CALIBRATION PROCEDURE

NI PXI-4130

このドキュメントには、日本語ページも含まれています。

This document contains instructions for writing a manual calibration procedure for the NI PXI-4130 programmable, high-power source measure unit (SMU). Calibration is generally performed at National Instruments or a metrology lab with an external high-precision digital multimeter (DMM) and some additional test equipment. 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 **File»Page Setup»Options** directs you to pull down the **File** menu, select the **Page Setup** item, and select **Options** from the last dialog box.
	This icon denotes a note, which alerts you to important information.
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, 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

To calibrate the NI PXI-4130, you must install NI-DCPower version 1.2 or later on the calibration system. NI-DCPower includes all the functions and VIs necessary for calibration. You can download the latest version of NI-DCPower at `ni.com/idnet`.

NI-DCPower supports programming the calibration procedures in C and LabVIEW. For LabWindows™/CVI™, C calibration functions are installed in and are accessible from the NI-DCPower function panel, `niDCPower.fp`. For LabVIEW, calibration VIs are installed in the `niDCPower.llb` and accessible in LabVIEW from the Functions palette. Refer to Table 1 for file locations.

In this document, the LabVIEW VI is shown first, followed by the corresponding C function call. C function calls are valid for any compiler capable of calling a 32-bit DLL. Many of the functions use constants

defined in the `niDCPower.h` file. To use these constants in C, you must include `niDCPower.h` in the calibration program.

For more information about calibration VIs and functions, refer to the *NI DC Power Supplies and SMUs Help*, accessible at **Start» All Programs»National Instruments»NI-DCPower»Documentation» NI DC Power Supplies and SMUs Help**.

Table 1. Calibration File Locations (NI-DCPower 1.2 or Later)

File Name and Location	Description
`IVI\Bin\niDCPower_32.dll`	NI-DCPower driver containing the entire NI-DCPower API, including calibration functions.
`IVI\Lib\msc\niDCPower.lib`	NI-DCPower library for Microsoft C containing the entire NI-DCPower API, including calibration functions.
`<`*`LabVIEW`*`>\instr.lib\niDCPower Calibrate\niDCPower.llb`	LabVIEW VI library containing VIs for calling the NI-DCPower calibration API. You can access calibration functions from the NI-DCPower calibration section of the LabVIEW function palette.
`IVI\Drivers\niDCPower\niDCPower.fp`	CVI function panel file that includes calibration function prototypes and help on using NI-DCPower in the CVI environment.
`IVI\Include\niDCPower.h`	NI-DCPower header file, which you must include in any C program accessing calibration functions. This file includes the entire NI-DCPower API, including calibration functions.

Documentation Requirements

You might find the following documentation helpful as you write the calibration procedure:

- *NI PXI-4130 Specifications*
- *NI DC Power Supplies and SMUs Getting Started Guide*
- *NI DC Power Supplies and SMUs Help,* including LabVIEW VI and C function programming references

These documents are installed with NI-DCPower. You can also download the latest versions at `ni.com/manuals`.

Password

The default password for password-protected operations is `NI`.

Calibration Interval

The measurement accuracy requirements of your application determine how often you should calibrate your device. NI recommends that you perform a complete calibration for the NI PXI-4130 at least once a year. You can shorten this calibration interval based on the accuracy demands of your application. Refer to *Appendix A: Calibration Options* for more information.

Test Equipment

Table 2 lists the equipment required to calibrate the NI PXI-4130. If you do not have the recommended equipment, select a substitute calibration standard using the specifications listed in Table 2.

Table 2. Required Equipment Specifications for NI PXI-4130 Calibration

Required Equipment	Recommended Equipment	Specifications
Digital multimeter (DMM)	NI 4071	Voltage: better than ±50 ppm accuracy, better than 30 µV resolution; Current: better than ±0.04% accuracy, better than 1 µA resolution
External calibrator	Fluke 5700A/5720A	—
Auxiliary power supply	NI APS-4100	11 V to 15.5 V, 5 A
Twisted pair, shielded cabling wire	Belden 83319E 009100	18 AWG to 22 AWG

Test Conditions

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

- Keep cabling wire as short as possible. Long cables and wires act as antennae, picking up extra noise that can affect measurements. To further reduce noise, twist signal/common wires together.
- Verify that all connections, including front panel connections, are secure.
- Ensure that the PXI chassis fan speed is set to HI, that the fan filters are clean, and that the empty slots contain filler panels. For more information, refer to the *Maintain Forced-Air Cooling Note to Users* document available at `ni.com/manuals`.
- Keep relative humidity between 10% and 90%, noncondensing.
- Maintain an ambient temperature of 23 °C ±5 °C.
- Allow a warm-up time of at least 30 minutes after the NI-DCPower driver is loaded. Unless manually disabled, the NI-DCPower driver automatically loads with the operating system and enables the device. The warm-up time ensures that the measurement circuitry of the NI PXI-4130 is at a stable operating temperature.

Calibration Procedures

The calibration process includes the following procedures:

1. *Initial Setup*—Install the device and configure it in Measurement & Automation Explorer (MAX).
2. *Verification*—Verify the existing operation of the device. This procedure confirms whether the device is operating within its specified range prior to calibration.
3. *Adjustment*—Perform an external adjustment of the device that adjusts the calibration constants with respect to a known voltage source. The adjustment procedure automatically stores the calibration date on the EEPROM to allow traceability.
4. *Reverification*—Repeat the verification procedure to ensure that the device is operating within its specifications after adjustment.

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

Note The complete external calibration procedure consists of verifying the performance of the SMU, adjusting the calibration constants, and verifying performance again after the adjustments. In some cases, a complete calibration procedure may not be required. Refer to *Appendix A: Calibration Options* for more information.

Initial Setup

Refer to the *NI DC Power Supplies and SMUs Getting Started Guide* for information about how to install the software and hardware and how to configure the device in MAX.

Verification

This section describes the program you must write to verify the published specifications for the NI PXI-4130.

Verification consists of generating and measuring a series of outputs using the NI PXI-4130, verifying the accuracy with a DMM, and comparing the results to the calibration test limits. If the results fall within the test limits, the NI PXI-4130 meets its published specifications, and adjustment is optional. If the results fall outside of the test limits, you must adjust the NI PXI-4130.

Verification tests the following NI PXI-4130 specifications:

- Voltage programming accuracy
- Voltage measurement accuracy
- Current programming accuracy
- Current measurement accuracy

Tables 3 and 4 list configuration information for the calibration equipment required for verification.

Table 3. Calibration Equipment Configuration for Voltage Programming and Measurement Verification/Adjustment

NI PXI-4130		DMM*		
Channel(s)	**Range**	**Function**	**Range†**	**Input Impedance†**
0	6 V	DC Voltage	10 V	10 GΩ
1	6 V		10 V	10 GΩ
	20 V		100 V	10 MΩ

* Use the highest resolution available on the DMM. The DMM should have a minimum of 6.5 digit resolution.

† Assumes an NI 4071 DMM. For all other DMMs, use the range and input impedance closest to the values listed in this table.

Table 4. Calibration Equipment Configuration for Current Programming and Measurement Verification/Adjustment

NI PXI-4130		DMM*				
Channel	**Range**	**Function**	**Range**	**Input Impedance**	**Resolution in Digits**	**Calibrator Resistance**
0	1 A	DC Current	1 A	N/A	6.5	N/A
1	200 µA	DC Voltage	10 V	10 GΩ	7.5	10 kΩ
	2 mA	DC Voltage	10 V	10 GΩ	7.5	1 kΩ
	20 mA	DC Voltage	10 V	10 GΩ	7.5	100 Ω
	200 mA	DC Voltage	10 V	10 GΩ	7.5	10 Ω
	2 A	DC Current	3 A	N/A	6.5	N/A

* Use the highest resolution available on the DMM.

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

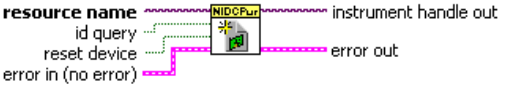
Verification of the NI PXI-4130 is complete only after you have successfully completed all tests in this section.

Note If verification fails post-adjustment, confirm that you have met the required *Test Conditions* before you return the NI PXI-4130 to NI for repair.

Verifying Voltage Programming Accuracy

Complete the following steps to verify the voltage programming accuracy of the NI PXI-4130. Complete this test for each iteration in Table 5.

1. Open a session and obtain a session handle using the niDCPower Initialize VI.

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_init` with the following parameters: **resourceName**: The device name assigned by MAX **idQuery**: `VI_FALSE` **resetDevice**: `VI_TRUE`

2. Connect the DMM to the channel *x* output terminals of the NI PXI-4130 as shown in Figure 1.

Note Channel *x* represents the channel under test. Replace the variable *x* in the program with the actual channel name.

