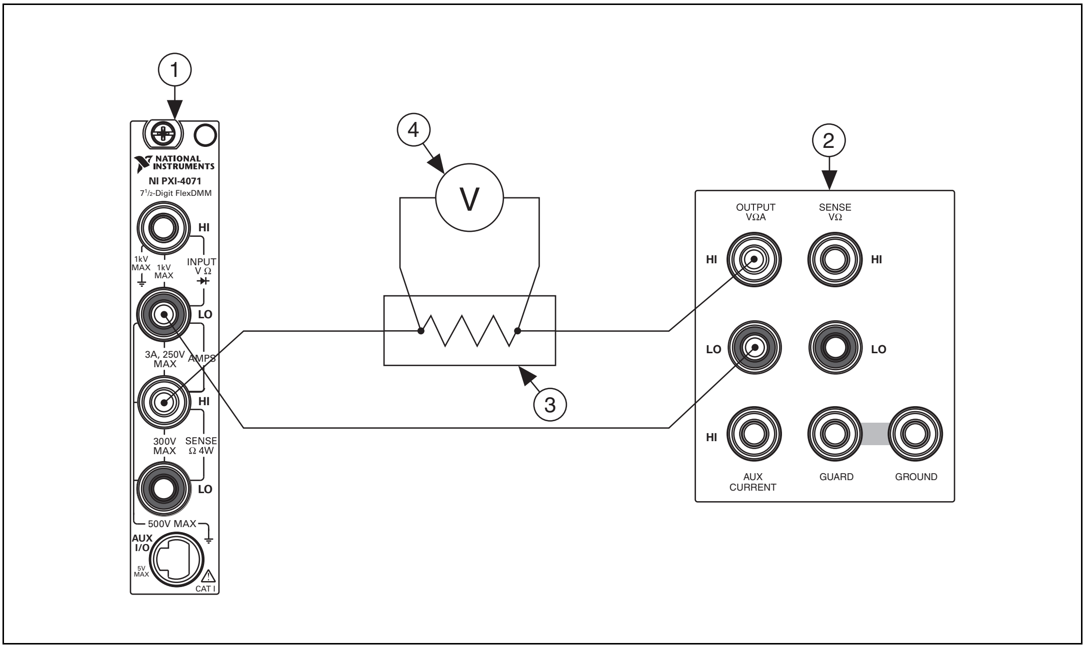
LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_read` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Reading**: The measurement returned by the function. Verify that this measurement falls between the limits listed in Table 25. **Maximum_Time**: `-1`

Note The 90-day DC current uncertainty of the Fluke 5720A is not adequate to calibrate the 100 µA and 1 mA DC current ranges listed in Table 25. You must characterize the calibrator output to better than published specifications. Use metrology laboratory methods to obtain the required uncertainty. Monitor the current applied using a DC current shunt/voltmeter system in series with the calibrator output. Refer to Table 13 for the recommended DC current uncertainty requirements for the DC current shunt/voltmeter system.

Table 13. Recommended DC current uncertainty requirements for Fluke 5720A

DC Current Calibration Point	Recommended Precision Shunt Resistance	Uncertainty (k=2)
±100 µA	10 kΩ	40 µA/A
±1 mA	1 kΩ	40 µA/A

16. Connect the Fluke 5720A calibrator to the NI 4071 with a DC current shunt/voltmeter system in series, as shown in Figure 5. Table 14 lists the cable connections. Use the shunt recommended for ±100 µA as indicated in Table 13.
17. Wait two minutes for the thermal EMF to stabilize.



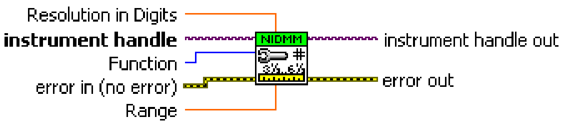
1	NI 4071	2	Fluke 5720A Calibrator	3	Precision Shunt Resistor	4	Voltmeter

Figure 5. Shunt/Voltmeter Connections for Current

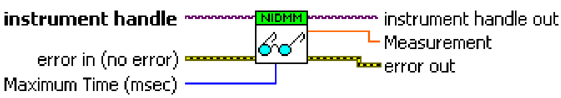
Table 14. Shunt/Voltmeter Connections

Banana Plug Connector (NI 4071)	Precision Shunt Resistor	Voltmeter	Binding Post (Fluke 5720A Calibrator)
—	HI	HI	OUTPUT HI
HI SENSE	LO	LO	—
LO	—	—	OUTPUT LO
—	GND	V GUARD	V GUARD

18. Call the niDMM Config Measurement VI and select the **Resolution in Digits** instance.

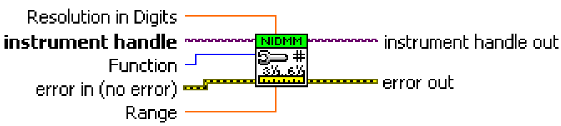
LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_ConfigureMeasurementDigits` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Resolution_Digits**: `6.5` **Measurement_Function**: `NIDMM_VAL_DC_CURRENT` **Range**: `100e-6`

19. Call the niDMM Read VI. Configure the NI 4071 for a current mode before applying current.

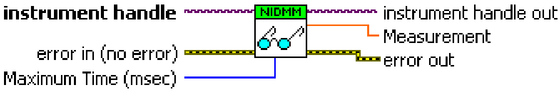
LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_read` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Reading**: The measurement returned by the function. Configure the NI 4071 for a current mode before applying current. **Maximum_Time**: `-1`

20. Generate 100 µA on the calibrator with the current output set to `NORM`.

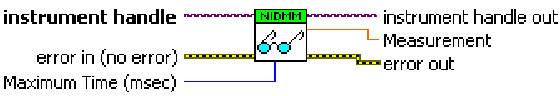
21. Call the niDMM Config Measurement VI and select the **Resolution in Digits** instance.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_ConfigureMeasurementDigits` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Resolution_Digits**: `6.5` **Measurement_Function**: `NIDMM_VAL_DC_CURRENT` **Range**: `100e-6`

22. Call the niDMM Read VI.

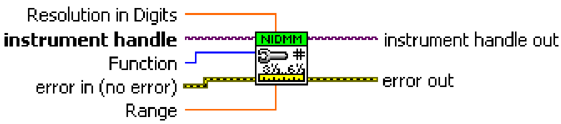
LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_read` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Reading**: The measurement returned by the function. Verify that this measurement falls between the limits listed in Table 25. **Maximum_Time**: `-1`

23. Generate –100 µA on the calibrator with the current output set to `NORM`.
24. Call the niDMM Read VI.

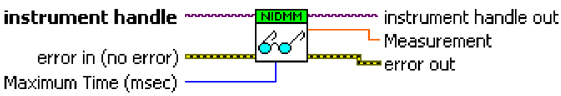
LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_read` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Reading**: The measurement returned by the function. Verify that the reading or measurement value falls between the limits listed in Table 25. **Maximum_Time**: `-1`

25. Replace the shunt recommended for 100 µA currents with the shunt recommended for 1 mA as indicated in Table 13.
26. Generate 1 mA on the calibrator with the current output set to `NORM`.

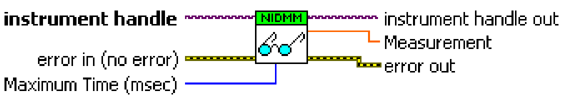
27. Call the niDMM Config Measurement VI and select the **Resolution in Digits** instance.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_ConfigureMeasurementDigits` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Resolution_Digits**: `6.5` **Measurement_Function**: `NIDMM_VAL_DC_CURRENT` **Range**: `1e-3`

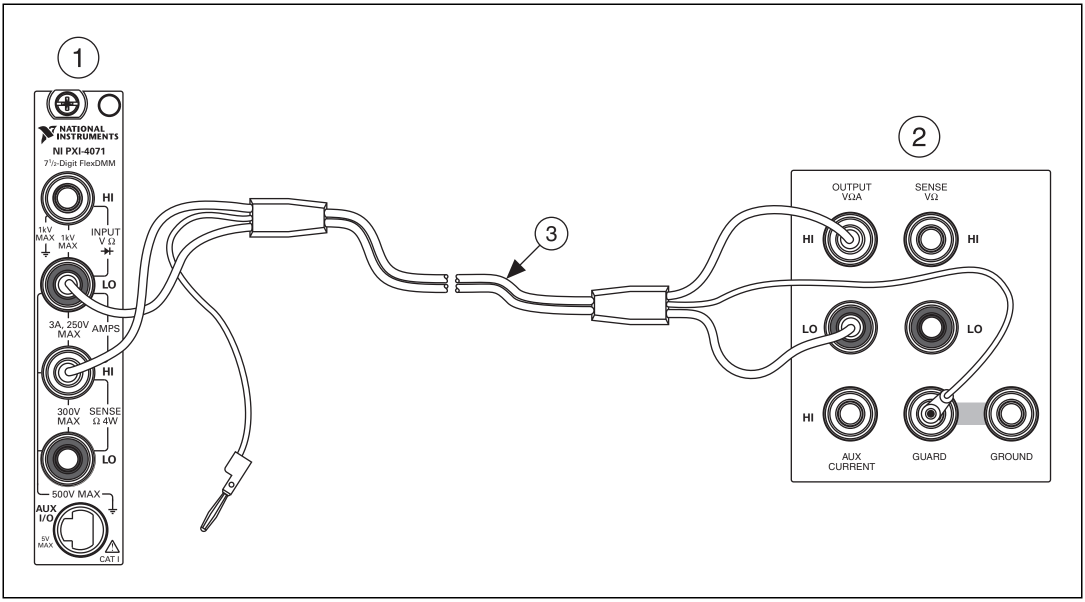
28. Call the niDMM Read VI.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_read` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Reading**: The measurement returned by the function. Verify that the reading or measurement value falls between the limits listed in Table 25. **Maximum_Time**: `-1`

29. Generate –1 mA on the calibrator with the current output set to `NORM`.
30. Call the niDMM Read VI.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_read` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Reading**: The measurement returned by the function. Verify that the reading or measurement value falls between the limits listed in Table 25. **Maximum_Time**: `-1`

31. Connect the NI 4071 and the Fluke 5720A calibrator using the Fluke 5440 cable, as shown in Figure 6. Table 15 lists the cable connections.



1 NI 4071	2 Fluke 5720A Calibrator	3 Fluke 5440 Cable

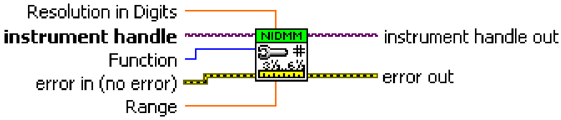
Figure 6. Cable Connections for Current

Table 15. Fluke 5440 Cable Connections

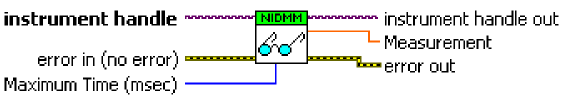
Banana Plug Connector (NI 4071)	Banana Plug Color (Fluke 5440 Cable)	Binding Post (Fluke 5720A Calibrator)
HI SENSE	Red	OUTPUT HI
LO	Black	OUTPUT LO
(No connection)	Blue	V GUARD

32. Wait two minutes for the thermal EMF to stabilize.

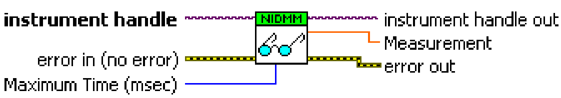
33. Call the niDMM Config Measurement VI and select the **Resolution in Digits** instance.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_ConfigureMeasurementDigits` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Resolution_Digits**: `6.5` **Measurement_Function**: `NIDMM_VAL_DC_CURRENT` **Range**: `0.01`

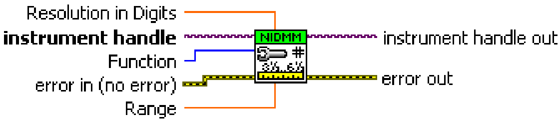
34. Generate 9 mA on the calibrator with the current output set to `NORM`.
35. Call the niDMM Read VI.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_read` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Reading**: The measurement returned by the function. Verify that this measurement falls between the limits listed in Table 25. **Maximum_Time**: `-1`

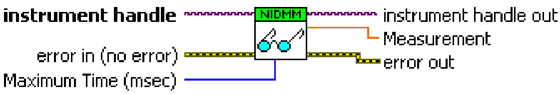
36. Generate –9 mA on the calibrator with the current output set to `NORM`.
37. Call the niDMM Read VI.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_read` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Reading**: The measurement returned by the function. Verify that this measurement falls between the limits listed in Table 25. **Maximum_Time**: `-1`

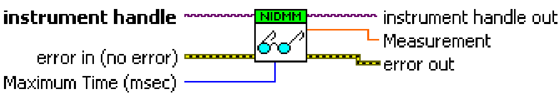
38. Generate 90 mA on the calibrator with the current output set to `NORM`.
39. Call the niDMM Config Measurement VI and select the **Resolution in Digits** instance.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_ConfigureMeasurementDigits` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Resolution_Digits**: `6.5` **Measurement_Function**: `NIDMM_VAL_DC_CURRENT` **Range**: `0.1`

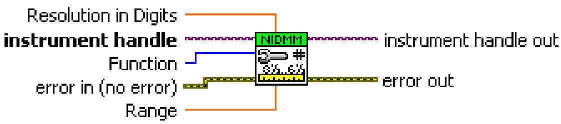
40. Call the niDMM Read VI.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_read` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Reading**: The measurement returned by the function. Verify that this measurement falls between the limits listed in Table 25. **Maximum_Time**: `-1`

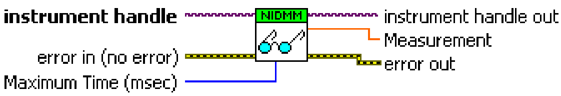
41. Generate –90 mA on the calibrator with the current output set to `NORM`.
42. Call the niDMM Read VI.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_read` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Reading**: The measurement returned by the function. Verify that this measurement falls between the limits listed in Table 25. **Maximum_Time**: `-1`

43. Call the niDMM Config Measurement VI and select the **Resolution in Digits** instance.

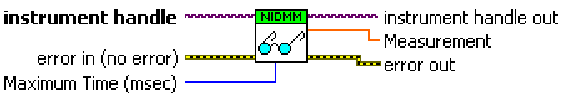
LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_ConfigureMeasurementDigits` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Resolution_Digits**: `6.5` **Measurement_Function**: `NIDMM_VAL_DC_CURRENT` **Range**: `1`

44. Call the niDMM Read VI. Configure the NI 4071 for a current mode before applying current.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_read` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Reading**: The measurement returned by the function. Configure the NI 4071 for a current mode before applying current. **Maximum_Time**: `-1`

45. Generate 0.9 A on the calibrator with the current output set to `NORM`.

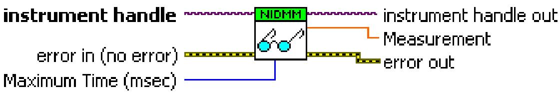
46. Call the niDMM Read VI.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_read` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Reading**: The measurement returned by the function. Verify that this measurement falls between the limits listed in Table 25. **Maximum_Time**: `-1`

47. Refer to Table 16 for the appropriate calibrator output signal and parameter measurement values as you complete the following steps:

    a. On the calibrator, generate the value listed under Calibrator Output in Table 16 for the current iteration.

    b. Call the niDMM Read VI.

<table>
<tr><th>LabVIEW Block Diagram</th><th>C/C++ Function Call</th></tr>
<tr><td>
</td><td>Call <code>niDMM_read</code> with the following parameters:

Instrument_Handle: The instrument handle from <code>niDMM_init</code>
Reading: The measurement returned by the function. Verify that this measurement falls between the limits listed in Table 25.
Maximum_Time: <code>-1</code></td></tr>
</table>

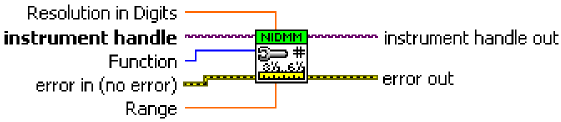
Table 16. niDMM 1 A Linearity Settings

Iteration	Calibrator Output
1	0.5 A
2	0.25 A
3	–0.25 A
4	–0.5 A
5	–0.9 A

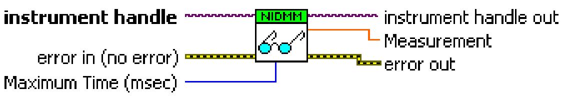
48. Repeat step 47 for each of the remaining iterations shown in Table 16.