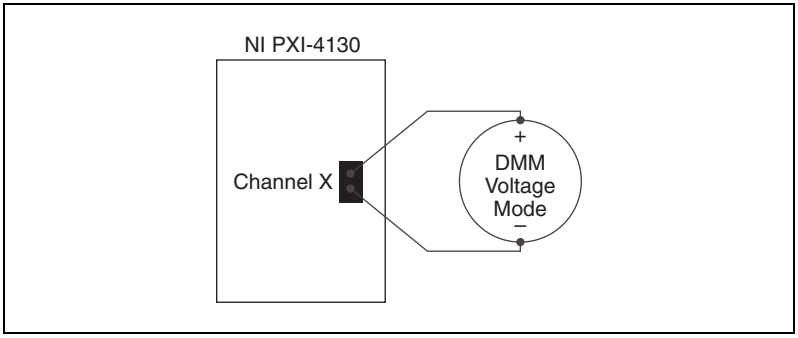


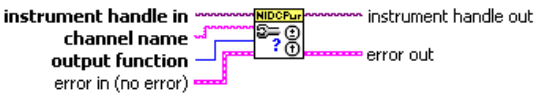
Figure 1. Voltage Accuracy Verification and Adjustment Setup for the NI PXI-4130

3. Configure the DMM for the mode and range listed for the corresponding channel and range in Table 3.

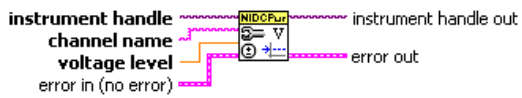
4. Place the NI PXI-4130 in delayed configuration mode using the niDCPower Abort VI.

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_Abort` with the following parameter: **vi**: The instrument handle from `niDCPower_init`

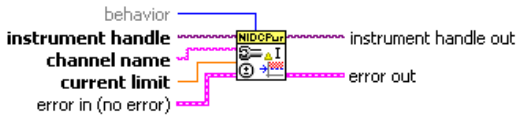
5. Set the output function to DC Voltage using the niDCPower Configure Output Function VI.

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_ConfigureOutputFunction` with the following parameters: **vi**: The instrument handle from `niDCPower_init` **channelName**: *x* **outputFunction**: `NIDCPOWER_VAL_DC_VOLTAGE`

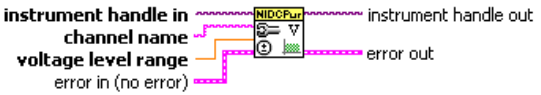
6. Configure the voltage level using the niDCPower Configure Voltage Level VI.

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_ConfigureVoltageLevel` with the following parameters: **vi**: The instrument handle from `niDCPower_init` **channelName**: *x* **level**: The *Output* value for the iteration of channel *x* in Table 5

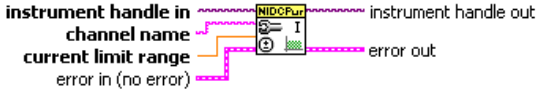
7. Configure the current limit using the niDCPower Configure Current Limit VI.

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_ConfigureCurrentLimit` with the following parameters: **vi**: The instrument handle from `niDCPower_init` **channelName**: *x* **behavior**: `NIDCPOWER_VAL_CURRENT_REGULATE` **limit**: `0.5`

8. Configure the voltage level range using the niDCPower Configure Voltage Level Range VI.

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_ConfigureVoltageLevelRange` with the following parameters: **vi**: The instrument handle from `niDCPower_init` **channelName**: *x* **voltageLevelRange**: The *Voltage Level Range* value for the iteration of channel *x* in Table 5

9. Configure the current limit range using the niDCPower Configure Current Limit Range VI.

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_ConfigureCurrentLimitRange` with the following parameters: **vi**: The instrument handle from `niDCPower_init` **channelName**: *x* **currentLimitRange**: 1

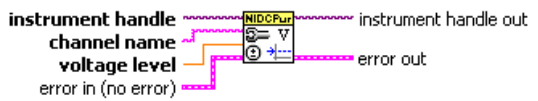
10. Enable the output using the niDCPower Configure Output Enabled VI.

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_ConfigureOutputEnabled` with the following parameters: **vi**: The instrument handle from `niDCPower_init` **channelName**: *x* **enabled**: `VI_TRUE`

11. Apply the configuration using the niDCPower Initiate VI.

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_Initiate` with the following parameter: **vi**: The instrument handle from `niDCPower_init`

12. Wait 3 s for the output of the NI PXI-4130 to settle.
13. Measure the output voltage with the DMM.
14. Record the measurement.
15. To calculate the output error, subtract the *Output* value for the iteration of channel *x* from the measurement you recorded in step 14.
16. Compare the output error to the *Test Limit* for the iteration of channel *x* in Table 5. If the output error is outside the test limit, you must adjust the NI PXI-4130.
17. Repeat steps 3 through 16 for each iteration of channel *x* in Table 5.
18. Set the voltage level to 0 using the niDCPower Configure Voltage Level VI.

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_ConfigureVoltageLevel` with the following parameters: **vi**: The instrument handle from `niDCPower_init` **channelName**: *x* **level**: `0`

19. Disable the output using the niDCPower Configure Output Enabled VI.

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_ConfigureOutputEnabled` with the following parameters: **vi**: The instrument handle from `niDCPower_init` **channelName**: *x* **enabled**: `VI_FALSE`

20. Disconnect the DMM.
21. Repeat steps 2 through 20 for all unverified channels in Table 5. When you have verified all iterations per channel, this part of the verification is complete.
22. End the session using the niDCPower Close VI.

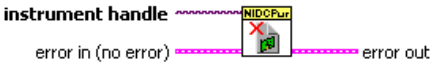
LabVIEW VI	C/C++ Function Call
	Call `niDCPower_close` with the following parameter: **vi**: The instrument handle from `niDCPower_init`

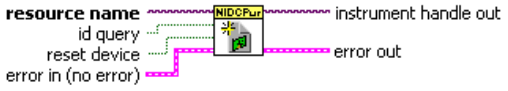
Table 5. NI PXI-4130 Output Parameters and Test Limits for Voltage Programming Accuracy Verification

Channel	Voltage Level Range (V)	Iteration	Output (V)	Test Limit (mV)
0	6	1	0	±4.00
		2	1.5	±4.75
		3	3	±5.50
		4	4.5	±6.25
		5	6	±7.00
1	6	1	0	±1.50
		2	1.5	±2.01
		3	3	±2.52
		4	4.5	±3.03
		5	6	±3.54
		6	–1.5	±2.01
		7	–3	±2.52
		8	–4.5	±3.03
		9	–6	±3.54
	20	1	0	±1.80
		2	5	±3.50
		3	10	±5.20
		4	15	±6.90
		5	20	±8.60
		6	–5	±3.50
		7	–10	±5.20
		8	–15	±6.90
		9	–20	±8.60

Verifying Voltage Measurement Accuracy

Complete the following steps to verify the voltage measurement accuracy of the NI PXI-4130. Complete this test for each iteration in Table 6.

1. Open a session and obtain a session handle using the niDCPower Initialize VI.

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_init` with the following parameters: **resourceName**: The device name assigned by MAX **idQuery**: `VI_FALSE` **resetDevice**: `VI_TRUE`

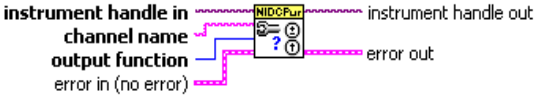
2. Connect the DMM to the channel *x* output terminals of the NI PXI-4130, as shown in Figure 1.

Note Channel *x* represents the channel under test. Replace the variable *x* in the program with the actual channel name.

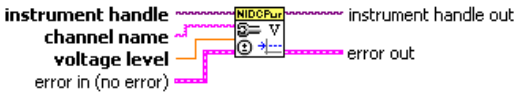
3. Configure the DMM for the mode and range listed for the corresponding channel and range in Table 3.
4. Place the NI PXI-4130 in delayed configuration mode using the niDCPower Abort VI.

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_Abort` with the following parameter: **vi**: The instrument handle from `niDCPower_init`

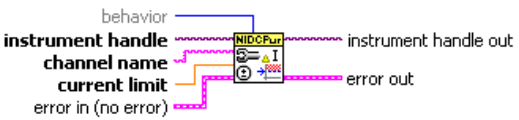
5. Set the output function to DC Voltage using the niDCPower Configure Output Function VI.

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_ConfigureOutputFunction` with the following parameters: **vi**: The instrument handle from `niDCPower_init` **channelName**: *x* **outputFunction**: `NIDCPOWER_VAL_DC_VOLTAGE`

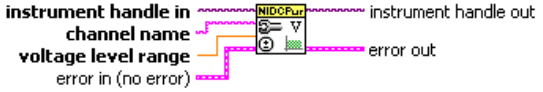
6. Configure the voltage level using the niDCPower Configure Voltage Level VI.

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_ConfigureVoltageLevel` with the following parameters: **vi**: The instrument handle from `niDCPower_init` **channelName**: *x* **level**: The *Output* value for the iteration of channel *x* in Table 6

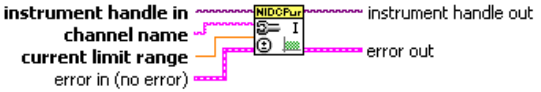
7. Configure the current limit using the niDCPower Configure Current Limit VI.

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_ConfigureCurrentLimit` with the following parameters: **vi**: The instrument handle from `niDCPower_init` **channelName**: *x* **behavior**: `NIDCPOWER_VAL_CURRENT_REGULATE` **limit**: `0.5`

8. Configure the voltage level range using the niDCPower Configure Voltage Level Range VI.

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_ConfigureVoltage LevelRange` with the following parameters: **vi**: The instrument handle from `niDCPower_init` **channelName**: *x* **voltageLevelRange**: The *Voltage Level Range* value for the iteration of channel *x* in Table 6

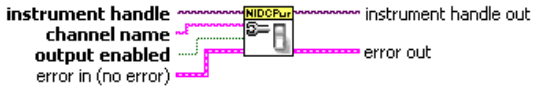
9. Configure the current limit range using the niDCPower Configure Current Limit Range VI.

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_ConfigureCurrent LimitRange` with the following parameters: **vi**: The instrument handle from `niDCPower_init` **channelName**: *x* **currentLimitRange**: 1

10. Specify the samples to average using the niDCPower property node.

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_SetAttribute ViInt32` with the following parameters: **vi**: The instrument handle from `niDCPower_init` **channelName**: *x* **attribute**: `NIDCPOWER_ATTR_SAMPLES_TO_AVERAGE` **value**: `300`

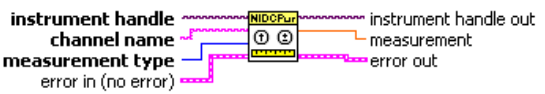
11. Enable the output using the niDCPower Configure Output Enabled VI.

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_ConfigureOutputEnabled` with the following parameters: **vi**: The instrument handle from `niDCPower_init` **channelName**: *x* **enabled**: `VI_TRUE`

12. Apply the configuration using the niDCPower Initiate VI.

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_Initiate` with the following parameter: **vi**: The instrument handle from `niDCPower_init`

13. Wait 3 s for the output of the NI PXI-4130 to settle.
14. Measure the output voltage with the DMM.
15. Record the measurement.
16. Measure the output voltage using the niDCPower Measure VI.

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_Measure` with the following parameters: **vi**: The instrument handle from `niDCPower_init` **channelName**: *x* **measurementType**: `NIDCPOWER_VAL_MEASURE_VOLTAGE`

17. Record the measurement.
18. To calculate the measurement error, subtract the measurement you recorded in step 15 from the measurement you recorded in step 17.
19. Calculate the upper and lower test limits using the offset and gain listed in the *Test Limit* column for the iteration of channel *x* in Table 6. Tolerances are provided instead of absolute limits because the DMM measures a unique value. Each limit is calculated by adding a

percentage of the DMM measurement and an offset voltage. Verify that the measurement error falls between the calculated limits. If the measurement error is outside the test limit, you must adjust the NI PXI-4130.

20. Repeat steps 4 through 19 for each iteration of channel *x* in Table 6.
21. Set the voltage level to 0 using the niDCPower Configure Voltage Level VI.

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_ConfigureVoltageLevel` with the following parameters: **vi**: The instrument handle from `niDCPower_init` **channelName**: *x* **level**: `0`

22. Disable the output using the niDCPower Configure Output Enabled VI.

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_ConfigureOutputEnabled` with the following parameters: **vi**: The instrument handle from `niDCPower_init` **channelName**: *x* **enabled**: `VI_FALSE`

23. Disconnect the DMM.
24. Repeat steps 2 through 23 for all unverified channels in Table 6. When you have verified all iterations per channel, this part of the verification is complete.
25. End the session using the niDCPower Close VI.

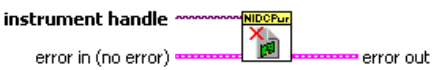
LabVIEW VI	C/C++ Function Call
	Call `niDCPower_close` with the following parameter: **vi**: The instrument handle from `niDCPower_init`

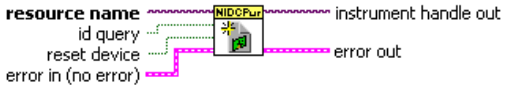
Table 6. NI PXI-4130 Output Parameters and Test Limits for Voltage Measurement Accuracy Verification

Channel	Voltage Level Range (V)	Iteration	Output (V)	Test Limit (V)
0	6	1	0	.05% + 4 mV
		2	1.5	
		3	3	
		4	4.5	
		5	6	
1	20	1	0	.03% + 1.5 mV
		2	5	
		3	10	
		4	15	
		5	20	
		6	–5	
		7	–10	
		8	–15	
		9	–20	

Verifying Current Programming Accuracy

Complete the following steps to verify the current programming accuracy of the NI PXI-4130. Complete this procedure for each channel iteration in Table 7. Please verify the output accuracy in the exact order listed in Table 7 to minimize any adverse effects caused by resistor heating.

1. Open a session and obtain a session handle using the niDCPower Initialize VI.

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_init` with the following parameters: **resourceName**: The device name assigned by MAX **idQuery**: `VI_FALSE` **resetDevice**: `VI_TRUE`

2. Connect the NI PXI-4130 channel x to the DMM or to the Fluke 5700A/5720A calibrator, as illustrated in Figure 2 or Figure 3.

   The setup in Figure 2 is used for the 200 µA, 2 mA, 20 mA, and 200 mA current ranges of channel 1. The setup in Figure 3 is used for the 1 A current range of channel 0 and the 2 A current range of channel 1.

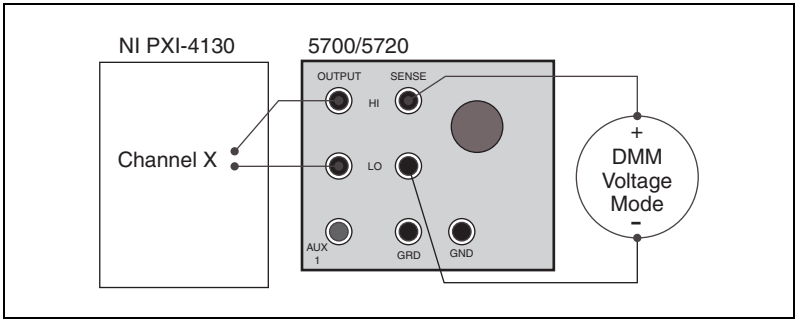


Figure 2. Current Programming Accuracy Verification Setup for the 200 µA, 2 mA, 20 mA, and 200 mA Current Ranges of Channel 1

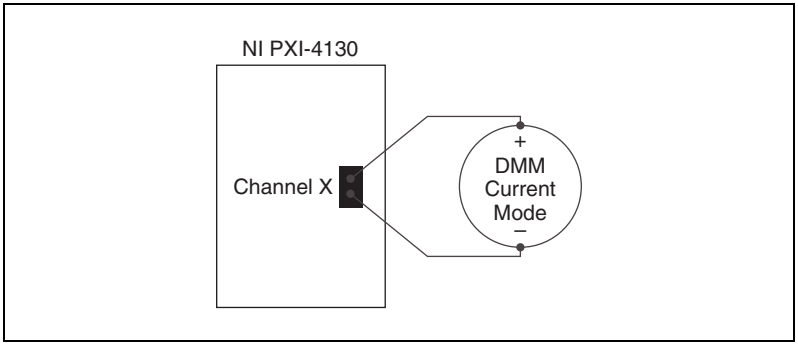


Figure 3. Current Programming Accuracy Verification Setup for the 1 A Current Range of Channel 0 and the 2 A Range of Channel 1

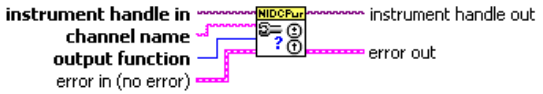
3. Configure the DMM to the mode and range listed for the corresponding channel and range in Table 4.
4. When applicable, configure the Fluke 5700/5720A calibrator to the *Resistance* value for the corresponding channel and iteration in Table 4. Enable external sense (4-wire mode) on the calibrator. Record the actual resistance value displayed by the calibrator.

Note Channel *x* represents the channel under test. Replace the variable *x* in the program with the actual channel name.

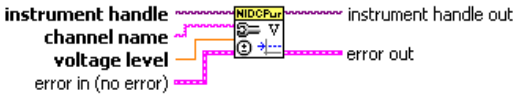
5. Place the NI PXI-4130 in delayed configuration mode using the niDCPower Abort VI.

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_Abort` with the following parameter: **vi**: The instrument handle from `niDCPower_init`

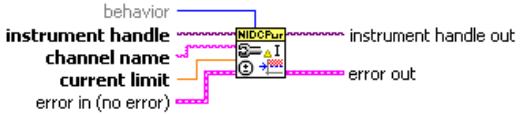
6. Set the output function to DC Voltage using the niDCPower Configure Output Function VI.

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_ConfigureOutputFunction` with the following parameters: **vi**: The instrument handle from `niDCPower_init` **channelName**: *x* **outputFunction**: `NIDCPOWER_VAL_DC_VOLTAGE`

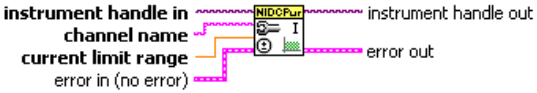
7. Configure the voltage level using the niDCPower Configure Voltage Level VI.

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_ConfigureVoltageLevel` with the following parameters: **vi**: The instrument handle from `niDCPower_init` **channelName**: *x* **voltageLevel**: The *Voltage Level* value for the iteration of channel *x* in Table 7

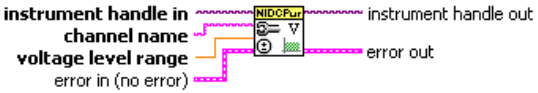
8. Configure the current limit for the corresponding channel and iteration in Table 7 using the niDCPower Configure Current Limit VI.