49. Generate 2.2 A on the calibrator with the current output set to `NORM`.

50. Call the niDMM Config Measurement VI and select the **Resolution in Digits** instance.

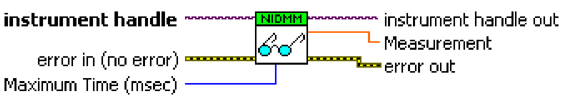
LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_ConfigureMeasurementDigits` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Resolution_Digits**: `6.5` **Measurement_Function**: `NIDMM_VAL_DC_CURRENT` **Range**: `3`

51. Call the niDMM Read VI.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_read` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Reading**: The measurement returned by the function. Verify that this measurement falls between the limits listed in Table 25. **Maximum_Time**: `-1`

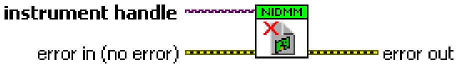
52. Generate –2.2 A on the calibrator with the current output set to `NORM`.

53. Call the niDMM Read VI.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_read` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Reading**: The measurement returned by the function. Verify that this measurement falls between the limits listed in Table 25. **Maximum_Time**: `-1`

You have completed verifying the DC current of the NI 4071. Select one of the following options:

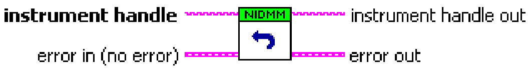
- If you want to continue verifying other modes, go to the *Verifying AC Current* section.
- If you do *not* want to verify other modes *and* you are performing a *post-adjustment* verification, go to the *Completing the Adjustment Procedures* section.
- If you do *not* want to verify any additional modes *and* you are performing a *pre-adjustment* verification, call the niDMM Close VI to close the session.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_close` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init`

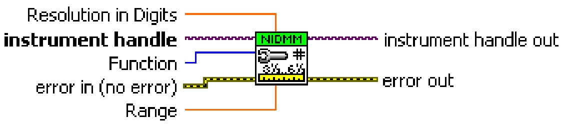
Verifying AC Current

To verify the AC current of the NI 4071, complete the following steps:

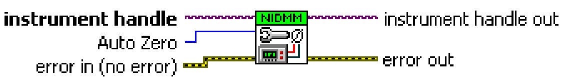
1. Reset the calibrator.
2. Connect the NI 4071 and the Fluke 5720A calibrator using the Fluke 5440 cable, as shown in Figure 6. Table 15 lists the cable connections.
3. Call the niDMM Reset VI to reset the NI 4071 to a known state.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_reset` with the following parameter: **Instrument_Handle**: The instrument handle from `niDMM_init`

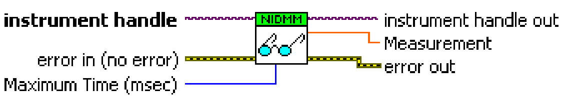
4. Call the niDMM Config Measurement VI and select the **Resolution in Digits** instance.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_ConfigureMeasurementDigits` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Resolution_Digits**: `6.5` **Measurement_Function**: `NIDMM_VAL_AC_CURRENT` **Range**: `100e-6`

5. Call the niDMM Configure Auto Zero VI.

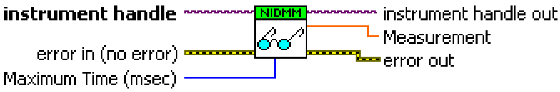
LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_ConfigureAutoZeroMode` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Auto_Zero_Mode**: `NIDMM_VAL_AUTO_ZERO_ON`

6. Call the niDMM Read VI. Configure the NI 4071 for a current mode before applying current.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_read` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Reading**: The measurement returned by the function. Configure the NI 4071 for a current mode before applying current. **Maximum_Time**: `-1`

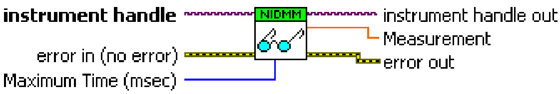
7. Generate 9 µA at 1 kHz on the calibrator with the current output set to `NORM`.

8. Call the niDMM Read VI.

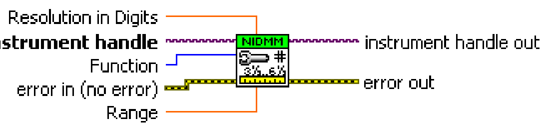
LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_read` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Reading**: The measurement returned by the function. Verify that this measurement falls between the limits listed in Table 26. **Maximum_Time**: `-1`

9. Generate 90 µA at 1 kHz on the calibrator with the current output set to `NORM`.

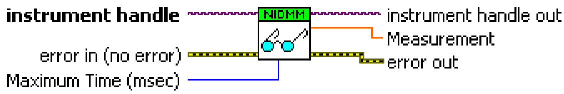
10. Call the niDMM Read VI.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_read` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Reading**: The measurement returned by the function. Verify that this measurement falls between the limits listed in Table 26. **Maximum_Time**: `-1`

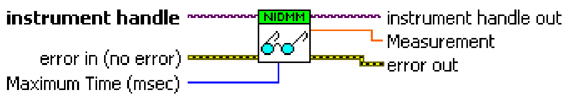
11. Call the niDMM Config Measurement VI and select the **Resolution in Digits** instance.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_ConfigureMeasurementDigits` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Resolution_Digits**: `6.5` **Measurement_Function**: `NIDMM_VAL_AC_CURRENT` **Range**: `1e-3`

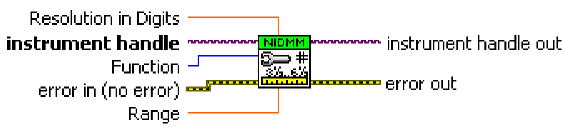
12. Call the niDMM Read VI.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_read` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Reading**: The measurement returned by the function. Verify that this measurement falls between the limits listed in Table 26. **Maximum_Time**: `-1`

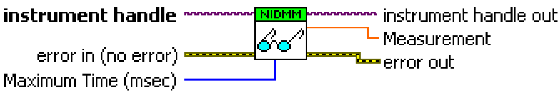
13. Generate 0.9 mA at 1 kHz on the calibrator with the current output set to `NORM`.
14. Call the niDMM Read VI.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_read` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Reading**: The measurement returned by the function. Verify that this measurement falls between the limits listed in Table 26. **Maximum_Time**: `-1`

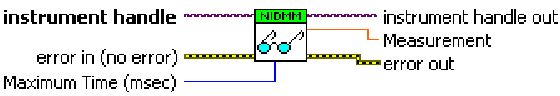
15. Call the niDMM Config Measurement VI and select the **Resolution in Digits** instance.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_ConfigureMeasurementDigits` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Resolution_Digits**: `6.5` **Measurement_Function**: `NIDMM_VAL_AC_CURRENT` **Range**: `0.01`

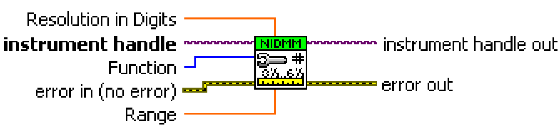
16. Call the niDMM Read VI.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_read` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Reading**: The measurement returned by the function. Verify that this measurement falls between the limits listed in Table 26. **Maximum_Time**: `-1`

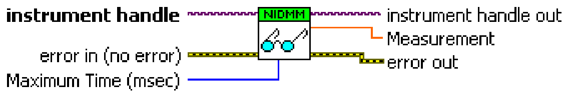
17. Generate 9 mA at 1 kHz on the calibrator with the current output set to `NORM`.
18. Call the niDMM Read VI.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_read` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Reading**: The measurement returned by the function. Verify that this measurement falls between the limits listed in Table 26. **Maximum_Time**: `-1`

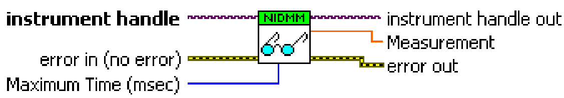
19. Call the niDMM Config Measurement VI and select the **Resolution in Digits** instance.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_ConfigureMeasurementDigits` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Resolution_Digits**: `6.5` **Measurement_Function**: `NIDMM_VAL_AC_CURRENT` **Range**: `0.1`

20. Call the niDMM Read VI.

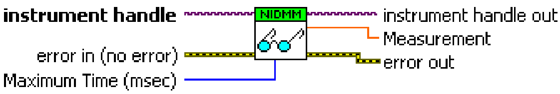
LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_read` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Reading**: The measurement returned by the function. Verify that this measurement falls between the limits listed in Table 26. **Maximum_Time**: `-1`

21. Generate 90 mA at 55 Hz on the calibrator with the current output set to `NORM`.
22. Call the niDMM Read VI.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_read` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Reading**: The measurement returned by the function. Verify that this measurement falls between the limits listed in Table 26. **Maximum_Time**: `-1`

23. Generate 90 mA at 5 kHz on the calibrator with the current output set to `NORM`.

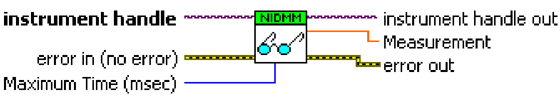
24. Call the niDMM Read VI.

<table>
<tr><th>LabVIEW Block Diagram</th><th>C/C++ Function Call</th></tr>
<tr><td>
</td><td>Call <code>niDMM_read</code> with the following parameters:

Instrument_Handle: The instrument handle from <code>niDMM_init</code>
Reading: The measurement returned by the function. Verify that this measurement falls between the limits listed in Table 26.
Maximum_Time: <code>-1</code></td></tr>
</table>

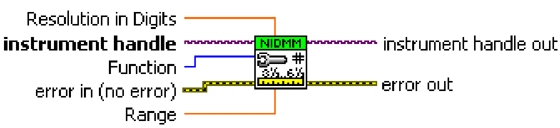
25. Generate 90 mA at 1 kHz on the calibrator with the current output set to `NORM`.

26. Call the niDMM Read VI.

<table>
<tr><th>LabVIEW Block Diagram</th><th>C/C++ Function Call</th></tr>
<tr><td>
</td><td>Call <code>niDMM_read</code> with the following parameters:

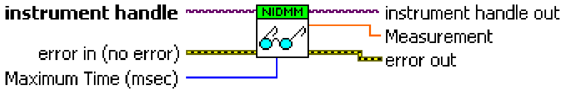
Instrument_Handle: The instrument handle from <code>niDMM_init</code>
Reading: The measurement returned by the function. Verify that this measurement falls between the limits listed in Table 26.
Maximum_Time: <code>-1</code></td></tr>
</table>

27. Call the niDMM Config Measurement VI and select the **Resolution in Digits** instance.

<table>
<tr><th>LabVIEW Block Diagram</th><th>C/C++ Function Call</th></tr>
<tr><td>
</td><td>Call <code>niDMM_ConfigureMeasurementDigits</code> with the following parameters:

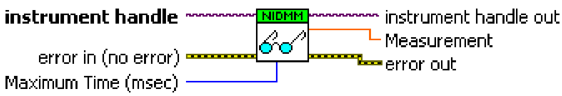
Instrument_Handle: The instrument handle from <code>niDMM_init</code>
Resolution_Digits: <code>6.5</code>
Measurement_Function: <code>NIDMM_VAL_AC_CURRENT</code>
Range: <code>1</code></td></tr>
</table>

28. Call the niDMM Read VI.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_read` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Reading**: The measurement returned by the function. Verify that this measurement falls between the limits listed in Table 26. **Maximum_Time**: `-1`

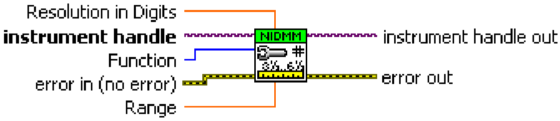
29. Generate 0.9 A at 1 kHz on the calibrator with the current output set to `NORM`.

30. Call the niDMM Read VI.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_read` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Reading**: The measurement returned by the function. Verify that this measurement falls between the limits listed in Table 26. **Maximum_Time**: `-1`

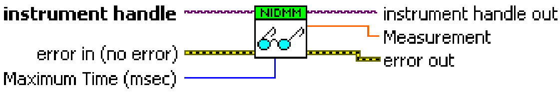
31. Generate 300 mA at 1 kHz on the calibrator with the current output set to `NORM`.

32. Call the niDMM Config Measurement VI and select the **Resolution in Digits** instance.

<table>
<tr><th>LabVIEW Block Diagram</th><th>C/C++ Function Call</th></tr>
<tr><td>
</td><td>Call <code>niDMM_ConfigureMeasurementDigits</code> with the following parameters:

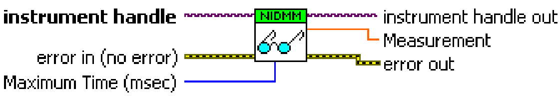
Instrument_Handle: The instrument handle from <code>niDMM_init</code>
Resolution_Digits: <code>6.5</code>
Measurement_Function: <code>NIDMM_VAL_AC_CURRENT</code>
Range: <code>3</code></td></tr>
</table>

33. Call the niDMM Read VI. Verify that this measurement falls between the limits listed in Table 26.

<table>
<tr><th>LabVIEW Block Diagram</th><th>C/C++ Function Call</th></tr>
<tr><td>
</td><td>Call <code>niDMM_read</code> with the following parameters:

Instrument_Handle: The instrument handle from <code>niDMM_init</code>
Reading: Verify that this measurement falls between the limits listed in Table 26.
Maximum_Time: <code>-1</code></td></tr>
</table>

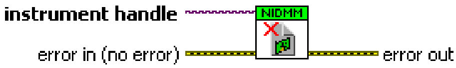
34. Generate 2.2 A at 1 kHz on the calibrator with the current output set to `NORM`.
35. Call the niDMM Read VI.

<table>
<tr><th>LabVIEW Block Diagram</th><th>C/C++ Function Call</th></tr>
<tr><td>
</td><td>Call <code>niDMM_read</code> with the following parameters:

Instrument_Handle: The instrument handle from <code>niDMM_init</code>
Reading: The measurement returned by the function. Verify that this measurement falls between the limits listed in Table 26.
Maximum_Time: <code>-1</code></td></tr>
</table>

You have completed verifying the AC current of the NI 4071. Select one of the following options:

- If you want to continue verifying other modes, go to the *Verifying Frequency* section.
- If you do *not* want to verify other modes *and* you are performing a *post-adjustment* verification, go to the *Completing the Adjustment Procedures* section.
- If you do *not* want to verify any additional modes *and* you are performing a *pre-adjustment* verification, call the niDMM Close VI to close the session.

<table>
<tr><th>LabVIEW Block Diagram</th><th>C/C++ Function Call</th></tr>
<tr><td>
</td><td>Call <code>niDMM_close</code> with the following parameters:

Instrument_Handle: The instrument handle from <code>niDMM_init</code></td></tr>
</table>

Verifying Frequency

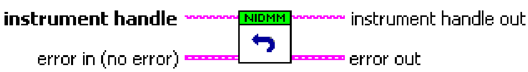
Note The frequency of the NI 4071 is *not* user adjustable. If this verification procedure indicates that the frequency is out of specification, return the NI 4071 to NI for repair.

To verify the frequency of the NI 4071, complete the following steps:

1. Remove all connections from the NI 4071.

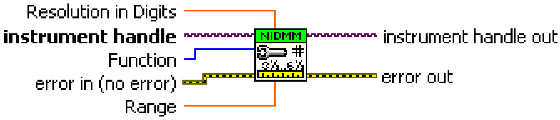
Note Polarity is *not* important in step 2.

2. Connect a device capable of generating a pulse train at the frequencies listed in Table 27 to the HI and LO terminals of the NI 4071. The pulse train should be 0 to 5 V and have a duty cycle of 50%. See Table 1 for possible devices and equipment to use in this step.
3. Call the niDMM Reset VI to reset the NI 4071 to a known state.