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_ConfigureCurrentLimit` with the following parameters: **vi**: The instrument handle from `niDCPower_init` **channelName**: *x* **currentLimit**: The *Output* value for the iteration of channel *x* in Table 7

9. Configure the current limit range using the niDCPower Configure Current Limit Range VI for the corresponding channel and iteration in Table 7.

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_ConfigureCurrentLimitRange` with the following parameters: **vi**: The instrument handle from `niDCPower_init` **channelName**: *x* **currentLimitRange**: The *Current Limit Range* value for the iteration of channel *x* in Table 7

10. Configure the voltage level range for the corresponding channel and iteration in Table 7 using the niDCPower Configure Voltage Level Range VI.

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_ConfigureVoltageLevelRange` with the following parameters: **vi**: The instrument handle from `niDCPower_init` **channelName**: *x* **voltageLevelRange**: The *Voltage Level Range* value for the iteration of channel *x* in Table 7

11. Enable the output using the niDCPower Configure Output Enabled VI.

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_ConfigureOutputEnabled` with the following parameters: **vi**: The instrument handle from `niDCPower_init` **channelName**: *x* **enabled**: `VI_TRUE`

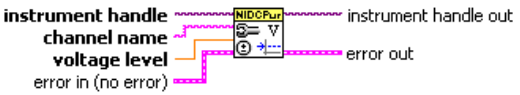
12. Apply the configuration using the niDCPower Initiate VI.

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_Initiate` with the following parameter: **vi**: The instrument handle from `niDCPower_init`

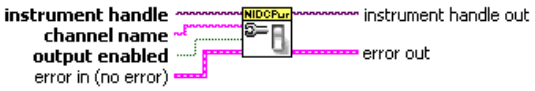
13. Wait 3 s for the output of the NI PXI-4130 to settle.
14. Measure the output voltage or current using the DMM.
15. Record the measurement.
16. For the 200 µA, 2 mA, 20 mA, and 200 mA current ranges of channel 1, divide the voltage measurement you recorded in step 15 by the resistance measurement you recorded in step 4 to calculate the output current. For the 1 A current range of channel 0 and the 2 A current range of channel 1, the output current is measured directly by the DMM. Subtract the *Output* value for the iteration of channel *x* in Table 7 from the output current calculated above to obtain the output error.
17. Compare the output error to the *Test Limit* for the iteration of channel *x* in Table 7. If the output error is outside the test limit, you must adjust the NI PXI-4130.
18. Repeat steps 2 through 17 for all iterations of channel *x* in Table 7 per current range.

Note For channel 1, each current limit range has several iterations with a positive voltage level and another set of iterations with a negative voltage level.

19. Set the voltage level to 0 V using the niDCPower Configure Voltage Level VI.

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_ConfigureVoltageLevel` with the following parameters: **vi**: The instrument handle from `niDCPower_init` **channelName**: *x* **level**: `0`

20. Disable the output using the niDCPower Configure Output Enabled VI.

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_ConfigureOutputEnabled` with the following parameters: **vi**: The instrument handle from `niDCPower_init` **channelName**: *x* **enabled**: `VI_FALSE`

21. Disconnect the DMM and the calibrator.
22. Repeat steps 3 through 21 for the all unverified channels in Table 7. When you have verified all iterations per channel and range, this part of the verification is complete.
23. End the session using the niDCPower Close VI.

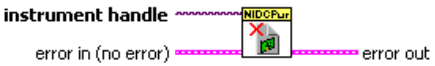
LabVIEW VI	C/C++ Function Call
	Call `niDCPower_close` with the following parameter: **vi**: The instrument handle from `niDCPower_init`

Table 7. NI PXI-4130 Output Parameters and Test Limits for Current Programming Accuracy Verification

Channel(s)	Current Limit Range	Voltage Level Range	Voltage Level	Iteration	Output (A)	Test Limit
0	1 A	6 V	6 V	1	50 mA	±4.08 mA
				2	350 mA	±4.53 mA
				3	700 mA	±5.05 mA
1	200 µA	20 V	20 V	1	4.00 µA	±0.101 µA
				2	50.0 µA	±0.115 µA
				3	0.10 mA	±0.130 µA
				4	0.15 mA	±0.145 µA
				5	0.20 mA	±0.160 µA
			–20 V	6	4.00 µA	±0.101 µA
				7	0.50 mA	±0.115 µA
				8	0.10 mA	±0.130 µA
				9	0.15 mA	±0.145 µA
				10	0.20 mA	±0.160 µA
	2 mA		20 V	1	40.0 µA	±1.01 µA
				2	0.50 mA	±1.15 µA
				3	1.00 mA	±1.30 µA
				4	1.50 mA	±1.45 µA
				5	2.00 mA	±1.60 µA
			–20 V	6	40.0 µA	±1.01 µA
				7	0.50 mA	±1.15 µA
				8	1.00 mA	±1.30 µA
				9	1.50 mA	±1.45 µA
				10	2.00 mA	±1.60 µA
	20 mA		20 V	1	0.40 mA	±10.1 µA
				2	5.00 mA	±11.5 µA
				3	10.0 mA	±13.0 µA
				4	15.0 mA	±14.5 µA

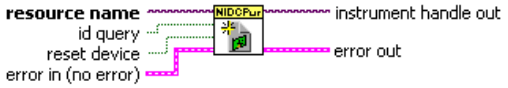
Table 7. NI PXI-4130 Output Parameters and Test Limits for Current Programming Accuracy Verification (Continued)

Channel(s)	Current Limit Range	Voltage Level Range	Voltage Level	Iteration	Output (A)	Test Limit
1	20 mA	20 V	20 V	5	20.0 mA	±16.0 µA
			–20 V	6	0.40 mA	±10.1 µA
				7	5.00 mA	±11.5 µA
				8	10.0 mA	±13.0 µA
				9	15.0 mA	±14.5 µA
				10	20.0 mA	±16.0 µA
	200 mA		20 V	1	4.00 mA	±0.101 mA
				2	50.0 mA	±0.115 mA
				3	100 mA	±0.130 mA
				4	150 mA	±0.145 mA
				5	200 mA	±0.160 mA
			–20 V	6	4.00 mA	±0.101 mA
				7	50.0 mA	±0.115 mA
				8	100 mA	±0.130 mA
				9	150 mA	±0.145 mA
				10	200 mA	±0.160 mA
	2 A		20 V	1	40.0 mA	±1.05 mA
				2	0.5 A	±1.60 mA
				3	1.0 A	±2.70 mA
				4	1.5 A	±4.68 mA
				5	2.0 A	±8.40 mA
			–20 V	6	40.0 mA	±1.05 mA
				7	0.5 A	±1.60 mA
				8	1.0 A	±2.70 mA
				9	1.5 A	±4.68 mA
				10	2.0 A	±8.40 mA

Verifying Current Measurement Accuracy

Complete the following steps to verify the current measurement accuracy of the NI PXI-4130. Complete this procedure for each channel iteration per supported range in Table 8.

1. Open a session and obtain a session handle using the niDCPower Initialize VI.

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_init` with the following parameters: **resourceName**: The device name assigned by MAX **idQuery**: `VI_FALSE` **resetDevice**: `VI_TRUE`

2. If the *Voltage Level* in Table 8 for this iteration of channel *x* is 0, skip to step 5. Do *not* connect the DMM or the Fluke 5700A/5720A calibrator to the channel *x* output terminals of the NI PXI-4130.

   For *Output* values other than 0, connect the NI PXI-4130 channel *x* to the DMM or to the Fluke 5700A/5720A calibrator, as illustrated in Figure 2 or Figure 3.

   The setup in Figure 2 is used for the 200 µA, 2 mA, 20 mA, and 200 mA current ranges of channel 1. The setup in Figure 3 is used for the 1 A current range of channel 0 and the 2 A current range of channel 1.

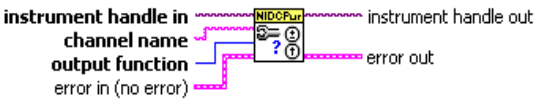
Note Channel *x* represents the channel under test. Replace the variable *x* in the program with the actual channel name.

3. Configure the DMM to the mode and range listed for the corresponding channel and range in Table 4.
4. When applicable, configure the Fluke 5700/5720A calibrator to the *Resistance* value for the corresponding channel and iteration in Table 4. Enable external sense (4-wire mode) on the calibrator. Record the actual resistance value displayed by the calibrator.

5. Place the NI PXI-4130 in delayed configuration mode using the niDCPower Abort VI.

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_Abort` with the following parameter: **vi**: The instrument handle from `niDCPower_init`

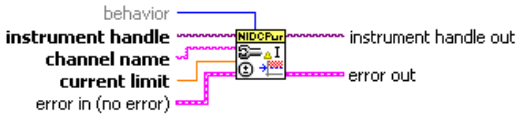
6. Set the output function to voltage control using the niDCPower Configure Output Function VI.

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_ConfigureOutput Function` with the following parameters: **vi**: The instrument handle from `niDCPower_init` **channelName**: *x* **outputFunction**: `NIDCPOWER_VAL_DC_VOLTAGE`

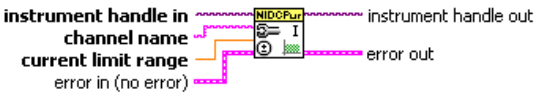
7. Configure the voltage level using the niDCPower Configure Voltage Level VI.