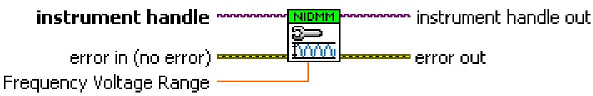
<table>
<tr><th>LabVIEW Block Diagram</th><th>C/C++ Function Call</th></tr>
<tr><td>
</td><td>Call <code>niDMM_reset</code> with the following parameter:

Instrument_Handle: The instrument handle from <code>niDMM_init</code></td></tr>
</table>

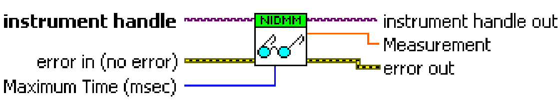
4. Call the niDMM Config Measurement VI and select the **Resolution in Digits** instance.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_ConfigureMeasurement Digits` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Resolution_Digits**: `6.5` **Measurement_Function**: `NIDMM_VAL_FREQ` **Range**: `1`

5. Call the niDMM Configure Frequency Voltage Range VI with **Voltage Range** set to `5`.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_ConfigureFrequency VoltageRange` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Frequency_Voltage_Range**: `5`

6. Configure the external source to generate a pulse train with a voltage of 0 to 5 V and a duty cycle of 50%. The frequency should be set to 1 Hz.
7. Call the niDMM Read VI.

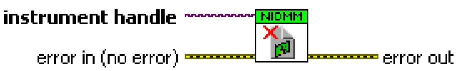
LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_read` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Reading**: The measurement returned by the function. Verify that this measurement falls between the limits listed in Table 27. **Maximum_Time**: `-1`

8. Repeat steps 6 and 7 with the following modification: Change the frequency to 20 kHz.

9. Repeat steps 6 and 7 with the following modification: Change the frequency to 500 kHz.

You have completed verifying the frequency of the NI 4071. Select one of the following options:

- If you do *not* want to verify other modes *and* you are performing a *post-adjustment* verification, go to the *Completing the Adjustment Procedures* section.
- If you do *not* want to verify any additional modes *and* you are performing a *pre-adjustment* verification, call the niDMM Close VI to close the session.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_close` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init`

Adjustment Procedures

This section explains how to adjust the NI 4071. You can choose to perform these adjustment procedures with or without performing the verification procedures first.

The parameters **Range**, **Resolution**, **Expected Measurement**, and **Frequency** used in function calls in this section have floating point values. For example, if **Range** = `1`, the floating point value is `1.0`. Refer to the *NI Digital Multimeters Help* for more information about parameter values.

Note NI recommends repeating the verification procedures after you perform these adjustment procedures. Reverification ensures that the device you have calibrated is operating within specifications after adjustments.

Caution If you skip any of the steps within a section of the adjustment procedures, NI-DMM does *not* allow you to store your new calibration coefficients. Instead, NI-DMM restores the original coefficients to the EEPROM.

Setting Up the Test Equipment

If you have not already set up the test equipment, complete the following steps:

1. Remove all connections from the four input banana plug connectors on the NI 4071.
2. Verify that the calibrator has been calibrated within the time limits specified in the *Test Equipment* section, and that DC zeros calibration has been performed within the last 30 days. Consult the Fluke 5720A user documentation for instructions on calibrating these devices.

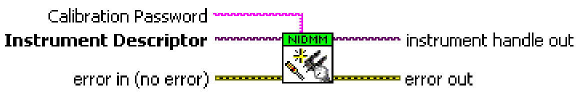
Note Ensure that the calibrator is warmed up for at least 60 minutes before you begin this procedure.

3. Reset the calibrator.
4. If you have not already done so, allow the NI 4071 to warm up for 60 minutes within a powered-on PXI chassis.

Adjusting DC Voltage and Resistance

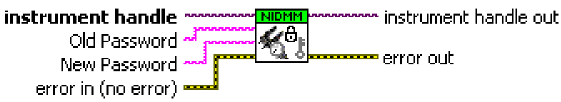
To adjust the DC voltage and resistance of the NI 4071, complete the following steps:

1. Connect the NI 4071 and the Fluke 5720A calibrator using the Fluke 5440 cable, as shown in Figure 1. Table 2 lists the cable connections.
2. Wait two minutes for the thermal EMF to stabilize if the cable was not previously connected in this configuration.
3. Call the niDMM Initialize External Cal VI with the resource descriptor of the NI 4071 and your valid user password to output a calibration session (**Cal Session**) that you can use to perform NI-DMM calibration or regular measurement functions.

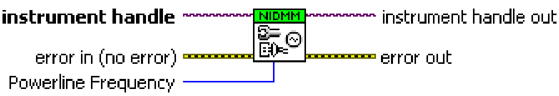
LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_InitExtCal` with the following parameters: **Instrument_Descriptor**: The name of the device to calibrate. You can find this name under Devices and Interfaces in Measurement & Automation Explorer (MAX). **Calibration_Password**: `NI`

Note You use **Cal Session** in all subsequent function calls.

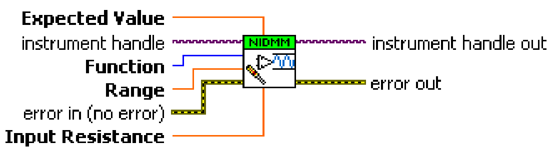
The default user password for adjusting the NI 4071 is `NI`. Call the niDMM Cal Set Password VI to change the password.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_SetCalPassword` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init`

4. Call the niDMM Configure Powerline Frequency VI with **Powerline Frequency** set to `50` or `60`, depending on the power line frequency (in hertz) that your instruments are powered from; select `50` for 400 Hz power line frequencies.

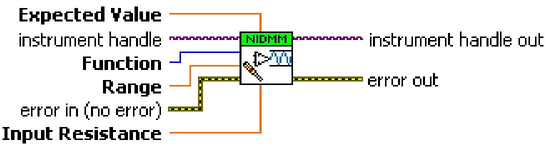
LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_ConfigurePowerLineFrequency` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Powerline_Frequency**: `50` or `60`

5. Generate 100 mV on the calibrator with the range locked to 2.2 V.
6. Call the niDMM Cal Adjust Gain VI with the following parameters:

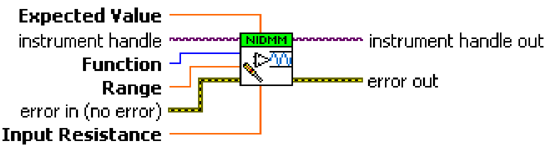
LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_CalAdjustGain` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Mode**: `NIDMM_VAL_DC_VOLTS` **Range**: `0.1` **Input_Resistance**: `NIDMM_VAL_10_MEGAOHM` **Expected_Value**: `0.1`

7. Generate –100 mV on the calibrator with the range locked to 2.2 V.

8. Call the niDMM Cal Adjust Gain VI with the following parameters:

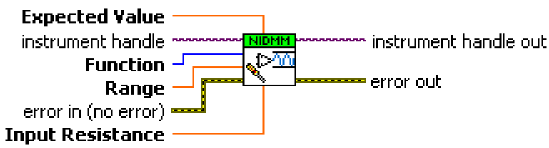
LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_CalAdjustGain` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Mode**: `NIDMM_VAL_DC_VOLTS` **Range**: `0.1` **Input_Resistance**: `NIDMM_VAL_10_MEGAOHM` **Expected_Value**: `-0.1`

9. Generate 10 V on the calibrator.
10. Call the niDMM Cal Adjust Gain VI with the following parameters:

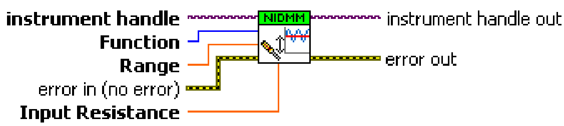
LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_CalAdjustGain` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Mode**: `NIDMM_VAL_DC_VOLTS` **Range**: `10` **Input_Resistance**: `NIDMM_VAL_GREATER_THAN_10_GIGAOHM` **Expected_Value**: `10`

11. Generate –10 V on the calibrator.

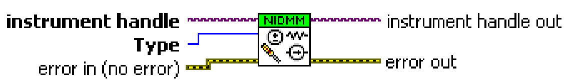
12. Call the niDMM Cal Adjust Gain VI with the following parameters:

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_CalAdjustGain` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Mode**: `NIDMM_VAL_DC_VOLTS` **Range**: `10` **Expected_Value**: `-10` **Input_Resistance**: `NIDMM_VAL_GREATER_THAN_10_ GIGAOHM`

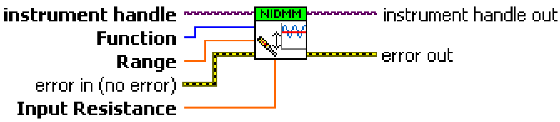
13. Disconnect the Fluke 5440 cable from the NI 4071 banana plug connectors, leaving the other end of the cable connected to the calibrator binding posts.
14. Plug in the insulated banana plug shorting bar across the *HI* and *LO* banana plug connectors of the NI 4071.
15. Wait two minutes for the thermal EMF to stabilize.
16. Call the niDMM Cal Adjust Offset VI with the following parameters:

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_CalAdjustOffset` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Mode**: `NIDMM_VAL_DC_VOLTS` **Range**: `10` **Input_Resistance**: `NIDMM_VAL_GREATER_THAN_10_ GIGAOHM`

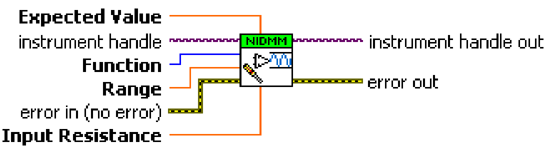
17. Call the niDMM Cal Adjust Miscellaneous VI.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_CalAdjustMisc` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Type**: `NIDMM_EXTCAL_MISCCAL_VREF`

18. Call the niDMM Cal Adjust Offset VI with the following parameters:

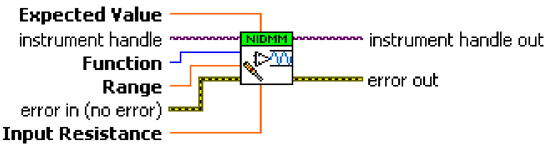
LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_CalAdjustOffset` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Mode**: `NIDMM_VAL_DC_VOLTS` **Range**: `0.1` **Input_Resistance**: `NIDMM_VAL_10_MEGAOHM`

19. Remove the shorting bar, and plug the Fluke 5440 cable back into the NI 4071 banana plug connectors, as shown in Figure 1.
20. Wait one minute for the thermal EMF to stabilize.
21. Generate 0 Ω from the calibrator without external sense or 2-wire compensation.
22. Call the niDMM Cal Adjust Gain VI with the following parameters:

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_CalAdjustGain` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Mode**: `NIDMM_VAL_2_WIRE_RES` **Range**: `10e6` **Input_Resistance**: `NIDMM_VAL_RESISTANCE_NA` **Expected_Value**: the display on the calibrator for 0 W

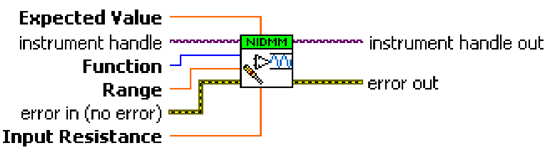
23. Generate 10 MΩ from the calibrator without external sense.

24. Call the niDMM Cal Adjust Gain VI with the following parameters:

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_CalAdjustGain` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Mode**: `NIDMM_VAL_2_WIRE_RES` **Range**: `10e6` **Input_Resistance**: `NIDMM_VAL_RESISTANCE_NA` **Expected_Value**: the display on the calibrator for 10 MW

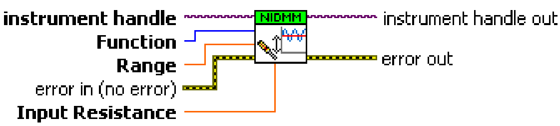
25. Generate 1.9 MΩ from the calibrator without external sense.

26. Call the niDMM Cal Adjust Gain VI with the following parameters:

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_CalAdjustGain` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Mode**: `NIDMM_VAL_2_WIRE_RES` **Range**: `10e6` **Input_Resistance**: `NIDMM_VAL_RESISTANCE_NA` **Expected_Value**: the display on the calibrator for 1.9 MW

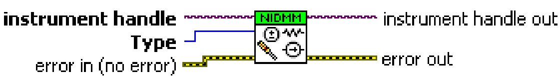
27. Generate 0 Ω from the calibrator without external sense or 2-wire compensation.

28. Call the niDMM Cal Adjust Offset VI with the following parameters:

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_CalAdjustOffset` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Mode**: `NIDMM_VAL_2_WIRE_RES` **Range**: `10e6` **Input_Resistance**: `NIDMM_VAL_RESISTANCE_NA`

29. Disconnect the Fluke 5440 cable from the NI 4071.

30. Call the niDMM Cal Adjust Miscellaneous VI with the following parameters:

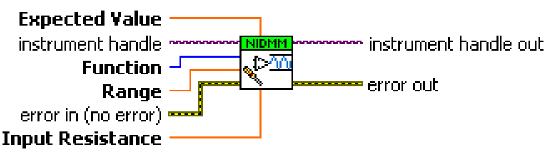
LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_CalAdjustMisc` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Type**: `NIDMM_EXTCAL_MISCCAL_ZINT`

31. Plug the two sets of Fluke 5440 cables into the appropriate banana plug connectors on the NI 4071, as shown in Figure 2 for 4-wire resistance.

32. Wait two minutes for the thermal EMF to stabilize.

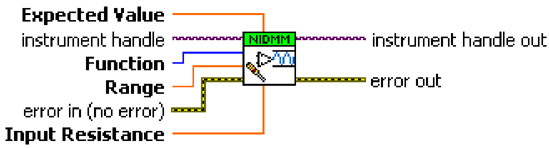
33. Generate 100 MΩ from the calibrator without external sense.

34. Call the niDMM Cal Adjust Gain VI with the following parameters:

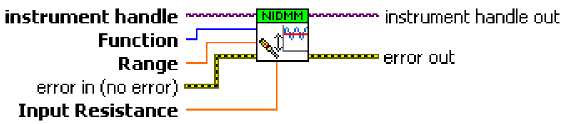
LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_CalAdjustGain` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Mode**: `NIDMM_VAL_2_WIRE_RES` **Range**: `100e6` **Input_Resistance**: `NIDMM_VAL_RESISTANCE_NA` **Expected_Value**: The value that is displayed on the calibrator for 100 MΩ

35. Generate 0 Ω from the calibrator without external sense or 2-wire compensation.

36. Call the niDMM Cal Adjust Gain VI with the following parameters:

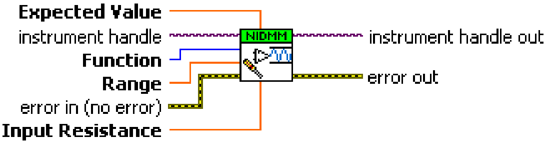
LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_CalAdjustGain` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Mode**: `NIDMM_VAL_2_WIRE_RES` **Range**: `100e6` **Input_Resistance**: `NIDMM_VAL_RESISTANCE_NA` **Expected_Value**: The value that is displayed on the calibrator for 0 Ω

37. Call the niDMM Cal Adjust Offset VI with the following parameters:

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_CalAdjustOffset` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Mode**: `NIDMM_VAL_2_WIRE_RES` **Range**: `100e6` **Input_Resistance**: `NIDMM_VAL_RESISTANCE_NA`

38. Generate 0 Ω on the calibrator with external sense turned on and 2-wire compensation turned off.

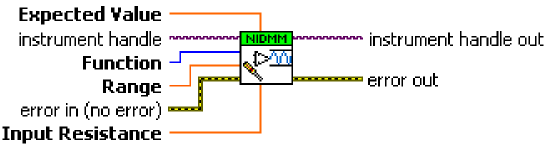
39. Call the niDMM Cal Adjust Gain VI with the following parameters:

<table>
<tr><th>LabVIEW Block Diagram</th><th>C/C++ Function Call</th></tr>
<tr><td>
</td><td>Call <code>niDMM_CalAdjustGain</code> with the following parameters:

Instrument_Handle: The instrument handle from <code>niDMM_init</code>
Mode: <code>NIDMM_VAL_4_WIRE_RES</code>
Range: <code>10e6</code>
Input_Resistance: <code>NIDMM_VAL_RESISTANCE_NA</code>
Expected_Value: The value that is displayed on the calibrator for 0 Ω</td></tr>
</table>

40. Generate 10 MΩ on the calibrator with external sense turned on and 2-wire compensation turned off.

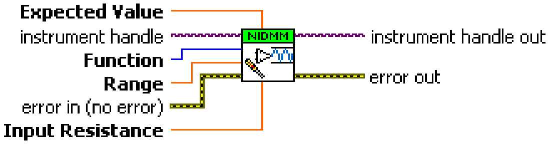
41. Call the niDMM Cal Adjust Gain VI with the following parameters:

<table>
<tr><th>LabVIEW Block Diagram</th><th>C/C++ Function Call</th></tr>
<tr><td>
</td><td>Call <code>niDMM_CalAdjustGain</code> with the following parameters:

Instrument_Handle: The instrument handle from <code>niDMM_init</code>
Mode: <code>NIDMM_VAL_4_WIRE_RES</code>
Range: <code>10e6</code>
Input_Resistance: <code>NIDMM_VAL_RESISTANCE_NA</code>
Expected_Value: The value that is displayed on the calibrator for 10 MΩ</td></tr>
</table>

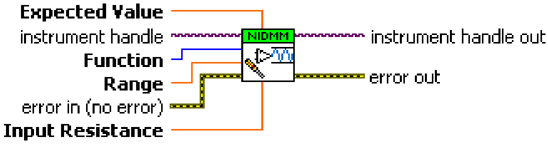
42. Generate 1.9 MΩ on the calibrator with external sense turned on and 2-wire compensation turned off.