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_ConfigureVoltageLevel` with the following parameters: **vi**: The instrument handle from `niDCPower_init` **channelName**: *x* **level**: The *Voltage Level* value for the iteration of channel *x* in Table 8

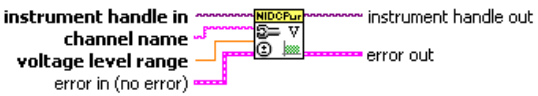
8. Configure the current limit for the corresponding channel and iteration in Table 8 using the niDCPower Configure Current Limit VI.

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_ConfigureCurrentLimit` with the following parameters: **vi**: The instrument handle from `niDCPower_init` **channelName**: *x* **currentLimit**: The *Output* value for the iteration of channel *x* in Table 8

9. Configure the current limit range using the niDCPower Configure Current Limit Range VI for the corresponding channel and iteration in Table 8.

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_ConfigureCurrentLimitRange` with the following parameters: **vi**: The instrument handle from `niDCPower_init` **channelName**: *x* **currentLimitRange**: The *Current Limit Range* value for the iteration of channel *x* in Table 8

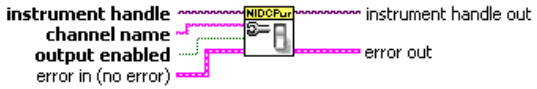
10. Configure the voltage level range for the corresponding channel and iteration in Table 8 using the niDCPower Configure Voltage Level Range VI.

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_ConfigureVoltageLevelRange` with the following parameters: **vi**: The instrument handle from `niDCPower_init` **channelName**: *x* **voltageLevelRange**: The *Voltage Level Range* value for the iteration of channel *x* in Table 8

11. Specify the samples to average using the niDCPower property node.

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_SetAttributeViInt32` with the following parameters: **vi**: The instrument handle from `niDCPower_init` **channelName**: *x* **attribute**: `NIDCPOWER_ATTR_SAMPLES_TO_AVERAGE` **value**: `300`

12. Enable the output using the niDCPower Configure Output Enabled VI.

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_ConfigureOutputEnabled` with the following parameters: **vi**: The instrument handle from `niDCPower_init` **channelName**: *x* **enabled**: `VI_TRUE`

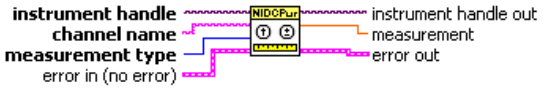
13. Apply the configuration using the niDCPower Initiate VI.

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_Initiate` with the following parameter: **vi**: The instrument handle from `niDCPower_init`

14. Wait 3 s for the output of the NI PXI-4130 to settle.
15. If the *Output* value for this iteration of channel *x* is 0, check that the DMM or calibrator is disconnected from the NI PXI-4130 and skip to step 16.

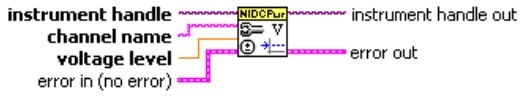
    For all other *Output* values, measure the output voltage or current using the DMM.

16. Record the measurement. If the *Voltage Level* value for this iteration of channel *x* is 0, do not take a measurement; record "0" in place of the measurement.
17. For the 200 µA, 2 mA, 20 mA, and 200 mA current ranges of channel 1, divide the voltage measurement you recorded in step 16 by the resistance measurement you recorded in step 4 to calculate the output current. For the 1 A current range of channel 0 and the 2 A current range of channel 1, the output current is measured directly by the DMM in the previous step.
18. Measure the output current using the niDCPower Measure VI.

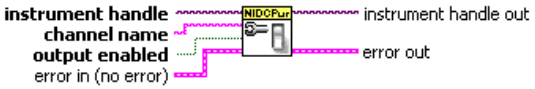
LabVIEW VI	C/C++ Function Call
	Call `niDCPower_Measure` with the following parameters: **vi**: The instrument handle from `niDCPower_init` **channelName**: *x* **measurementType**: `NIDCPOWER_VAL_MEASURE_CURRENT`

19. Record the measurement.
20. To calculate the measurement error, subtract the measurement you calculated in step 17 from the measurement you recorded in step 19.
21. Calculate the upper and lower test limits using the offset and gain listed in the *Test Limits* column for the iteration of channel *x* in Table 8. Tolerances are provided instead of absolute limits because the DMM measures a unique value. Each limit is calculated by adding a percentage of the actual output current and an offset current. Use the measurement from step 17 to calculate the test limits. Verify that the measurement error falls between the calculated limits. If the measurement error is outside the test limits, you must adjust the NI PXI-4130.
22. Repeat steps 2 through 21 for all iterations of channel *x* per supported current range in Table 8.

23. Set the voltage level to 0 V using the niDCPower Configure Voltage Level VI.

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_ConfigureVoltageLevel` with the following parameters: **vi**: The instrument handle from `niDCPower_init` **channelName**: *x* **level**: `0`

24. Disable the output using the niDCPower Configure Output Enabled VI.

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_ConfigureOutputEnabled` with the following parameters: **vi**: The instrument handle from `niDCPower_init` **channelName**: *x* **enabled**: `VI_FALSE`

25. Disconnect the DMM and the calibrator.
26. Repeat steps 3 through 25 for the all unverified channels in Table 8. When you have verified all iterations per channel and range, this part of the verification is complete.
27. End the session using the niDCPower Close VI.

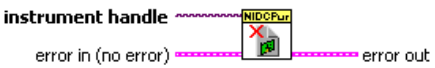
LabVIEW VI	C/C++ Function Call
	Call `niDCPower_close` with the following parameter: **vi**: The instrument handle from `niDCPower_init`

Table 8. NI PXI-4130 Output Parameters and Test Limits for Current Measurement Accuracy Verification

Channel(s)	Current Limit Range	Voltage Level Range (V)	Voltage Level (V)	Iteration	Output	Test Limit ± (% of reading + offset)
0	1 A	6	0	1	350 mA	.15% + 4 mA[1]
			6	2	350 mA	
				3	700 mA	
1	200 µA	20	0	1	50.0 µA	.03% + 0.02 µA
			20	2	50.0 µA	
				3	0.10 mA	
				4	0.150 mA	
				5	0.20 mA	
			–20	6	50.0 µA	
				7	0.10 mA	
				8	0.15 mA	
				9	0.2.0 mA	
	2 mA		0	1	0.50 mA	.03% + 0.2 µA
			20	2	0.50 mA	
				3	1.00 mA	
				4	1.50 mA	
				5	2.00 mA	
			–20	6	0.50 mA	
				7	1.00 mA	
				8	1.50 mA	
				9	2.00 mA	
	20 mA		0	1	5.00 mA	.03% + 2.0 µA
			20	2	5.00 mA	
				3	10.0 mA	
				4	15.0 mA	
				5	20.0 mA	
			–20	6	5.00 mA	

Table 8. NI PXI-4130 Output Parameters and Test Limits for Current Measurement Accuracy Verification (Continued)

Channel(s)	Current Limit Range	Voltage Level Range (V)	Voltage Level (V)	Iteration	Output	Test Limit ± (% of reading + offset)
1	20 mA	20 V	–20	7	10.0 mA	.03% + 2.0 µA
				8	15.0 mA	
				9	20.0 mA	
	200 mA		0	1	50.0 mA	.03% + 40 µA
			20	2	50.0 mA	
				3	100 mA	
				4	150 mA	
				5	200 mA	
			–20	6	50.0 mA	
				7	100 mA	
				8	150 mA	
				9	200 mA	
	2 A		0	1	0.5 A	.12% + 200 µA[1]
			20	2	0.5 A	
				3	1.0 A	
				4	1.5 A	
				5	2.0 A	
			–20 V	6	0.5 A	
				7	1.0 A	
				8	1.5 A	
				9	2.0 A	

[1] For currents ≥ 500 mA, refer to the additional derating information in Figure 4, *Accuracy Derating versus Load Current*.

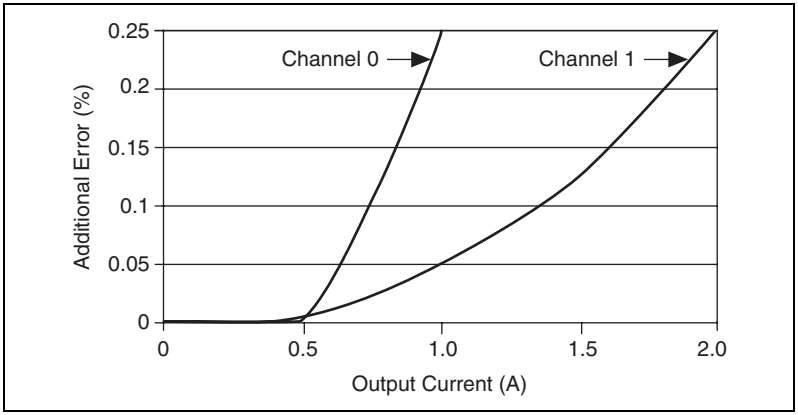


Figure 4. Accuracy Derating versus Load Current

Adjustment

Adjustment improves the accuracy of the NI PXI-4130 and updates the calibration date and temperature in the EEPROM. Perform an adjustment once a year or when the accuracy of NI PXI-4130 is outside the calibration test limits.

Adjustment corrects the following NI PXI-4130 specifications:

- Voltage programming accuracy
- Voltage measurement accuracy
- Current programming accuracy
- Current measurement accuracy

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

Note If the NI PXI-4130 has passed initial verification and is within all test limits, NI recommends, but does not require, an adjustment to guarantee the published specifications for the next year. If you choose to skip adjustment, run the niDCPower Initialize External Calibration VI and end with the niDCPower Close External Calibration VI with **action** set to **Commit** to update the calibration date and onboard calibration temperature without making any adjustments to the device.

After adjustment, repeat the *Verification* section to verify that the adjustment was successful.

Adjusting Voltage Programming Accuracy

Complete the following steps to adjust the voltage programming accuracy of the NI PXI-4130. Complete this test for each iteration in Table 9.

1. Open a session and obtain a session handle using the niDCPower Initialize External Calibration VI.

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_InitExtCal` with the following parameters: **resourceName**: The device name assigned by MAX **password**: `NI`

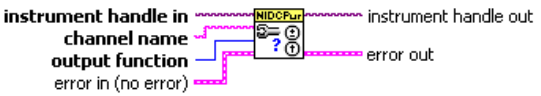
2. Connect the DMM to the channel *x* output terminals of the NI PXI-4130, as shown in Figure 1.

Note Channel *x* represents the channel under test. Replace the variable *x* in the program with the actual channel name.

3. Configure the DMM for the range and mode listed for the corresponding channel and range in Table 3.
4. Place the NI PXI-4130 in delayed configuration mode using the niDCPower Abort VI.

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_Abort` with the following parameter: **vi**: The instrument handle from `niDCPower_InitExtCal`

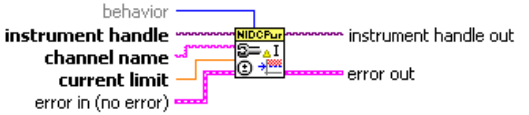
5. Configure the output function using the niDCPower Configure Output Function VI.

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_ConfigureOutput Function` with the following parameters: **vi**: The instrument handle from `niDCPower_InitExtCal` **channelName**: *x* **outputFunction**: `NIDCPOWER_VAL_DC_VOLTAGE`

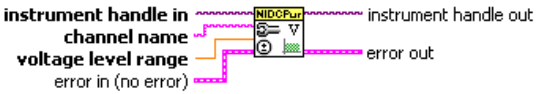
6. Configure the voltage level using the niDCPower Configure Voltage Level VI.

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_ConfigureVoltageLevel` with the following parameters: **vi**: The instrument handle from `niDCPower_InitExtCal` **channelName**: *x* **level**: The *Output* value for the iteration of channel *x* in Table 9

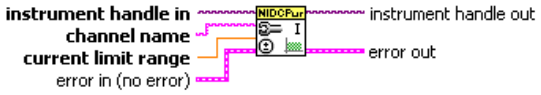
7. Configure the current limit using the niDCPower Configure Current Limit VI.

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_ConfigureCurrentLimit` with the following parameters: **vi**: The instrument handle from `niDCPower_InitExtCal` **channelName**: *x* **behavior**: `NIDCPOWER_VAL_CURRENT_REGULATE` **limit**: `0.5`

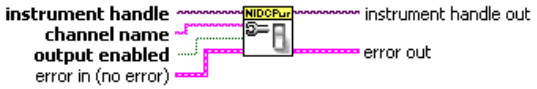
8. Configure the voltage level range using the niDCPower Configure Voltage Level Range VI.

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_ConfigureVoltageLevelRange` with the following parameters: **vi**: The instrument handle from `niDCPower_InitExtCal` **channelName**: *x* **voltageLevelRange**: The *Voltage Level Range* value for the iteration of channel *x* in Table 9

9. Configure the current limit range using the niDCPower Configure Current Limit Range VI.

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_ConfigureCurrentLimitRange` with the following parameters: **vi**: The instrument handle from `niDCPower_InitExtCal` **channelName**: *x* **currentLimitRange**: 1

10. Enable the output using the niDCPower Configure Output Enabled VI.

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_ConfigureOutputEnabled` with the following parameters: **vi**: The instrument handle from `niDCPower_InitExtCal` **channelName**: *x* **enabled**: `VI_TRUE`

11. Apply the configuration using the niDCPower Initiate VI.

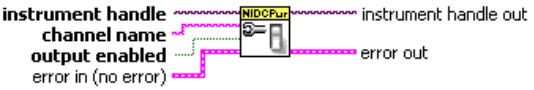
LabVIEW VI	C/C++ Function Call
	Call `niDCPower_Initiate` with the following parameter: **vi**: The instrument handle from `niDCPower_InitExtCal`

12. Wait 3 s for the output of the NI PXI-4130 to settle.
13. Measure the output voltage with the DMM.
14. Record the measurement.
15. Repeat steps 3 through 14 for each iteration of channel *x* in Table 9.

16. Set the voltage level to 0 V using the niDCPower Configure Voltage Level VI.

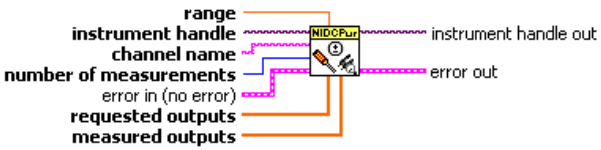
LabVIEW VI	C/C++ Function Call
	Call `niDCPower_ConfigureVoltageLevel` with the following parameters: **vi**: The instrument handle from `niDCPower_InitExtCal` **channelName**: *x* **level**: 0

17. Disable the output using the niDCPower Configure Output Enabled VI.

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_ConfigureOutput Enabled` with the following parameters: **vi**: The instrument handle from `niDCPower_InitExtCal` **channelName**: *x* **enabled**: `VI_FALSE`

18. Adjust the voltage level using the niDCPower Cal Adjust Voltage Level VI for every voltage level range of channel *x*.

Note When calibrating channel 1, each voltage level range requires a unique call to the niDCPower Cal Adjust Voltage Level VI. Positive and negative output values within the same range must be calibrated using unique calls as well.

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_CalAdjustVoltageLevel` with the following parameter: **vi**: The instrument handle from `niDCPower_InitExtCal` **channelName**: *x* **range**: The voltage level range of channel *x* **numberOfMeasurements**: An integer value of the total number of measurements. This value should match the number of elements in the **requestedOutputs** and **measuredOutputs** arrays **requestedOutputs**: An array composed of the *Output* values for each iteration of channel *x* in Table 9 for the range to be calibrated **measuredOutputs**: An array composed of the measurement values you recorded in step 14 for each iteration of channel *x* in Table 9 for the range to be calibrated

19. Disconnect the DMM.
20. Repeat steps 2 through 19 for all unadjusted channels in Table 9. When you have adjusted all voltage measurements on all channels, this part of the adjustment is complete.

21. End the session using the niDCPower Close External Calibration VI.

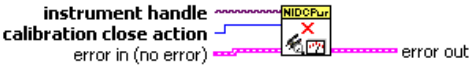
LabVIEW VI	C/C++ Function Call
	Call `niDCPower_CloseExtCal` with the following parameters: **vi**: The instrument handle from `niDCPower_InitExtCal` **action**: The instrument handle from `niDCPower_VAL_COMMIT`

Table 9. NI PXI-4130 Output Parameters for Voltage Programming Accuracy Adjustment

Channel	Iteration	Voltage Level Range (V)	Output
0	1	6	0 V
	2		3 V
	3		6 V
1	1	6	1 mV
	2		3 V
	3		6 V
	1		–0.1 mV
	2		–3 V
	3		–6 V
	1	20	1 mV
	2		10 V
	3		20 V
	1		–1 mV
	2		–10 V
	3		–20 V

Adjusting Voltage Measurement Accuracy

Complete the following steps to adjust the voltage measurement accuracy of the NI PXI-4130. Complete this test for each iteration in Table 10.

1. Open a session and obtain a session handle using the niDCPower Initialize External Calibration VI.

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_InitExtCal` with the following parameters: **resourceName**: The device name assigned by MAX **password**: `NI`

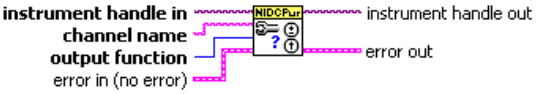
2. Connect the DMM to the channel *x* output terminals of the NI PXI-4130, as shown in Figure 1.

Note Channel *x* represents the channel under test. Replace the variable *x* in the program with the actual channel name.

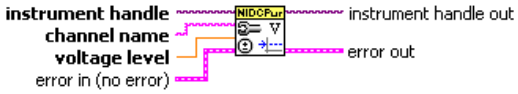
3. Configure the DMM for the mode and range listed for the corresponding channel and range in Table 3.
4. Place the NI PXI-4130 in delayed configuration mode using the niDCPower Abort VI.

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_Abort` with the following parameter: **vi**: The instrument handle from `niDCPower_InitExtCal`

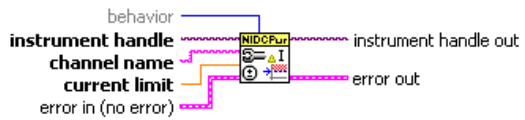
5. Set the output function to DC Voltage using the niDCPower Configure Output Function VI.