43. Call the niDMM Cal Adjust Gain VI with the following parameters:

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_CalAdjustGain` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Mode**: `NIDMM_VAL_4_WIRE_RES` **Range**: `10e6` **Input_Resistance**: `NIDMM_VAL_RESISTANCE_NA` **Expected_Value**: The value that is displayed on the calibrator for 1.9 MΩ

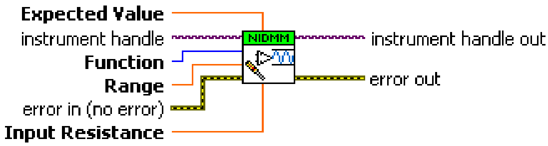
44. Generate 100 kΩ on the calibrator with external sense turned on and 2-wire compensation turned off.

45. Call the niDMM Cal Adjust Gain VI with the following parameters:

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_CalAdjustGain` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Mode**: `NIDMM_VAL_4_WIRE_RES` **Range**: `100e3` **Input_Resistance**: `NIDMM_VAL_RESISTANCE_NA` **Expected_Value**: The value that is displayed on the calibrator for 100 kΩ

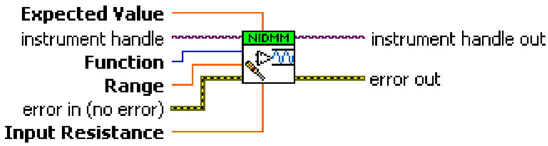
46. Generate 0 Ω on the calibrator with external sense turned on and 2-wire compensation turned off.

47. Call the niDMM Cal Adjust Gain VI with the following parameters:

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_CalAdjustGain` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Mode**: `NIDMM_VAL_4_WIRE_RES` **Range**: `100e3` **Input_Resistance**: `NIDMM_VAL_RESISTANCE_NA` **Expected_Value**: The value that is displayed on the calibrator for 0 Ω

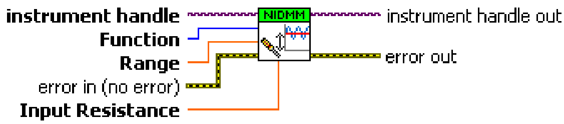
48. Generate 10 kΩ on the calibrator with external sense turned on and 2-wire compensation turned off.

49. Call the niDMM Cal Adjust Gain VI with the following parameters:

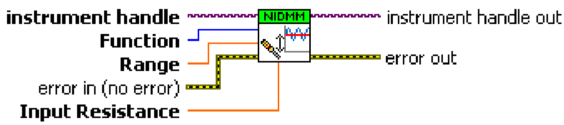
LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_CalAdjustGain` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Mode**: `NIDMM_VAL_4_WIRE_RES` **Range**: `10e3` **Input_Resistance**: `NIDMM_VAL_RESISTANCE_NA` **Expected_Value**: The value that is displayed on the calibrator for 10 kΩ

50. Generate 0 Ω on the calibrator with external sense turned on and 2-wire compensation turned off.

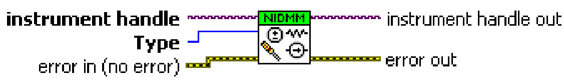
51. Call the niDMM Cal Adjust Offset VI with the following parameters:

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_CalAdjustOffset` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Mode**: `NIDMM_VAL_4_WIRE_RES` **Range**: `100e3` **Input_Resistance**: `NIDMM_VAL_RESISTANCE_NA`

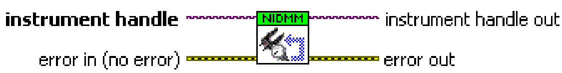
52. Call the niDMM Cal Adjust Offset VI with the following parameters:

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_CalAdjustOffset` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Mode**: `NIDMM_VAL_4_WIRE_RES` **Range**: `10e3` **Input_Resistance**: `NIDMM_VAL_RESISTANCE_NA`

53. Call the niDMM Cal Adjust Miscellaneous VI.

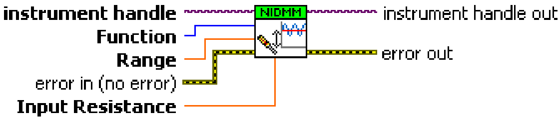
LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_CalAdjustMisc` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Type**: `NIDMM_EXTCAL_MISCAL_RREF`

54. Call the niDMM Self Cal VI to self-calibrate the NI 4071.

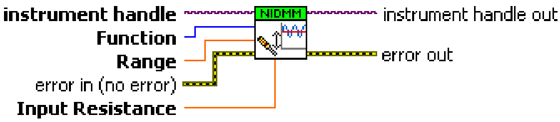
LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_SelfCal` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init`

55. Generate 0 Ω on the calibrator with external sense turned on, but with 2-wire compensation turned off.

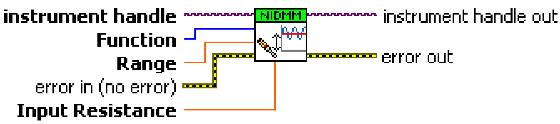
56. Call the niDMM Cal Adjust Offset VI with the following parameters:

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_CalAdjustOffset` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Mode**: `NIDMM_VAL_4_WIRE_RES` **Range**: `10e6` **Input_Resistance**: `NIDMM_VAL_RESISTANCE_NA`

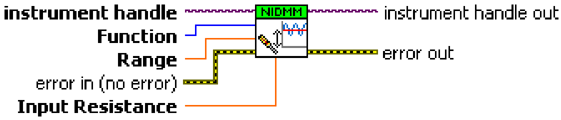
57. Call the niDMM Cal Adjust Offset VI with the following parameters:

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_CalAdjustOffset` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Mode**: `NIDMM_VAL_4_WIRE_RES` **Range**: `1e6` **Input_Resistance**: `NIDMM_VAL_RESISTANCE_NA`

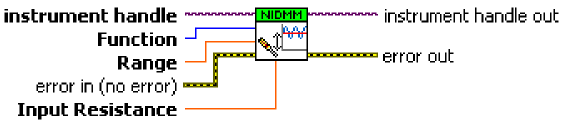
58. Call the niDMM Cal Adjust Offset VI with the following parameters:

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_CalAdjustOffset` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Mode**: `NIDMM_VAL_4_WIRE_RES` **Range**: `1e3` **Input_Resistance**: `NIDMM_VAL_RESISTANCE_NA`

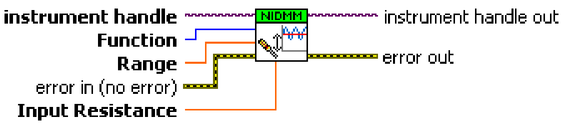
59. Call the niDMM Cal Adjust Offset VI with the following parameters:

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_CalAdjustOffset` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Mode**: `NIDMM_VAL_4_WIRE_RES` **Range**: `100` **Input_Resistance**: `NIDMM_VAL_RESISTANCE_NA`

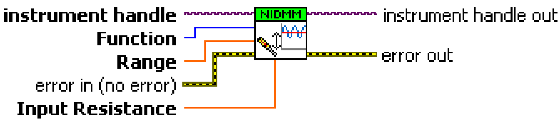
60. Remove the Fluke 5440 cables from the NI 4071.
61. Plug in the insulated shorting bar across the *HI* and *LO* banana plug connectors of the NI 4071.
62. Wait two minutes for the thermal EMF to stabilize.
63. Call the niDMM Cal Adjust Offset VI with the following parameters:

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_CalAdjustOffset` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Mode**: `NIDMM_VAL_2_WIRE_RES` **Range**: `10e6` **Input_Resistance**: `NIDMM_VAL_RESISTANCE_NA`

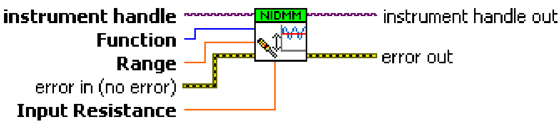
64. Call the niDMM Cal Adjust Offset VI with the following parameters:

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_CalAdjustOffset` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Mode**: `NIDMM_VAL_2_WIRE_RES` **Range**: `1e6` **Input_Resistance**: `NIDMM_VAL_RESISTANCE_NA`

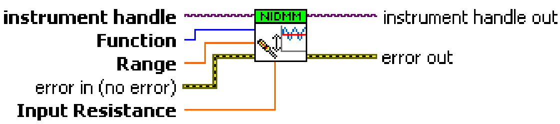
65. Call the niDMM Cal Adjust Offset VI with the following parameters:

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_CalAdjustOffset` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Mode**: `NIDMM_VAL_2_WIRE_RES` **Range**: `100e3` **Input_Resistance**: `NIDMM_VAL_RESISTANCE_NA`

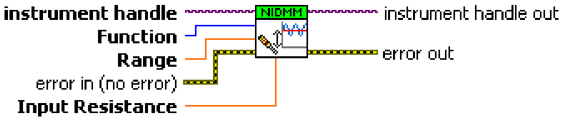
66. Call the niDMM Cal Adjust Offset VI with the following parameters:

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_CalAdjustOffset` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Mode**: `NIDMM_VAL_2_WIRE_RES` **Range**: `10e3` **Input_Resistance**: `NIDMM_VAL_RESISTANCE_NA`

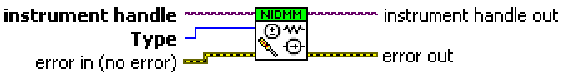
67. Call the niDMM Cal Adjust Offset VI with the following parameters:

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_CalAdjustOffset` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Mode**: `NIDMM_VAL_2_WIRE_RES` **Range**: `1e3` **Input_Resistance**: `NIDMM_VAL_RESISTANCE_NA`

68. Call the niDMM Cal Adjust Offset VI with the following parameters:

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_CalAdjustOffset` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Mode**: `NIDMM_VAL_2_WIRE_RES` **Range**: `100` **Input_Resistance**: `NIDMM_VAL_RESISTANCE_NA`

69. Call the niDMM Cal Adjust Miscellaneous VI with the following parameters:

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_CalAdjustMisc` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Type**: `NIDMM_EXTCAL_MISCCAL_SECTION`

You have completed adjusting the DC voltage and resistance modes of the NI 4071. Select one of the following options:

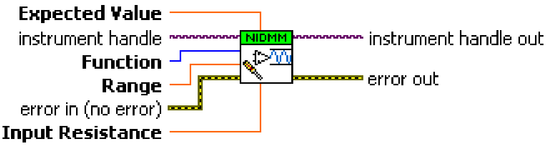
- If you are performing additional adjustments, refer to the following sections, as applicable:
  - *Adjusting AC Voltage (AC- and DC-Coupled) Modes*
  - *Adjusting Current Modes*
- If you are *not* performing additional adjustments, refer to one of the following sections:
  - *Verification Procedures*—to verify your new calibration coefficients before saving them to the EEPROM
  - *Completing the Adjustment Procedures*—if you do *not* want to verify the adjustments you have just made

Adjusting AC Voltage (AC- and DC-Coupled) Modes

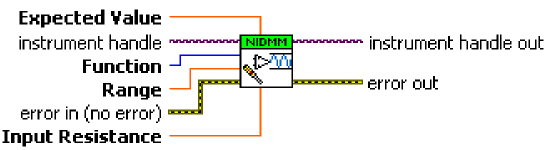
Note If you do not use the AC voltage modes for any measurements, or the accuracy of these modes is irrelevant, you can skip this section and go directly to the *Adjusting Current Modes* section.

To adjust the AC voltage of the NI 4071, complete the following steps:

1. Reset the calibrator.
2. Connect the NI 4071 and the Fluke 5720A calibrator using the Fluke 5440 cable, as shown in Figure 1. Table 2 lists the cable connections.
3. Refer to Table 17 for the appropriate calibrator output and parameter values as you complete the following steps:
   a. On the calibrator, generate the value listed under Calibrator Output in Table 17 for the current iteration.
   b. Call the niDMM Cal Adjust Gain VI with **Mode** set to `NIDMM_VAL_AC_VOLTS`. Set the remaining parameters as shown in Table 17 for the current iteration.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_CalAdjustGain` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Mode**: `NIDMM_VAL_AC_VOLTS`

   c. Call the niDMM Cal Adjust Gain VI again, changing **Mode** to `NIDMM_VAL_AC_VOLTS_DCCOUPLED`.

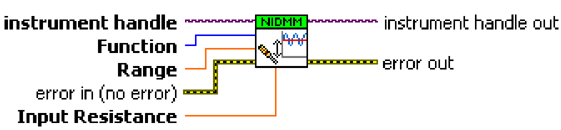
LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_CalAdjustGain` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Mode**: `NIDMM_VAL_AC_VOLTS_ DCCOUPLED`

4. Repeat step 3 for each of the remaining iterations listed in Table 17.

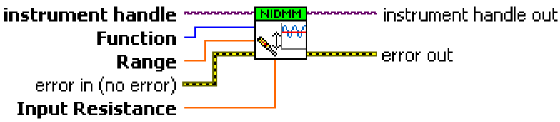
Table 17. niDMM_CalAdjustGain Parameters

Iteration	Calibrator Output		niDMM_CalAdjustGain Parameters			
	Amplitude	f (kHz)	Mode	Range (V)	Input Resistance	Expected Value
1	50 mV	1	`NIDMM_VAL_AC_VOLTS`	`0.05`	`NIDMM_VAL_10_MEGAOHM`	0.05
	50 mV	1	`NIDMM_VAL_AC_VOLTS_DCCOUPLED`	`0.05`	`NIDMM_VAL_10_MEGAOHM`	0.05
2	500 mV	1	`NIDMM_VAL_AC_VOLTS`	`0.5`	`NIDMM_VAL_10_MEGAOHM`	0.5
	500 mV	1	`NIDMM_VAL_AC_VOLTS_DCCOUPLED`	`0.5`	`NIDMM_VAL_10_MEGAOHM`	0.5
3	5 V	1	`NIDMM_VAL_AC_VOLTS`	`5`	`NIDMM_VAL_10_MEGAOHM`	5
	5 V	1	`NIDMM_VAL_AC_VOLTS_DCCOUPLED`	`5`	`NIDMM_VAL_10_MEGAOHM`	5
4	50 V	1	`NIDMM_VAL_AC_VOLTS`	`50`	`NIDMM_VAL_10_MEGAOHM`	50
	50 V	1	`NIDMM_VAL_AC_VOLTS_DCCOUPLED`	`50`	`NIDMM_VAL_10_MEGAOHM`	50
5	100 V	1	`NIDMM_VAL_AC_VOLTS`	`700`	`NIDMM_VAL_10_MEGAOHM`	100
	100 V	1	`NIDMM_VAL_AC_VOLTS_DCCOUPLED`	`700`	`NIDMM_VAL_10_MEGAOHM`	100

5. Refer to Table 18 for the appropriate parameter values as you complete the following steps:
   a. Generate 0 V on the calibrator.
   b. Call the niDMM Cal Adjust Offset VI with **Mode** set to `NIDMM_VAL_AC_VOLTS` and the remaining parameters as shown in Table 18 for the current iteration.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_CalAdjustOffset` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Mode**: `NIDMM_VAL_AC_VOLTS`

c. Call the niDMM Cal Adjust Offset VI again, changing **Mode** to NIDMM_VAL_AC_VOLTS_DCCOUPLED.