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_ConfigureOutput Function` with the following parameters: **vi**: The instrument handle from `niDCPower_InitExtCal` **channelName**: *x* **outputFunction**: `NIDCPOWER_VAL_DC_VOLTAGE`

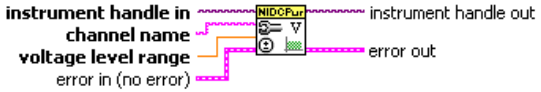
6. Configure the voltage level using the niDCPower Configure Voltage Level VI.

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_ConfigureVoltageLevel` with the following parameters: **vi**: The instrument handle from `niDCPower_InitExtCal` **channelName**: *x* **level**: The *Output* value for the iteration of channel *x* in Table 10

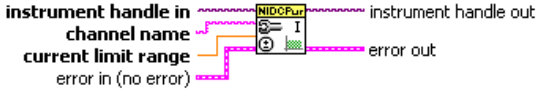
7. Configure the current limit using the niDCPower Configure Current Limit VI.

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_ConfigureCurrentLimit` with the following parameters: **vi**: The instrument handle from `niDCPower_InitExtCal` **channelName**: *x* **behavior**: `NIDCPOWER_VAL_CURRENT_REGULATE` **limit**: `0.5`

8. Configure the voltage level range using the niDCPower Configure Voltage Level Range VI.

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_ConfigureVoltageLevelRange` with the following parameters: **vi**: The instrument handle from `niDCPower_InitExtCal` **channelName**: *x* **voltageLevelRange**: The *Voltage Level Range* value for the iteration of channel *x* in Table 10

9. Configure the current limit range using the niDCPower Configure Current Limit Range VI.

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_ConfigureCurrentLimitRange` with the following parameters: **vi**: The instrument handle from `niDCPower_InitExtCal` **channelName**: *x* **currentLimitRange**: 1

10. Specify the samples to average using the niDCPower property node.

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_SetAttributeViInt32` with the following parameters: **vi**: The instrument handle from `niDCPower_InitExtCal` **channelName**: *x* **attribute**: `NIDCPOWER_ATTR_SAMPLES_TO_AVERAGE` **value**: `300`

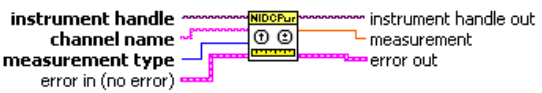
11. Enable the output using the niDCPower Configure Output Enabled VI.

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_ConfigureOutputEnabled` with the following parameters: **vi**: The instrument handle from `niDCPower_InitExtCal` **channelName**: *x* **enabled**: `VI_TRUE`

12. Apply the configuration using the niDCPower Initiate VI.

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_Initiate` with the following parameter: **vi**: The instrument handle from `niDCPower_InitExtCal`

13. Wait 3 s for the output of the NI PXI-4130 to settle.
14. Measure the output voltage with the DMM.
15. Record the measurement.
16. Measure the output voltage using the niDCPower Measure VI.

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_Measure` with the following parameters: **vi**: The instrument handle from `niDCPower_InitExtCal` **channelName**: *x* **measurementType**: `NIDCPOWER_VAL_MEASURE_VOLTAGE`

17. Record the measurement.
18. Repeat steps 3 through 17 for each iteration of channel *x* in Table 10.

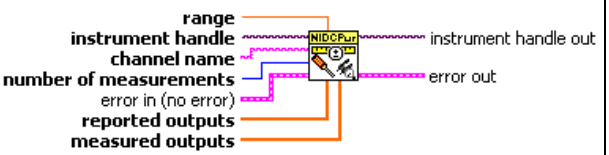
19. Set the voltage level to 0 V using the niDCPower Configure Voltage Level VI.

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_ConfigureVoltageLevel` with the following parameters: **vi**: The instrument handle from `niDCPower_InitExtCal` **channelName**: *x* **level**: `0`

20. Disable the output using the niDCPower Configure Output Enabled VI.

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_ConfigureOutputEnabled` with the following parameters: **vi**: The instrument handle from `niDCPower_InitExtCal` **channelName**: *x* **enabled**: `VI_FALSE`

21. Adjust the voltage measurement using the niDCPower Cal Adjust Voltage Measurement VI.

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_CalAdjustVoltageMeasurement` with the following parameters: **vi**: The instrument handle from `niDCPower_InitExtCal` **channelName**: *x* **range**: The voltage range of channel *x* **numberOfMeasurements**: 3 **reportedOutputs**: An array composed of the measurements you took with the NI PXI-4130 and recorded in step 17 for each iteration of channel *x* **measuredOutputs**: An array composed of the measurements you took with the DMM and recorded in step 15 for each iteration of channel *x*

22. Disconnect the DMM.
23. Repeat steps 2 through 22 for all unadjusted channels in Table 10. When you have adjusted voltage measurements on all channels, this part of the adjustment is complete.
24. End the session using the niDCPower Close External Calibration VI.

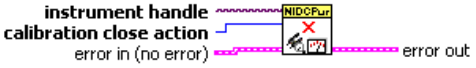
LabVIEW VI	C/C++ Function Call
	Call `niDCPower_CloseExtCal` with the following parameters: **vi**: The instrument handle from `niDCPower_InitExtCal` **action**: The instrument handle from `niDCPower_VAL_COMMIT`

Table 10. NI PXI-4130 Output Parameters for Voltage Measurement Accuracy Adjustment

Channel	Iteration	Voltage Level Range (V)	Output (V)
0	1	6	0
	2		3
	3		6
1	1	20	–20
	2		0
	3		20

Adjusting Current Programming Accuracy

Complete the following steps to adjust the current programming accuracy of the NI PXI-4130. Complete this test for each channel iteration in Table 11.

1. Open a session and obtain a session handle using the niDCPower Initialize External Calibration VI.

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_InitExtCal` with the following parameters: **resourceName**: The device name assigned by MAX **password**: `NI`

2. Connect the NI PXI-4130 channel *x* to the DMM or to the Fluke 5700A/5720A calibrator, as illustrated in Figure 2 or Figure 3.

   The setup in Figure 2 is used for the 200 µA, 2 mA, 20 mA, and 200 mA current ranges of channel 1. The setup in Figure 3 is used for the 1 A current range of channel 0 and the 2 A current range of channel 1.

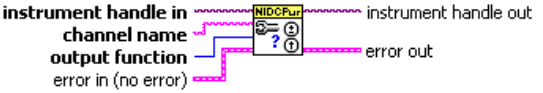
Note Channel *x* represents the channel under test. Replace the variable *x* in the program with the actual channel name.

3. Configure the DMM to the mode and range listed for the corresponding channel and range in Table 4.

4. When applicable, configure the Fluke 5700/5720A calibrator to the *Resistance* value for the corresponding channel and range in Table 4. Enable external sense (4-wire mode) on the calibrator. Record the resistance value displayed by the calibrator.
5. Place the NI PXI-4130 in delayed configuration mode using the niDCPower Abort VI.

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_Abort` with the following parameter: **vi**: The instrument handle from `niDCPower_InitExtCal`

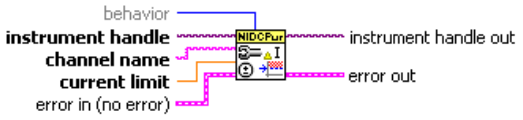
6. Set the output function to DC Voltage using the niDCPower Configure Output Function VI.

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_ConfigureOutput Function` with the following parameters: **vi**: The instrument handle from `niDCPower_InitExtCal` **channelName**: *x* **outputFunction**: `NIDCPOWER_VAL_DC_VOLTAGE`

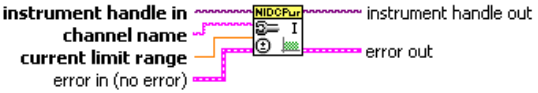
7. Configure the voltage level using the niDCPower Configure Voltage Level VI.

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_ConfigureVoltageLevel` with the following parameters: **vi**: The instrument handle from `niDCPower_InitExtCal` **channelName**: *x* **voltageLevel**: The *Voltage Level* value for the iteration of channel *x* in Table 11

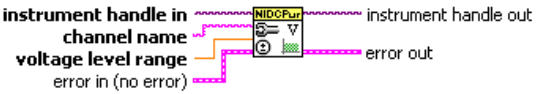
8. Configure the current limit using the niDCPower Configure Current Limit VI.

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_ConfigureCurrentLimit` with the following parameters: **vi**: The instrument handle from `niDCPower_InitExtCal` **channelName**: *x* **limit**: The *Output* value for the iteration of channel *x* in Table 11

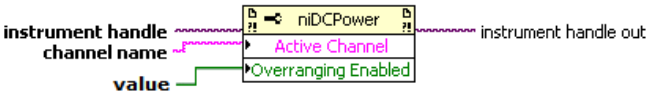
9. Configure the current limit range using the niDCPower Configure Current Limit Range VI.