LabVIEW Block Diagram	C/C++ Function Call
	Call niDMM_CalAdjustOffset with the following parameters: **Instrument_Handle**: The instrument handle from niDMM_init **Mode**: NIDMM_VAL_AC_VOLTS_DCCOUPLED

6. Repeat step 5 for each of the remaining iterations shown in Table 18.

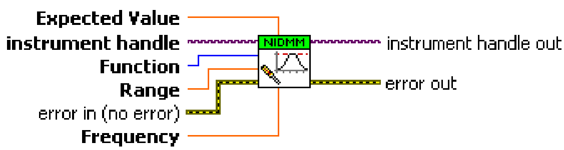
Table 18. niDMM_CalAdjustOffset Parameters

Iteration	niDMM_CalAdjustOffset Parameters		
	Mode	**Range (V)**	**Input Resistance (Ω)**
1	NIDMM_VAL_AC_VOLTS	0.05	NIDMM_VAL_10_MEGAOHM
	NIDMM_VAL_AC_VOLTS_DCCOUPLED	0.05	NIDMM_VAL_10_MEGAOHM
2	NIDMM_VAL_AC_VOLTS	0.5	NIDMM_VAL_10_MEGAOHM
	NIDMM_VAL_AC_VOLTS_DCCOUPLED	0.5	NIDMM_VAL_10_MEGAOHM
3	NIDMM_VAL_AC_VOLTS	5	NIDMM_VAL_10_MEGAOHM
	NIDMM_VAL_AC_VOLTS_DCCOUPLED	5	NIDMM_VAL_10_MEGAOHM
4	NIDMM_VAL_AC_VOLTS	50	NIDMM_VAL_10_MEGAOHM
	NIDMM_VAL_AC_VOLTS_DCCOUPLED	50	NIDMM_VAL_10_MEGAOHM
5	NIDMM_VAL_AC_VOLTS	700	NIDMM_VAL_10_MEGAOHM
	NIDMM_VAL_AC_VOLTS_DCCOUPLED	700	NIDMM_VAL_10_MEGAOHM

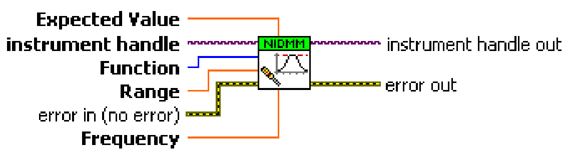
7. Refer to Table 19 for the appropriate calibrator outputs and parameter values as you complete the following steps:

    a. On the calibrator, generate the value listed under Calibrator Output in Table 19 for the current iteration.

b. Call the niDMM Cal Adjust AC Filter VI with **Mode** set to `NIDMM_VAL_AC_VOLTS`. Set the remaining parameters as shown in Table 19 for the current iteration.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_CalAdjustACFilter` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Mode**: `NIDMM_VAL_AC_VOLTS`

c. Call the niDMM Cal Adjust AC Filter VI again, changing **Mode** to `NIDMM_VAL_AC_VOLTS_DCCOUPLED`.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_CalAdjustACFilter` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Mode**: `NIDMM_VAL_AC_VOLTS_DCCOUPLED`

8. Repeat step 7 for each of the remaining iterations shown in Table 19.

Table 19. niDMM_CalAdjustACFilter Parameters

Iteration	Calibrator Output		niDMM_CalAdjustACFilter Parameters			
	Amplitude	Frequency (kHz)	Mode	Range (V)	Frequency (Hz)	Expected Value
1	50 mV	1	`NIDMM_VAL_AC_VOLTS`	`0.05`	`1e3`	0.05
	50 mV	1	`NIDMM_VAL_AC_VOLTS_DCCOUPLED`	`0.05`	`1e3`	0.05
2	50 mV	5	`NIDMM_VAL_AC_VOLTS`	`0.05`	`5e3`	0.05
	50 mV	5	`NIDMM_VAL_AC_VOLTS_DCCOUPLED`	`0.05`	`5e3`	0.05
3	50 mV	20	`NIDMM_VAL_AC_VOLTS`	`0.05`	`20e3`	0.05
	50 mV	20	`NIDMM_VAL_AC_VOLTS_DCCOUPLED`	`0.05`	`20e3`	0.05
4	50 mV	50	`NIDMM_VAL_AC_VOLTS`	`0.05`	`50e3`	0.05
	50 mV	50	`NIDMM_VAL_AC_VOLTS_DCCOUPLED`	`0.05`	`50e3`	0.05
5	50 mV	100	`NIDMM_VAL_AC_VOLTS`	`0.05`	`100e3`	0.05
	50 mV	100	`NIDMM_VAL_AC_VOLTS_DCCOUPLED`	`0.05`	`100e3`	0.05

Table 19. niDMM_CalAdjustACFilter Parameters (Continued)

Iteration	Calibrator Output		niDMM_CalAdjustACFilter Parameters			
	Amplitude	Frequency (kHz)	Mode	Range (V)	Frequency (Hz)	Expected Value
6	50 mV	200	`NIDMM_VAL_AC_VOLTS`	`0.05`	`200e3`	0.05
	50 mV	200	`NIDMM_VAL_AC_VOLTS_DCCOUPLED`	`0.05`	`200e3`	0.05
7	50 mV	300	`NIDMM_VAL_AC_VOLTS`	`0.05`	`300e3`	0.05
	50 mV	300	`NIDMM_VAL_AC_VOLTS_DCCOUPLED`	`0.05`	`300e3`	0.05
8	50 mV	500	`NIDMM_VAL_AC_VOLTS`	`0.05`	`500e3`	0.05
	50 mV	500	`NIDMM_VAL_AC_VOLTS_DCCOUPLED`	`0.05`	`500e3`	0.05
9	500 mV	1	`NIDMM_VAL_AC_VOLTS`	`0.5`	`1e3`	0.5
	500 mV	1	`NIDMM_VAL_AC_VOLTS_DCCOUPLED`	`0.5`	`1e3`	0.5
10	500 mV	5	`NIDMM_VAL_AC_VOLTS`	`0.5`	`5e3`	0.5
	500 mV	5	`NIDMM_VAL_AC_VOLTS_DCCOUPLED`	`0.5`	`5e3`	0.5
11	500 mV	20	`NIDMM_VAL_AC_VOLTS`	`0.5`	`20e3`	0.5
	500 mV	20	`NIDMM_VAL_AC_VOLTS_DCCOUPLED`	`0.5`	`20e3`	0.5
12	500 mV	50	`NIDMM_VAL_AC_VOLTS`	`0.5`	`50e3`	0.5
	500 mV	50	`NIDMM_VAL_AC_VOLTS_DCCOUPLED`	`0.5`	`50e3`	0.5
13	500 mV	100	`NIDMM_VAL_AC_VOLTS`	`0.5`	`100e3`	0.5
	500 mV	100	`NIDMM_VAL_AC_VOLTS_DCCOUPLED`	`0.5`	`100e3`	0.5
14	500 mV	200	`NIDMM_VAL_AC_VOLTS`	`0.5`	`200e3`	0.5
	500 mV	200	`NIDMM_VAL_AC_VOLTS_DCCOUPLED`	`0.5`	`200e3`	0.5
15	500 mV	300	`NIDMM_VAL_AC_VOLTS`	`0.5`	`300e3`	0.5
	500 mV	300	`NIDMM_VAL_AC_VOLTS_DCCOUPLED`	`0.5`	`300e3`	0.5
16	500 mV	500	`NIDMM_VAL_AC_VOLTS`	`0.5`	`500e3`	0.5
	500 mV	500	`NIDMM_VAL_AC_VOLTS_DCCOUPLED`	`0.5`	`500e3`	0.5
17	5 V	1	`NIDMM_VAL_AC_VOLTS`	`5`	`1e3`	5
	5 V	1	`NIDMM_VAL_AC_VOLTS_DCCOUPLED`	`5`	`1e3`	5
18	5 V	5	`NIDMM_VAL_AC_VOLTS`	`5`	`5e3`	5
	5 V	5	`NIDMM_VAL_AC_VOLTS_DCCOUPLED`	`5`	`5e3`	5
19	5 V	20	`NIDMM_VAL_AC_VOLTS`	`5`	`20e3`	5
	5 V	20	`NIDMM_VAL_AC_VOLTS_DCCOUPLED`	`5`	`20e3`	5

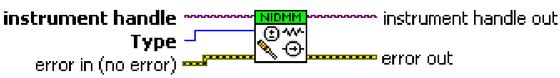
Table 19. niDMM_CalAdjustACFilter Parameters (Continued)

Iteration	Calibrator Output		niDMM_CalAdjustACFilter Parameters			
	Amplitude	Frequency (kHz)	Mode	Range (V)	Frequency (Hz)	Expected Value
20	5 V	50	`NIDMM_VAL_AC_VOLTS`	5	50e3	5
	5 V	50	`NIDMM_VAL_AC_VOLTS_DCCOUPLED`	5	50e3	5
21	5 V	100	`NIDMM_VAL_AC_VOLTS`	5	100e3	5
	5 V	100	`NIDMM_VAL_AC_VOLTS_DCCOUPLED`	5	100e3	5
22	5 V	200	`NIDMM_VAL_AC_VOLTS`	5	200e3	5
	5 V	200	`NIDMM_VAL_AC_VOLTS_DCCOUPLED`	5	200e3	5
23	5 V	300	`NIDMM_VAL_AC_VOLTS`	5	300e3	5
	5 V	300	`NIDMM_VAL_AC_VOLTS_DCCOUPLED`	5	300e3	5
24	5 V	500	`NIDMM_VAL_AC_VOLTS`	5	500e3	5
	5 V	500	`NIDMM_VAL_AC_VOLTS_DCCOUPLED`	5	500e3	5
25	50 V	1	`NIDMM_VAL_AC_VOLTS`	50	1e3	50
	50 V	1	`NIDMM_VAL_AC_VOLTS_DCCOUPLED`	50	1e3	50
26	50 V	5	`NIDMM_VAL_AC_VOLTS`	50	5e3	50
	50 V	5	`NIDMM_VAL_AC_VOLTS_DCCOUPLED`	50	5e3	50
27	50 V	20	`NIDMM_VAL_AC_VOLTS`	50	20e3	50
	50 V	20	`NIDMM_VAL_AC_VOLTS_DCCOUPLED`	50	20e3	50
28	50 V	50	`NIDMM_VAL_AC_VOLTS`	50	50e3	50
	50 V	50	`NIDMM_VAL_AC_VOLTS_DCCOUPLED`	50	50e3	50
29	50 V	100	`NIDMM_VAL_AC_VOLTS`	50	100e3	50
	50 V	100	`NIDMM_VAL_AC_VOLTS_DCCOUPLED`	50	100e3	50
30	50 V	200	`NIDMM_VAL_AC_VOLTS`	50	200e3	50
	50 V	200	`NIDMM_VAL_AC_VOLTS_DCCOUPLED`	50	200e3	50
31	50 V	300	`NIDMM_VAL_AC_VOLTS`	50	300e3	50
	50 V	300	`NIDMM_VAL_AC_VOLTS_DCCOUPLED`	50	300e3	50
32	10 V	500	`NIDMM_VAL_AC_VOLTS`	50	500e3	10
	10 V	500	`NIDMM_VAL_AC_VOLTS_DCCOUPLED`	50	500e3	10
33	100 V	1	`NIDMM_VAL_AC_VOLTS`	700	1e3	100
	100 V	1	`NIDMM_VAL_AC_VOLTS_DCCOUPLED`	700	1e3	100

Table 19. niDMM_CalAdjustACFilter Parameters (Continued)

Iteration	Calibrator Output		niDMM_CalAdjustACFilter Parameters			
	Amplitude	Frequency (kHz)	Mode	Range (V)	Frequency (Hz)	Expected Value
34	100 V	5	`NIDMM_VAL_AC_VOLTS`	`700`	`5e3`	100
	100 V	5	`NIDMM_VAL_AC_VOLTS_DCCOUPLED`	`700`	`5e3`	100
35	100 V	20	`NIDMM_VAL_AC_VOLTS`	`700`	`20e3`	100
	100 V	20	`NIDMM_VAL_AC_VOLTS_DCCOUPLED`	`700`	`20e3`	100
36	100 V	50	`NIDMM_VAL_AC_VOLTS`	`700`	`50e3`	100
	100 V	50	`NIDMM_VAL_AC_VOLTS_DCCOUPLED`	`700`	`50e3`	100
37	100 V	100	`NIDMM_VAL_AC_VOLTS`	`700`	`100e3`	100
	100 V	100	`NIDMM_VAL_AC_VOLTS_DCCOUPLED`	`700`	`100e3`	100
38	100 V	200	`NIDMM_VAL_AC_VOLTS`	`700`	`200e3`	100
	100 V	200	`NIDMM_VAL_AC_VOLTS_DCCOUPLED`	`700`	`200e3`	100
39	50 V	300	`NIDMM_VAL_AC_VOLTS`	`700`	`300e3`	50
	50 V	300	`NIDMM_VAL_AC_VOLTS_DCCOUPLED`	`700`	`300e3`	50
40	10 V	500	`NIDMM_VAL_AC_VOLTS`	`700`	`500e3`	10
	10 V	500	`NIDMM_VAL_AC_VOLTS_DCCOUPLED`	`700`	`500e3`	10

9. Reset the calibrator for safety reasons.
10. Call the niDMM Cal Adjust Miscellaneous VI.

<table>
<tr><th>LabVIEW Block Diagram</th><th>C/C++ Function Call</th></tr>
<tr><td>


</td><td>Call <code>niDMM_CalAdjustMisc</code> with the following parameters:

Instrument_Handle: The instrument handle from <code>niDMM_init</code>
Type: <code>NIDMM_EXTCAL_MISCCAL_SECTION</code></td></tr>
</table>

You have completed adjusting the AC voltage modes of the NI 4071. Select one of the following options:

- If you are performing additional adjustments, refer to the following section, as applicable:
  - *Adjusting Current Modes*
- If you are *not* performing additional adjustments, refer to one of the following sections:
  - *Verification Procedures*—to verify your new calibration coefficients before saving them to the EEPROM
  - *Completing the Adjustment Procedures*—if you do *not* want to verify the adjustments you have just made

Adjusting Current Modes

If you do not use the current modes (DC and AC), or the accuracy is insignificant for your application, you can skip this section and select one of the following options:

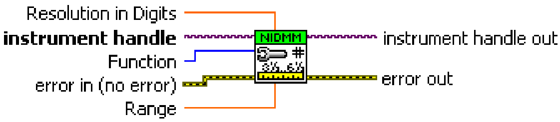
- If you skip this section and you want to verify the new calibration coefficients before saving them to the EEPROM, repeat the *Verification Procedures* section (except for *Verifying DC Voltage*).
- If you skip this section and you do not want to verify the new calibration coefficients, go to the *Completing the Adjustment Procedures* section.

Note The 90-day DC current uncertainty of the Fluke 5720A is not adequate to calibrate the four lowest DC current ranges listed in Table 25. You must characterize the calibrator output to better than published specifications. Use metrology laboratory methods to obtain the required uncertainty. Monitor the current applied using a DC current shunt/voltmeter system in series with the calibrator output. Refer to Table 13 for the recommended DC current uncertainty requirements for the DC current shunt/voltmeter system.

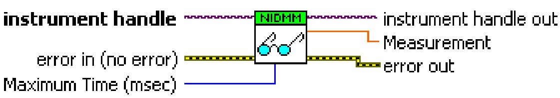
To adjust the current modes of the NI 4071, complete the following steps:

1. Reset the calibrator
2. Connect the NI 4071 and the Fluke 5720A calibrator using the Fluke 5440 cable, as shown in Figure 6. Table 15 lists the cable connections.
3. Wait two minutes for the thermal EMF to stabilize.

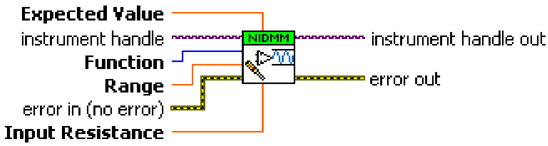
4. Call the niDMM Config Measurement VI and select the **Resolution in Digits** instance.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_ConfigureMeasurementDigits` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Resolution_Digits**: `6.5` **Measurement_Function**: `NIDMM_VAL_DC_CURRENT` **Range**: `1e-6`

5. Call the niDMM Read VI. Configure the NI 4071 for a current mode before applying current.

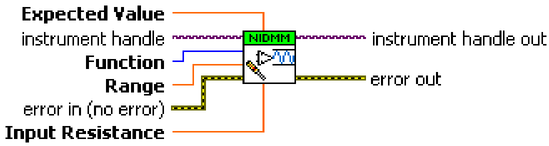
LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_read` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Reading**: The measurement returned by the function. Configure the NI 4071 for a current mode before applying current. **Maximum_Time**: `-1`

6. Generate 1 µA on the calibrator with the current output set to `NORM`.
7. Call the niDMM Cal Adjust Gain VI with the following parameters:

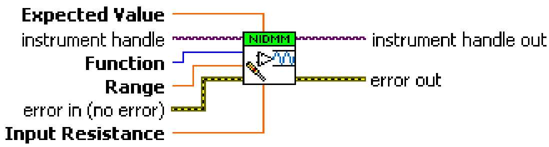
LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_CalAdjustGain` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Mode**: `NIDMM_VAL_DC_CURRENT` **Range**: `1e-6` **Input_Resistance**: `NIDMM_VAL_RESISTANCE_NA` **Expected_Value**: `1e-6`

8. Generate –1 µA on the calibrator with the current output set to `NORM`.

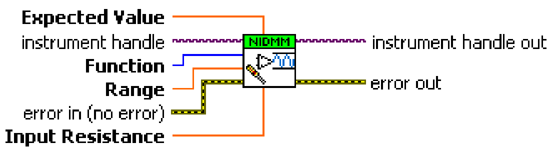
9. Call the niDMM Cal Adjust Gain VI with the following parameters:

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_CalAdjustGain` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Mode**: `NIDMM_VAL_DC_CURRENT` **Range**: `1e-6` **Input_Resistance**: `NIDMM_VAL_RESISTANCE_NA` **Expected_Value**: `-1e-6`

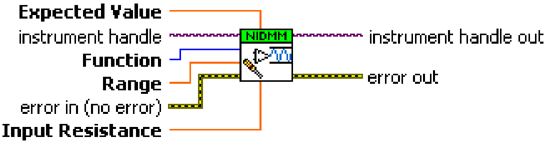
10. Generate 10 µA on the calibrator with the current output set to `NORM`.
11. Call the niDMM Cal Adjust Gain VI with the following parameters:

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_CalAdjustGain` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Mode**: `NIDMM_VAL_DC_CURRENT` **Range**: `10e-6` **Input_Resistance**: `NIDMM_VAL_RESISTANCE_NA` **Expected_Value**: `10e-6`

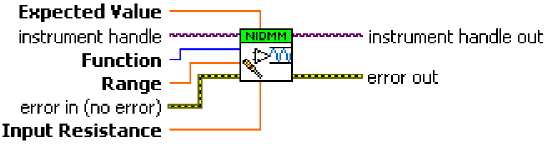
12. Generate –10 µA on the calibrator with the current output set to `NORM`.
13. Call the niDMM Cal Adjust Gain VI with the following parameters:

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_CalAdjustGain` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Mode**: `NIDMM_VAL_DC_CURRENT` **Range**: `10e-6` **Input_Resistance**: `NIDMM_VAL_RESISTANCE_NA` **Expected_Value**: `-10e-6`

14. Connect the Fluke 5720A calibrator to the NI 4071 with a DC current shunt/voltmeter system in series, as shown in Figure 5. Table 14 lists the cable connections. Use the shunt recommended for ±100 µA as indicated in Table 13.
15. Wait two minutes for the thermal EMF to stabilize.
16. Generate 100 µA on the calibrator with the current output set to `NORM`.
17. Call the niDMM Cal Adjust Gain VI with the following parameters:

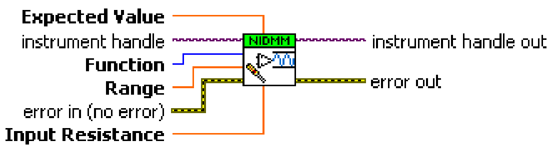
LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_CalAdjustGain` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Mode**: `NIDMM_VAL_DC_CURRENT` **Range**: `100e-6` **Input_Resistance**: `NIDMM_VAL_RESISTANCE_NA` **Expected_Value**: The characterized 100 µA value.

18. Generate –100 µA on the calibrator with the current output set to `NORM`.
19. Call the niDMM Cal Adjust Gain VI with the following parameters:

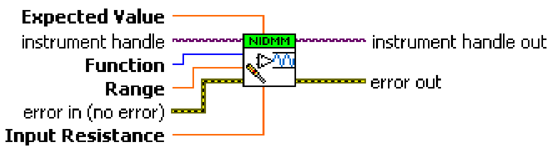
LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_CalAdjustGain` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Mode**: `NIDMM_VAL_DC_CURRENT` **Range**: `100e-6` **Input_Resistance**: `NIDMM_VAL_RESISTANCE_NA` **Expected_Value**: The characterized –100 µA value.

20. Replace the shunt recommended for 100 µA currents with the shunt recommended for 1 mA as indicated in Table 13.
21. Generate 1 mA on the calibrator with the current output set to `NORM`.

22. Call the niDMM Cal Adjust Gain VI with the following parameters:

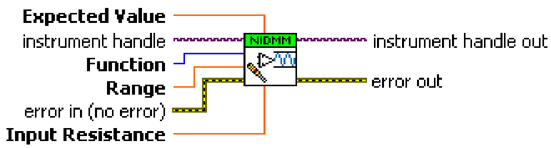
LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_CalAdjustGain` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Mode**: `NIDMM_VAL_DC_CURRENT` **Range**: `1e-3` **Input_Resistance**: `NIDMM_VAL_RESISTANCE_NA` **Expected_Value**: The characterized 1 mA value.

23. Generate –1 mA on the calibrator with the current output set to `NORM`.
24. Call the niDMM Cal Adjust Gain VI with the following parameters:

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_CalAdjustGain` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Mode**: `NIDMM_VAL_DC_CURRENT` **Range**: `1e-3` **Input_Resistance**: `NIDMM_VAL_RESISTANCE_NA` **Expected_Value**: The characterized –1 mA value.

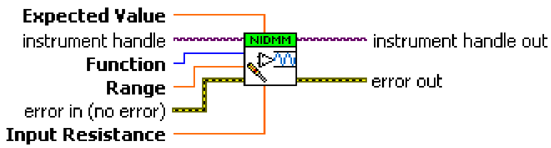
25. Connect the NI 4071 and the Fluke 5720A calibrator using the Fluke 5440 cable, as shown in Figure 6. Table 15 lists the cable connections.
26. Wait two minutes for the thermal EMF to stabilize.
27. Generate 10 mA on the calibrator with the current output set to `NORM`.

28. Call the niDMM Cal Adjust Gain VI with the following parameters:

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_CalAdjustGain` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Mode**: `NIDMM_VAL_DC_CURRENT` **Range**: `0.01` **Input_Resistance**: `NIDMM_VAL_RESISTANCE_NA` **Expected_Value**: `0.01`

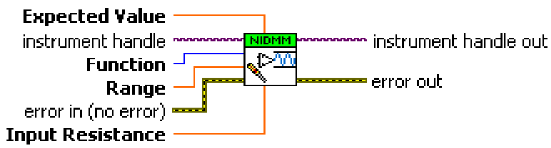
29. Generate –10 mA on the calibrator with the current output set to `NORM`.

30. Call the niDMM Cal Adjust Gain VI with the following parameters:

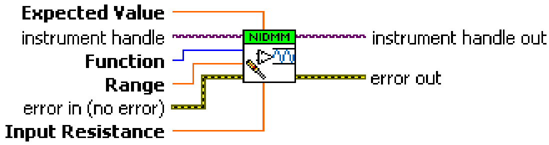
LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_CalAdjustGain` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Mode**: `NIDMM_VAL_DC_CURRENT` **Range**: `0.01` **Input_Resistance**: `NIDMM_VAL_RESISTANCE_NA` **Expected_Value**: `–0.01`

31. Generate 100 mA on the calibrator with the current output set to `NORM`.

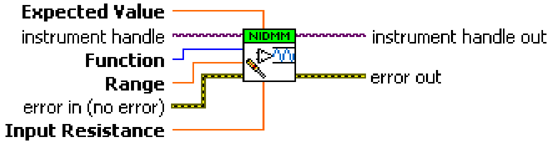
32. Call the niDMM Cal Adjust Gain VI with the following parameters:

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_CalAdjustGain` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Mode**: `NIDMM_VAL_DC_CURRENT` **Range**: `0.1` **Input_Resistance**: `NIDMM_VAL_RESISTANCE_NA` **Expected_Value**: `0.1`

33. Generate –100 mA on the calibrator with the current output set to `NORM`.
34. Call the niDMM Cal Adjust Gain VI with the following parameters:

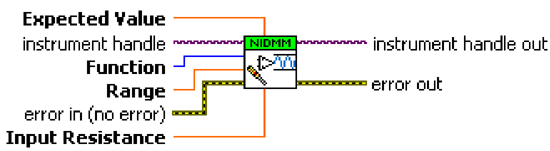
LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_CalAdjustGain` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Mode**: `NIDMM_VAL_DC_CURRENT` **Range**: `0.1` **Input_Resistance**: `NIDMM_VAL_RESISTANCE_NA` **Expected_Value**: `–0.1`

35. Generate 1 A on the calibrator with the current output set to `NORM`.
36. Call the niDMM Cal Adjust Gain VI with the following parameters:

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_CalAdjustGain` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Mode**: `NIDMM_VAL_DC_CURRENT` **Range**: `1` **Input_Resistance**: `NIDMM_VAL_RESISTANCE_NA` **Expected_Value**: `1`

37. Generate –1 A on the calibrator with the current output set to `NORM`.

38. Call the niDMM Cal Adjust Gain VI with the following parameters:

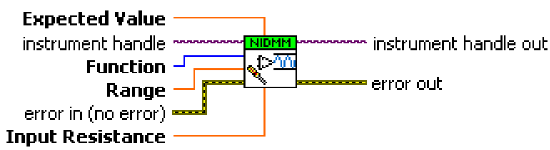
<table>
<tr><th>LabVIEW Block Diagram</th><th>C/C++ Function Call</th></tr>
<tr><td>


</td><td>
Call <code>niDMM_CalAdjustGain</code> with the following parameters:

Instrument_Handle: The instrument handle from <code>niDMM_init</code>
Mode: <code>NIDMM_VAL_DC_CURRENT</code>
Range: <code>1</code>
Input_Resistance: <code>NIDMM_VAL_RESISTANCE_NA</code>
Expected_Value: –<code>1</code>
</td></tr>
</table>

39. Generate 2.2 A on the calibrator with the current output set to `NORM`.

40. Call the niDMM Cal Adjust Gain VI with the following parameters:

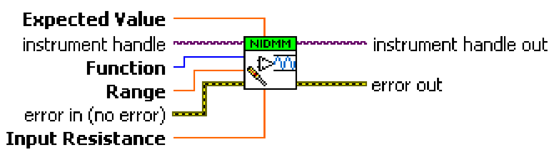
<table>
<tr><th>LabVIEW Block Diagram</th><th>C/C++ Function Call</th></tr>
<tr><td>


</td><td>
Call <code>niDMM_CalAdjustGain</code> with the following parameters:

Instrument_Handle: The instrument handle from <code>niDMM_init</code>
Mode: <code>NIDMM_VAL_DC_CURRENT</code>
Range: <code>3</code>
Input_Resistance: <code>NIDMM_VAL_RESISTANCE_NA</code>
Expected_Value: <code>2.2</code>
</td></tr>
</table>

41. Generate –2.2 A on the calibrator with the current output set to `NORM`.

42. Call the niDMM Cal Adjust Gain VI with the following parameters:

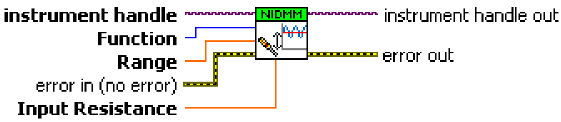
<table>
<tr><th>LabVIEW Block Diagram</th><th>C/C++ Function Call</th></tr>
<tr><td>


</td><td>
Call <code>niDMM_CalAdjustGain</code> with the following parameters:

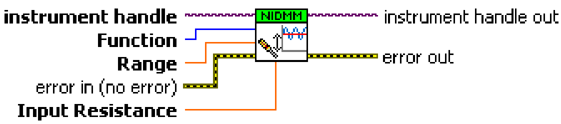
Instrument_Handle: The instrument handle from <code>niDMM_init</code>
Mode: <code>NIDMM_VAL_DC_CURRENT</code>
Range: <code>3</code>
Input_Resistance: <code>NIDMM_VAL_RESISTANCE_NA</code>
Expected_Value: –<code>2.2</code>
</td></tr>
</table>

43. Remove all connections from the four input banana plug connectors on the NI 4071.

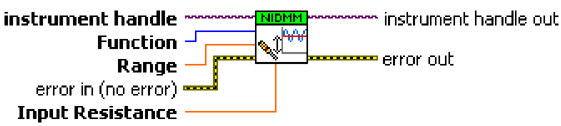
44. Call the niDMM Cal Adjust Offset VI with the following parameters:

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_CalAdjustOffset` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Mode**: `NIDMM_VAL_DC_CURRENT` **Range**: `1e–6` **Input_Resistance**: `NIDMM_VAL_RESISTANCE_NA`

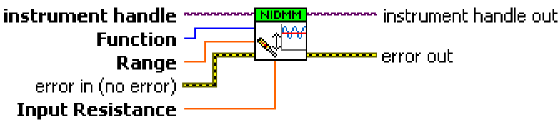
45. Call the niDMM Cal Adjust Offset VI with the following parameters:

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_CalAdjustOffset` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Mode**: `NIDMM_VAL_DC_CURRENT` **Range**: `10e–6` **Input_Resistance**: `NIDMM_VAL_RESISTANCE_NA`

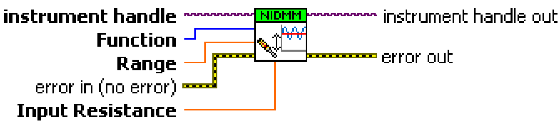
46. Call the niDMM Cal Adjust Offset VI with the following parameters:

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_CalAdjustOffset` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Mode**: `NIDMM_VAL_DC_CURRENT` **Range**: `100e–6` **Input_Resistance**: `NIDMM_VAL_RESISTANCE_NA`

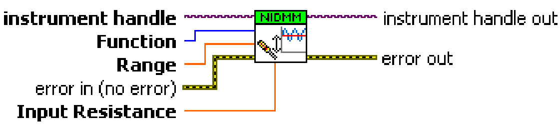
47. Call the niDMM Cal Adjust Offset VI with the following parameters:

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_CalAdjustOffset` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Mode**: `NIDMM_VAL_AC_CURRENT` **Range**: `100e–6` **Input_Resistance**: `NIDMM_VAL_RESISTANCE_NA`

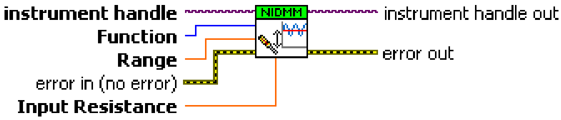
48. Call the niDMM Cal Adjust Offset VI with the following parameters:

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_CalAdjustOffset` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Mode**: `NIDMM_VAL_DC_CURRENT` **Range**: `1e–3` **Input_Resistance**: `NIDMM_VAL_RESISTANCE_NA`

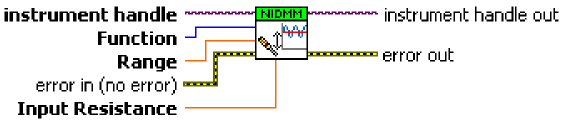
49. Call the niDMM Cal Adjust Offset VI with the following parameters:

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_CalAdjustOffset` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Mode**: `NIDMM_VAL_AC_CURRENT` **Range**: `1e–3` **Input_Resistance**: `NIDMM_VAL_RESISTANCE_NA`

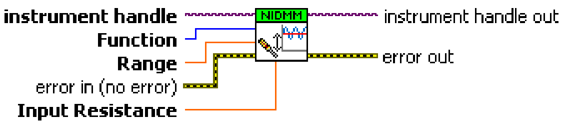
50. Call the niDMM Cal Adjust Offset VI with the following parameters:

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_CalAdjustOffset` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Mode**: `NIDMM_VAL_DC_CURRENT` **Range**: `0.01` **Input_Resistance**: `NIDMM_VAL_RESISTANCE_NA`

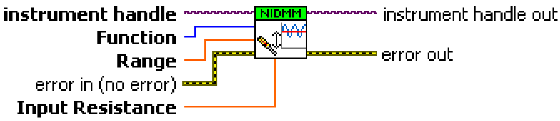
51. Call the niDMM Cal Adjust Offset VI with the following parameters:

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_CalAdjustOffset` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Mode**: `NIDMM_VAL_AC_CURRENT` **Range**: `0.01` **Input_Resistance**: `NIDMM_VAL_RESISTANCE_NA`

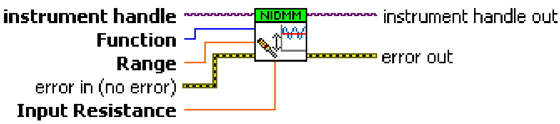
52. Call the niDMM Cal Adjust Offset VI with the following parameters:

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_CalAdjustOffset` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Mode**: `NIDMM_VAL_DC_CURRENT` **Range**: `0.1` **Input_Resistance**: `NIDMM_VAL_RESISTANCE_NA`

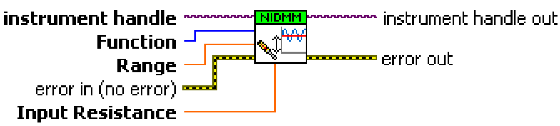
53. Call the niDMM Cal Adjust Offset VI with the following parameters:

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_CalAdjustOffset` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Mode**: `NIDMM_VAL_AC_CURRENT` **Range**: `0.1` **Input_Resistance**: `NIDMM_VAL_RESISTANCE_NA`

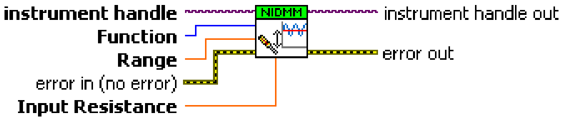
54. Call the niDMM Cal Adjust Offset VI with the following parameters:

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_CalAdjustOffset` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Mode**: `NIDMM_VAL_DC_CURRENT` **Range**: `1` **Input_Resistance**: `NIDMM_VAL_RESISTANCE_NA`

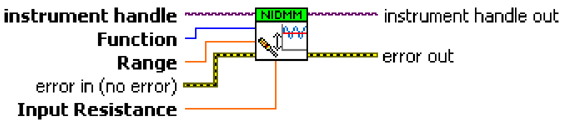
55. Call the niDMM Cal Adjust Offset VI with the following parameters:

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_CalAdjustOffset` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Mode**: `NIDMM_VAL_AC_CURRENT` **Range**: `1` **Input_Resistance**: `NIDMM_VAL_RESISTANCE_NA`

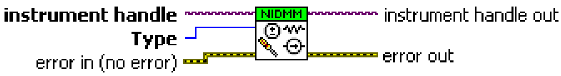
56. Call the niDMM Cal Adjust Offset VI with the following parameters:

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_CalAdjustOffset` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Mode**: `NIDMM_VAL_DC_CURRENT` **Range**: `3` **Input_Resistance**: `NIDMM_VAL_RESISTANCE_NA`

57. Call the niDMM Cal Adjust Offset VI with the following parameters:

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_CalAdjustOffset` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Mode**: `NIDMM_VAL_AC_CURRENT` **Range**: `3` **Input_Resistance**: `NIDMM_VAL_RESISTANCE_NA`

58. Call the niDMM Cal Adjust Miscellaneous VI.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_CalAdjustMisc` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Type**: `NIDMM_EXTCAL_MISCCAL_SECTION`

You have completed adjusting the current modes of the NI 4071. Select one of the following options:

- To verify your new calibration coefficients before saving them to the EEPROM, refer to the *Verification Procedures*.
- If you do *not* want to verify the adjustments you have just made, refer to the *Completing the Adjustment Procedures* section.

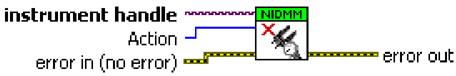
Completing the Adjustment Procedures

To complete the adjustment procedure for the NI 4071 and close the session, call the niDMM Close External Cal VI with the following parameter:

- **Action** = `NIDMM_EXTCAL_ACTION_SAVE` if the results of the calibration were satisfactory and you want to save the new calibration coefficients to the EEPROM.

  Otherwise,

- **Action** = `NIDMM_EXTCAL_ACTION_ABORT` if the results of the calibration were unsatisfactory and you want to restore the original calibration coefficients to the EEPROM.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niDMM_CloseExtCal` with the following parameters: **Instrument_Handle**: The instrument handle from `niDMM_init` **Action**: `NIDMM_EXTCAL_ACTION_SAVE` or `NIDMM_EXTCAL_ACTION_ABORT`

Verification Limits

This section includes the verification limits for DC voltage, AC voltage, 4-wire resistance, 2-wire resistance, DC current, AC current, and frequency for the NI 4071. Compare these limits to the results you obtain in the *Verification Procedures* section.

Note Use the 24-Hour Limits for a post-adjustment verification *only*. Otherwise, use the 2-Year Limits.

Limits in the following tables are based upon the September 2008 edition of the *NI 4071 Specifications*. Refer to the most recent NI 4071 specifications online at `ni.com/manuals`. If a more recent edition of the specifications is available, recalculate the limits based upon the latest specifications.

Calibrator uncertainty is included in the 24-hour limit calculations.

DC Voltage

Table 20. NI 4071 DC Voltage Verification Limits

Calibrator Amplitude	Range	Input Resistance	2-Year Limits		24-Hour Limits	
			Lower	Upper	Lower	Upper
0 V	100 mV	>10 GΩ/10 MΩ	–0.8 µV	0.8 µV	–0.4 µV	0.4 µV
0 V	1 V	>10 GΩ/10 MΩ	–2.1 µV	2.1 µV	–2.1 µV	2.1 µV
0 V	10 V	>10 GΩ/10 MΩ	–5 µV	5 µV	–5 µV	5 µV
0 V	100 V	10 MΩ	–200 µV	200 µV	–200 µV	200 µV
0 V	1000 V	10 MΩ	–500 µV	500 µV	–500 µV	500 µV
90 mV	100 mV	>10 GΩ/10 MΩ	89.9974 mV	90.0026 mV	89.99821 mV	90.00179 mV
–90 mV	100 mV	>10 GΩ/10 MΩ	–90.0026 mV	–89.9974 mV	–90.00179 mV	–89.99821 mV
900 mV	1 V	>10 GΩ/10 MΩ	899.9844 mV	900.0156 mV	899.9905 mV	900.0096 mV
–900 mV	1 V	>10 GΩ/10 MΩ	–900.0156 mV	–899.9844 mV	–900.0096 mV	–899.9905 mV
9 V	10 V	>10 GΩ/10 MΩ	8.999887 V	9.000113 V	8.999952 V	9.000048 V
5 V	10 V	>10 GΩ	4.999935 V	5.000065 V	4.99997 V	5.00003 V
2.5 V	10 V	>10 GΩ	2.499965 V	2.500035 V	2.499981 V	2.500019 V
–2.5 V	10 V	>10 GΩ	–2.500035 V	–2.499965 V	–2.500019 V	–2.499981 V
–5 V	10 V	>10 GΩ	–5.000065 V	–4.999935 V	–5.00003 V	–4.99997 V
–9 V	10 V	>10 GΩ/10 MΩ	–9.000113 V	–8.999887 V	–9.000048 V	–8.999952 V
90 V	100 V	10 MΩ	89.998 V	90.002 V	89.999 V	90.00101 V
–90 V	100 V	10 MΩ	–90.00200 V	–89.99800 V	–90.00101 V	–89.999 V
1000 V	1000 V	10 MΩ	999.9545 V	1000.0455 V	999.9656 V	1000.0344 V
–1000 V	1000 V	10 MΩ	–1000.0455 V	–999.9545 V	–1000.0344 V	–999.9656 V

AC Voltage

Table 21. NI 4071 AC Voltage Verification Limits

Calibrator Output		Range	Coupling	2-Year Limits	
Amplitude	Frequency			Lower	Upper
4.5 mV	1 kHz	50 mV	AC/DC	4.4878 mV	4.5123 mV
45 mV	30 Hz	50 mV	DC	44.9450 mV	45.0550 mV
45 mV	50 Hz	50 mV	AC/DC	44.9675 mV	45.0325 mV

Table 21. NI 4071 AC Voltage Verification Limits (Continued)

Calibrator Output		Range	Coupling	2-Year Limits	
Amplitude	Frequency			Lower	Upper
45 mV	1 kHz	50 mV	AC/DC	44.9675 mV	45.0325 mV
45 mV	1 kHz	500 mV	AC/DC	44.953 mV	45.048 mV
45 mV	20 kHz	50 mV	AC/DC	44.9675 mV	45.0325 mV
45 mV	50 kHz	50 mV	AC/DC	44.9585 mV	45.0415 mV
45 mV	100 kHz	50 mV	AC/DC	44.8550 mV	45.1450 mV
45 mV	300 kHz	50 mV	AC/DC	44.6100 mV	45.3900 mV
450 mV	30 Hz	500 mV	DC	449.525 mV	450.475 mV
450 mV	50 Hz	500 mV	AC/DC	449.750 mV	450.250 mV
450 mV	1 kHz	500 mV	AC/DC	449.750 mV	450.250 mV
450 mV	1 kHz	5 V	AC/DC	449.53 mV	450.48 mV
450 mV	20 kHz	500 mV	AC/DC	449.750 mV	450.250 mV
450 mV	50 kHz	500 mV	AC/DC	449.680 mV	450.320 mV
450 mV	100 kHz	500 mV	AC/DC	449.050 mV	450.950 mV
450 mV	300 kHz	500 mV	AC/DC	446.100 mV	453.900 mV
4.5 V	30 Hz	5 V	DC	4.49525 V	4.50475 V
4.5 V	50 Hz	5 V	AC/DC	4.49750 V	4.50250 V
1 V	1 kHz	5 V	AC/DC	0.99925 V	1.00075 V
2 V	1 kHz	5 V	AC/DC	1.99875 V	2.00125 V
3 V	1 kHz	5 V	AC/DC	2.99825 V	3.00175 V
4.5 V	1 kHz	5 V	AC/DC	4.49750 V	4.50250 V
4.5 V	1 kHz	50 V	AC/DC	4.4923 V	4.5077 V
4.5 V	20 kHz	5 V	AC/DC	4.49750 V	4.50250 V
4.5 V	50 kHz	5 V	AC/DC	4.49680 V	4.50320 V
4.5 V	100 kHz	5 V	AC/DC	4.49050 V	4.50950 V
4.5 V	300 kHz	5 V	AC/DC	4.46100 V	4.53900 V
45 V	30 Hz	50 V	DC	44.9525 V	45.0475 V
45 V	50 Hz	50 V	AC/DC	44.9680 V	45.0320 V
45 V	1 kHz	50 V	AC/DC	44.9680 V	45.0320 V
45 V	1 kHz	700 V	AC/DC	44.903 V	45.097 V
45 V	20 kHz	50 V	AC/DC	44.9680 V	45.0320 V

Table 21. NI 4071 AC Voltage Verification Limits (Continued)

Calibrator Output		Range	Coupling	2-Year Limits	
Amplitude	Frequency			Lower	Upper
45 V	50 kHz	50 V	AC/DC	44.9210 V	45.0790 V
45 V	100 kHz	50 V	AC/DC	44.7050 V	45.2950 V
45 V	300 kHz	50 V	AC/DC	43.5750 V	46.4250 V
200 V	30 Hz	700 V	DC	199.765 V	200.235 V
630 V	50 Hz	700 V	AC/DC	629.552 V	630.448 V
630 V	1 kHz	700 V	AC/DC	629.552 V	630.448 V
200 V	20 kHz	700 V	AC/DC	199.810 V	200.190 V
200 V	50 kHz	700 V	AC/DC	199.410 V	200.590 V
200 V	100 kHz	700 V	AC/DC	198.450 V	201.550 V
70 V	300 kHz	700 V	AC/DC	66.850 V	73.150 V

4-Wire Resistance

Note Tolerances are relative to the reported calibrator resistance value. For example, the following values and equations demonstrate the 4-wire verification tolerance calculations at 10 MΩ.

Table 22. Example of 4-wire Verification Tolerance Calculations at 10 MΩ

Parameter	Value
Fluke 5720A nominal value	10 MΩ
Fluke 5720A displayed value	10.10604 MΩ
NI 4071 range	10 MΩ
NI 4071 two-year specification in ppm of calibrator resistance	90 ppm
NI 4071 two-year specification in ppm of range	10 ppm
Tolerance in MΩ	(10.10604 * 90 / 1,000,000) + (10 * 10 / 1,000,000) = 0.00101
Lower limit	10.10604 – 0.00101 = 10.10503 MΩ
Upper limit	10.10604 + 0.00101 = 10.10705 MΩ

Table 23. NI 4071 4-Wire Resistance Verification Tolerances

Calibrator Resistance	Range	2-Year Tolerance ± (ppm of Calibrator Resistance + ppm of Range)	24-Hour Tolerance ± (NI 4071 Specification + Fluke 5720 Specification)	
			NI 4071 Specification ± (ppm of Calibrator Resistance + ppm of Range)	Fluke 5720 Specification* ± (ppm of Output)
10 MΩ	10 MΩ	90 + 10	60 +5	31
1 MΩ	1 MΩ	52 + 1	5 + 1	15
100 kΩ	100 kΩ	50 + 8	5 + 2	9
10 kΩ	10 kΩ	48 + 0.5	5 + 0.5	7.5
1 kΩ	1 kΩ	48 + 0.5	5 + 0.5	7.5
100 Ω	100 Ω	56 + 4	8 + 2.5	9
0 Ω	10 MΩ	0 + 10	0 + 5	0
0 Ω	1 MΩ	0 + 1	0 + 1	0
0 Ω	100 kΩ	0 + 8	0 + 2	0
0 Ω	10 kΩ	0 + 0.5	0 + 0.5	0

Table 23. NI 4071 4-Wire Resistance Verification Tolerances (Continued)

Calibrator Resistance	Range	2-Year Tolerance ± (ppm of Calibrator Resistance + ppm of Range)	24-Hour Tolerance ± (NI 4071 Specification + Fluke 5720 Specification)	
			NI 4071 Specification ± (ppm of Calibrator Resistance + ppm of Range)	Fluke 5720 Specification[*] ± (ppm of Output)
0 Ω	1 kΩ	0 + 0.5	0 + 0.5	40 μΩ
0 Ω	100 Ω	0 + 4	0 + 2.5	40 μΩ

[*] The Fluke 5720 Specification is a 90-day 95% confidence specification that is included in the 24-Hour Tolerance.

2-Wire Resistance

Note Tolerances are relative to the reported calibrator resistance value. For an example calculation of the verification tolerance, refer to Table 22.

Table 24. NI 4071 2-Wire Resistance Verification Tolerances

Calibrator Resistance	Range	2-Year Tolerance (ppm of Calibrator Resistance + ppm of Range)	24-Hour Tolerance ± (NI 4071 Specification + Fluke 5720 Specification)	
			NI 4071 Specification ± (ppm of Calibrator Resistance + ppm of Range)	Fluke 5720 Specification[*] ± (ppm of Output)
0 Ω	100 MΩ	0 + 20	0 + 6	N/A
0 Ω	10 MΩ	0 + 10	0 + 5	N/A
0 Ω	1 MΩ	0 + 1.2	0 + 1.2	N/A
0 Ω	100 kΩ	0 + 10	0 + 4	N/A
0 Ω	10 kΩ	0 + 20.5	0 + 20.5	N/A
0 Ω	1 kΩ	0 + 200.5	0 + 200.5	N/A
0 Ω	100 Ω	0 + 2004	0 + 2002.5	N/A
1 GΩ[†]	5 GΩ	50,000 + 2,000	10,000 + 2,000	Insert the uncertainty value for the standard resistor you are using.
100 MΩ	100 MΩ	6,000 + 20	500 + 6	95
10 MΩ	10 MΩ	90 + 10	60 + 5	31
1 MΩ	1 MΩ	52 + 1.2	5 + 1.2	15
100 kΩ	100 kΩ	50 + 10	5 + 4	9
10 kΩ	10 kΩ	48 + 20.5	5 + 20.5	12.5[‡]

Table 24. NI 4071 2-Wire Resistance Verification Tolerances (Continued)

Calibrator Resistance	Range	2-Year Tolerance (ppm of Calibrator Resistance + ppm of Range)	24-Hour Tolerance ± (NI 4071 Specification + Fluke 5720 Specification)	
			NI 4071 Specification ± (ppm of Calibrator Resistance + ppm of Range)	Fluke 5720 Specification* ± (ppm of Output)
1 kΩ	1 kΩ	48 + 200.5	5 + 200.5	17.5‡
100 Ω	100 Ω	56 + 2004	8 + 2002.5	67.5‡

* The Fluke 5720 Specification is a 90-day 95% confidence specification that is included in the 24-Hour Tolerance.

† Measured with a 1 GΩ standard resistor.

‡ Fluke 5720A specifications include derating for under current adder and use of calibrator active compensation. These specifications should be combined with both the 2-year and 24-hour tolerance of the PXI-4071.

DC Current

Note The 90-day DC current uncertainty of the Fluke 5720A is not adequate to calibrate the 100 µA and 1 mA DC current ranges listed in Table 25. You must characterize the calibrator output to better than published specifications. Use metrology laboratory methods to obtain the required uncertainty. Monitor the current applied using a DC current shunt/voltmeter system in series with the calibrator output. Refer to Table 13 for the recommended DC current uncertainty requirements for the DC current shunt/voltmeter system.

Note Tolerances are relative to the nominal calibrator amplitude unless otherwise noted.

Table 25. NI 4071 DC Current Verification Tolerances

Calibrator Amplitude	Range	2-Year Tolerance (ppm of Calibrator Amplitude + ppm of Range)	24-Hour Tolerance ± (NI 4071 Specification + Fluke 5720 Specification)	
			NI 4071 Specification ± (ppm of Calibrator Amplitude + ppm of Range)	Fluke 5720 Specification† ± (ppm of Output)
0 A	100 µA	0 + 20	0 + 20	N/A
0 A	1 mA	0 + 20	0 + 20	N/A
0 A	10 mA	0 + 20	0 + 20	N/A
0 A	100 mA	0 + 20	0 + 20	N/A
0 A	1 A	0 + 20	0 + 20	N/A
0 A	3 A	0 + 30	0 + 30	N/A
100 µA	100 µA	100 + 20*	10 + 20*	40
–100 µA	100 µA	100 + 20*	10 + 20*	40

Table 25. NI 4071 DC Current Verification Tolerances

Calibrator Amplitude	Range	2-Year Tolerance (ppm of Calibrator Amplitude + ppm of Range)	24-Hour Tolerance ± (NI 4071 Specification + Fluke 5720 Specification)	
			NI 4071 Specification ± (ppm of Calibrator Amplitude + ppm of Range)	Fluke 5720 Specification[†] ± (ppm of Output)
1 mA	1 mA	100 + 20[*]	4 + 20[*]	40
–1 mA	1 mA	100 + 20[*]	4 + 20[*]	40
9 mA	10 mA	110 + 20	12 + 20	30
-9 mA	10 mA	110 + 20	12 + 20	30
90 mA	100 mA	165 + 20	9 + 20	40
–90 mA	100 mA	165 + 20	9 + 20	40
900 mA	1 A	290 + 20	15 + 20	60
500 mA	1 A	290 + 20	15 + 20	60
250 mA	1 A	290 + 20	15 + 20	60
–250 mA	1 A	290 + 20	15 + 20	60
–500 mA	1 A	290 + 20	15 + 20	60
–900 mA	1 A	290 + 20	15 + 20	60
2.2 A	3 A	740 + 30	15 + 30	60
–2.2 A	3 A	740 + 30	15 + 30	60

[*] Limits are relative to the characterized values of the Fluke 5720A calibrator amplitude.

[†] The Fluke 5720 Specification is a 90-day 95% confidence specification that is included in the 24-Hour Tolerance.

AC Current

Table 26. NI 4071 AC Current Verification Limits

Calibrator Output		Range	2-Year Limits	
Amplitude	Frequency		Lower	Upper
9 µA	1 kHz	100 µA	8.9773 µA	9.0227 µA
90 µA	1 kHz	100 µA	89.953 µA	90.047 µA
90 µA	1 kHz	1 mA	89.773 µA	90.227 µA
900 µA	1 kHz	1 mA	899.530 µA	900.470 µA
900 µA	1 kHz	10 mA	897.73 µA	902.27 µA
9 mA	1 kHz	10 mA	8.99530 mA	9.00470 mA
9 mA	1 kHz	100 mA	8.9773 mA	9.0227 mA

Table 26. NI 4071 AC Current Verification Limits (Continued)

Calibrator Output		Range	2-Year Limits	
Amplitude	Frequency		Lower	Upper
90 mA	55 Hz	100 mA	89.9530 mA	90.0470 mA
90 mA	5 kHz	100 mA	89.9530 mA	90.0470 mA
90 mA	1 kHz	100 mA	89.9530 mA	90.0470 mA
90 mA	1 kHz	1 A	89.71 mA	90.29 mA
900 mA	1 kHz	1 A	898.90 mA	901.10 mA
300 mA	1 kHz	3 A	299.1 mA	300.9 mA
2.2 A	1 kHz	3 A	2.1972 A	2.2028 A

Frequency

Table 27. Frequency Limits

External Source Output Frequency	2-Year Limits	
	Lower	Upper
1 Hz	0.9999 Hz	1.0001 Hz
20 kHz	19.998 kHz	20.002 kHz
500 kHz	499.95 kHz	500.05 kHz

Appendix A: Calibration Options

The complete calibration process consists of verifying, adjusting, and reverifying a device. During verification, you compare the measured performance to an external standard of known measurement uncertainty to confirm that the product meets or exceeds specifications. Figure 7 shows the procedural flow for verification. During adjustment, you correct the measurement error of the device by adjusting the calibration constants and storing the new calibration constants in the EEPROM. Figure 8 shows the procedural flow for adjustment.

Normally, the calibration sequence is as follows:

1. Verify the NI 4071 using the 2-year accuracy limits (or the 90-day accuracy limits if it has been externally calibrated within that time).
2. Adjust the NI 4071.
3. Reverify the NI 4071 using the 24-hour accuracy limits (or the 2-year accuracy limits when the 24-hour limits are not specified).

Note You must compare the verification limits provided in this procedure with the most recent specifications. Refer to the latest NI 4071 specifications at `ni.com/manuals`.

Depending on your measurement and accuracy requirements, a complete calibration of the NI 4071 may not be necessary. A number of options are available that can shorten the calibration time. The following options are available:

- Complete calibration—Performing the entire calibration procedure from beginning to end; guarantees that the NI 4071 performs within the published specifications for all modes and ranges
- Complete calibration with exceptions:
  - Omitting AC voltage mode steps if you do not use the AC voltage modes or if the AC voltage accuracy is irrelevant
  - Omitting DC/AC current mode steps if you do not use the current modes or if the DC/AC current accuracy is irrelevant
  - Omitting both AC voltage and DC/AC current mode steps if you do not use those modes or if the accuracy of those measurements is irrelevant

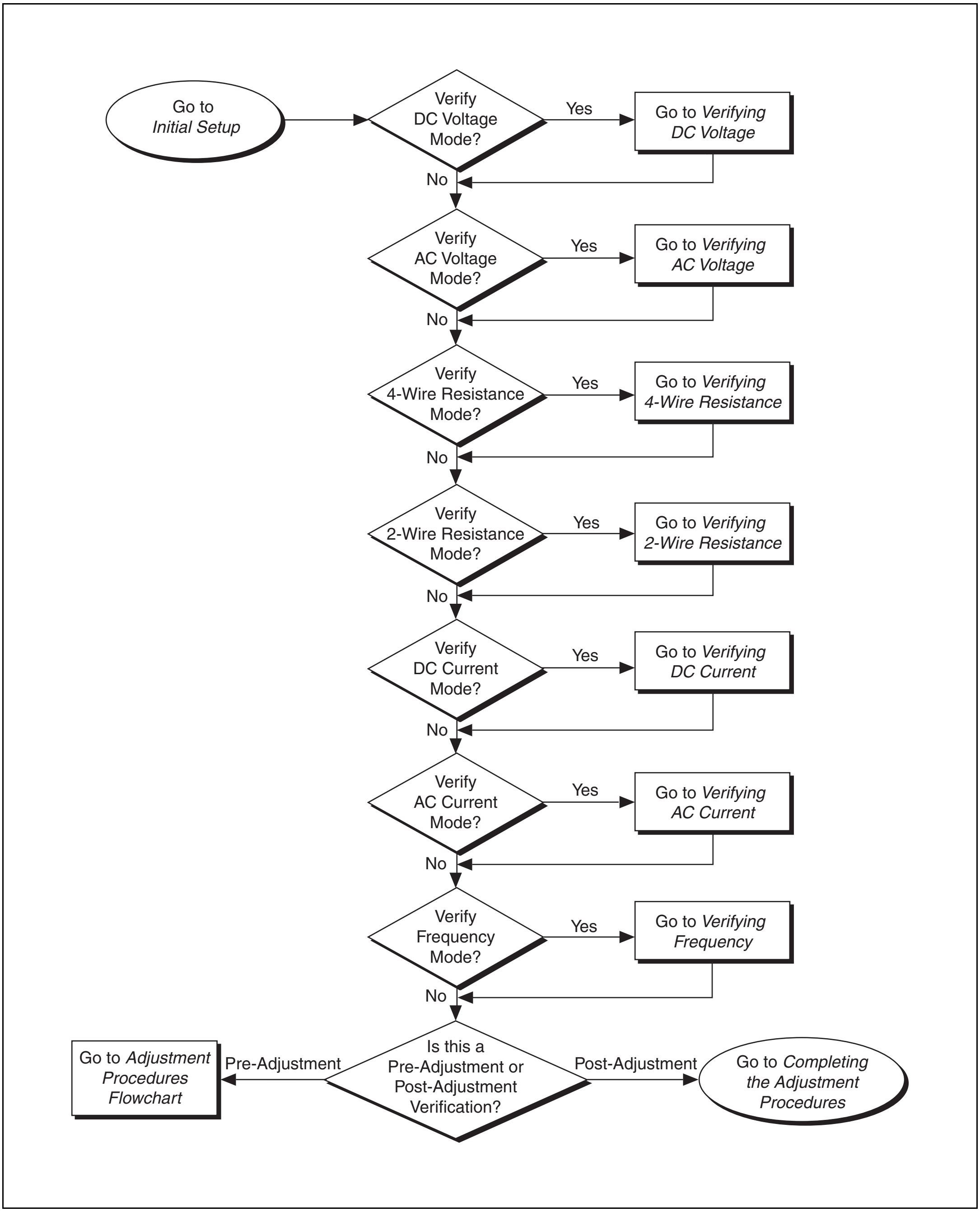


Figure 7. Verification Procedures Flowchart

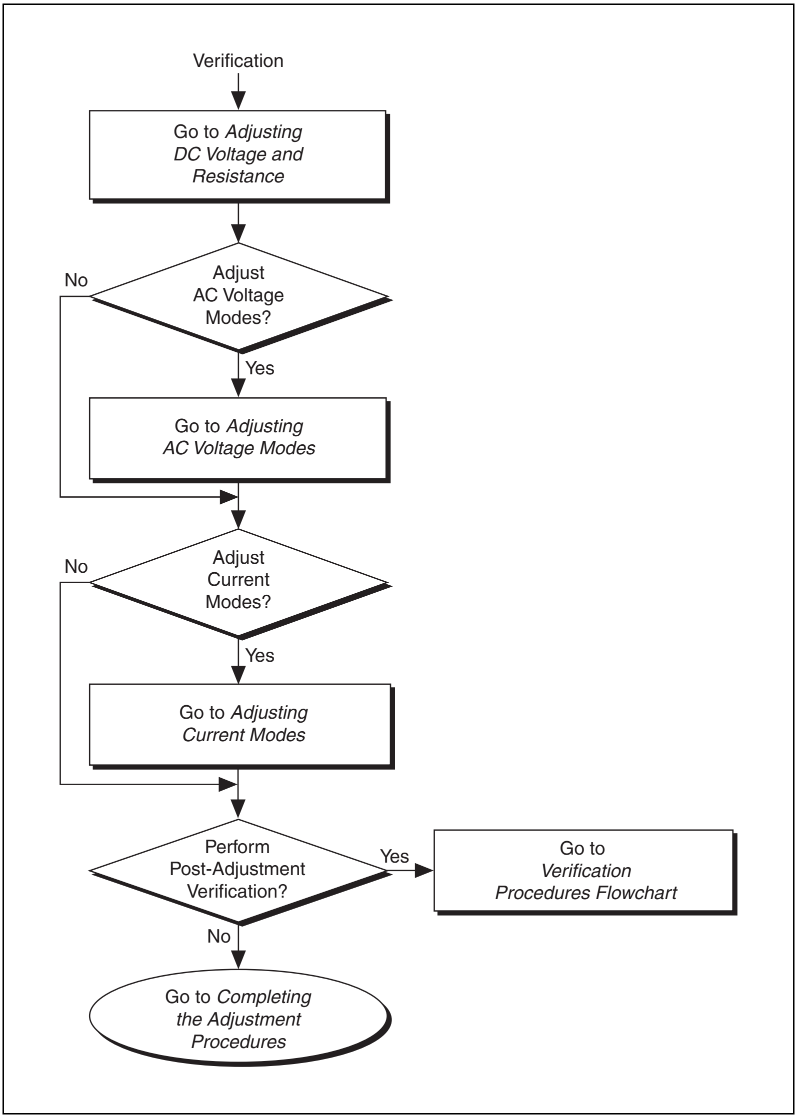


Figure 8. Adjustment Procedures Flowchart

Where to Go for Support

The National Instruments Web site is your complete resource for technical support. At `ni.com/support` you have access to everything from troubleshooting and application development self-help resources to email and phone assistance from NI Application Engineers.

A Declaration of Conformity (DoC) is our claim of compliance with the Council of the European Communities using the manufacturer's declaration of conformity. This system affords the user protection for electromagnetic compatibility (EMC) and product safety. You can obtain the DoC for your product by visiting `ni.com/certification`. If your product supports calibration, you can obtain the calibration certificate for your product at `ni.com/calibration`.

National Instruments corporate headquarters is located at 11500 North Mopac Expressway, Austin, Texas, 78759-3504. National Instruments also has offices located around the world to help address your support needs. For telephone support in the United States, create your service request at `ni.com/support` and follow the calling instructions or dial 512 795 8248. For telephone support outside the United States, visit the Worldwide Offices section of `ni.com/niglobal` to access the branch office Web sites, which provide up-to-date contact information, support phone numbers, email addresses, and current events.

371281G-01 Apr11