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_ConfigureCurrentLimitRange` with the following parameters: **vi**: The instrument handle from `niDCPower_InitExtCal` **channelName**: *x* **currentLimitRange**: The *Current Limit Range* value for the iteration of channel *x* in Table 11

10. Configure the voltage level range using the niDCPower Configure Voltage Level Range VI.

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_ConfigureVoltageLevelRange` with the following parameters: **vi**: The instrument handle from `niDCPower_InitExtCal` **channelName**: *x* **voltageLevelRange**: The *Voltage Level Range* value for the iteration of channel *x* in Table 11

11. Enable current overranging using the niDCPower property node.

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_SetAttribute` `ViInt32` with the following parameters: **vi**: The instrument handle from `niDCPower_InitExtCal` **channelName**: *x* **attribute**: `NIDCPOWER_ATTR_OVERRANGING_ENABLED` **value**: `VI_TRUE`

12. Enable the output using the niDCPower Configure Output Enabled VI.

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_ConfigureOutputEnabled` with the following parameters: **vi**: The instrument handle from `niDCPower_InitExtCal` **channelName**: *x* **enabled**: `VI_TRUE`

13. Apply the configuration using the niDCPower Initiate VI.

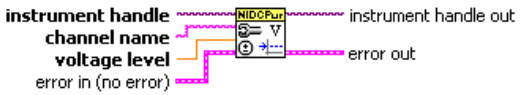
LabVIEW VI	C/C++ Function Call
	Call `niDCPower_Initiate` with the following parameter: **vi**: The instrument handle from `niDCPower_InitExtCal`

14. Wait 3 s for the output of the NI PXI-4130 to settle.
15. Measure the output voltage or current using the DMM.
16. Record the measurement.

17. For the 200 µA, 2 mA, 20 mA, and 200 mA current ranges of channel 1, divide the voltage measurement you recorded in step 16 by the resistance measurement you recorded in step 4 to calculate the output current. For the 1 A current range of channel 0 and the 2 A current range of channel 1, the output current is measured directly by the DMM.
18. Repeat steps 2 through 17 for all iterations of channel *x* in Table 11 per current range.

Note For channel 1, each current limit range has several iterations with a positive voltage level and another set of iterations with a negative voltage level.

19. Set the voltage level to 0 V using the niDCPower Configure Voltage Level VI.

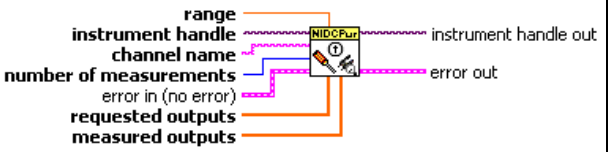
LabVIEW VI	C/C++ Function Call
	Call `niDCPower_ConfigureVoltageLevel` with the following parameters: **vi**: The instrument handle from `niDCPower_InitExtCal` **channelName**: *x* **voltageLevel**: `0`

20. Disable the output using the niDCPower Configure Output Enabled VI.

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_ConfigureOutputEnabled` with the following parameters: **vi**: The instrument handle from `niDCPower_InitExtCal` **channelName**: *x* **enabled**: `VI_FALSE`

21. Adjust the current limit using the niDCPower Cal Adjust Current Limit VI for each current limit range listed in Table 11.

Note When calibrating channel 1, each current level range requires a unique call to the niDCPower Cal Adjust Current Level VI. Positive and negative output values within the same range must be calibrated using unique calls as well.

LabVIEW VI	C/C++ Function Call
	Call `CalAdjustCurrentLimit` with the following parameters: **vi**: The instrument handle from `niDCPower_InitExtCal` **channelName**: *x* **range**: The *Current Limit Range* of channel *x* **numberOfMeasurements**: An integer value of the total number of measurements. This value should match the number of elements in the **requestedOutputs** and **measuredOutputs** arrays **requestedOutputs**: An array composed of the *Output* values for each iteration of channel *x* in Table 11 for the range to be calibrated **measuredOutputs**: An array composed of the measurements you recorded in step 16 for each iteration of channel *x* in Table 11 for the range to be calibrated

22. Repeat steps 2 through 21 for all current ranges of channel *x* in Table 11.
23. Disconnect the DMM and the calibrator.
24. Repeat steps 2 through 23 for all unadjusted channels in Table 11. When you have adjusted current programming on all channels, this part of the adjustment is complete.

25. End the session using the niDCPower Close External Calibration VI.

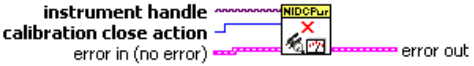
LabVIEW VI	C/C++ Function Call
	Call `niDCPower_CloseExtCal` with the following parameters: **vi**: The instrument handle from `niDCPower_InitExtCal` **action**: The instrument handle from `niDCPower_VAL_COMMIT`

Table 11. NI PXI-4130 Output Parameters and Test Limits for Current Programming Accuracy Adjustment

Channel	Current Limit Range	Voltage Level Range	Voltage Level	Iteration	Output
0	1 A	6 V	6 V	1	20 mA
				2	350 mA
				3	700 mA
1	200 µA	20 V	20 V	1	2.00 µA
				2	24.0 µA
				3	46.0 µA
				4	68.0 µA
				5	90.0 µA
				6	112 µA
				7	134 µA
				8	156 µA
				9	178 µA
				10	200 µA
			–20 V	11	2.00 µA
				12	24.0 µA
				13	46.0 µA
				14	68.0 µA
				15	90.0 µA
				16	112 µA
				17	134 µA

Table 11. NI PXI-4130 Output Parameters and Test Limits for Current Programming Accuracy Adjustment (Continued)

Channel	Current Limit Range	Voltage Level Range	Voltage Level	Iteration	Output
1	200 µA	20 V	–20 V	18	156 µA
				19	178 µA
				20	200 µA
	2 mA		20 V	1	20.0 µA
				2	0.240 mA
				3	0.460 mA
				4	0.680 mA
				5	0.900 mA
				6	1.12 mA
				7	1.34 mA
				8	1.56 mA
				9	1.78 mA
				10	2.00 mA
			–20 V	11	20.0 µA
				12	0.240 mA
				13	0.460 mA
				14	0.680 mA
				15	0.900 mA
				16	1.12 mA
				17	1.34 mA
				18	1.56 mA
				19	1.78 mA
				20	2.00 mA
	20 mA		20 V	1	0.200 mA
				2	2.40 mA
				3	4.60 mA
				4	6.80 mA
				5	9.00 mA
				6	11.2 mA

Table 11. NI PXI-4130 Output Parameters and Test Limits for Current Programming Accuracy Adjustment (Continued)

Channel	Current Limit Range	Voltage Level Range	Voltage Level	Iteration	Output
1	20 mA	20 V	20 V	7	13.4 mA
				8	15.6 mA
				9	17.8 mA
				10	20.0 mA
			–20 V	11	0.200 mA
				12	2.40 mA
				13	4.60 mA
				14	6.80 mA
				15	9.00 mA
				16	11.2 mA
				17	13.4 mA
				18	15.6 mA
				19	17.8 mA
				20	20.0 mA
	200 mA		20 V	1	0.200 mA
				2	2.40 mA
				3	4.60 mA
				4	6.80 mA
				5	9.00 mA
				6	112 mA
				7	134 mA
				8	156 mA
				9	178 mA
				10	200 mA
			–20 V	11	2.00 mA
				12	24.0 mA
				13	46.0 mA
				14	68.0 mA
				15	90.0 mA
				16	112 mA

Table 11. NI PXI-4130 Output Parameters and Test Limits for Current Programming Accuracy Adjustment (Continued)

Channel	Current Limit Range	Voltage Level Range	Voltage Level	Iteration	Output
1	200 mA	20 V	–20 V	17	134 mA
				18	156 mA
				19	178 mA
				20	200 mA
	2 A		20 V	1	20.0 mA
				2	240 mA
				3	460 mA
				4	680 mA
				5	900 mA
				6	1.12 A
				7	1.34 A
				8	1.56 A
				9	1.78 A
				10	2.00 A
			–20 V	11	20.0 mA
				12	240 mA
				13	460 mA
				14	680 mA
				15	900 mA
				16	1.12 A
				17	1.34 A
				18	1.56 A
				19	1.78 A
				20	2.00 A

Adjusting Current Measurement Accuracy

Complete the following steps to adjust the current measurement accuracy of the NI PXI-4130. Complete this test for each channel iteration in Table 12.

1. Open a session and obtain a session handle using the niDCPower Initialize External Calibration VI.

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_InitExtCal` with the following parameters: **resourceName**: The device name assigned by MAX **password**: `NI`

2. If the *Voltage Level* value for this iteration of channel *x* is 0, skip to step 5. Do *not* connect the DMM or the Fluke 5700A/5720A calibrator to the channel *x* output terminals of the NI PXI-4130.

   For all other *Output* values, connect the NI PXI-4130 channel *x* to the DMM or the Fluke 5700A/5720A calibrator, as illustrated in Figure 2 or Figure 3.

   The setup in Figure 2 is used for the 200 µA, 2 mA, 20 mA, and 200 mA current ranges of channel 1. The setup in Figure 3 is used for the 1 A current range of channel 0 and the 2 A current range of channel 1.

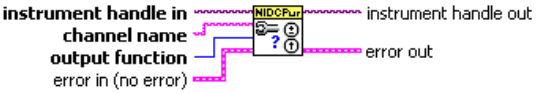
Note Channel *x* represents the channel under test. Replace the variable *x* in the program with the actual channel name.

3. Configure the DMM to the mode and range listed for the corresponding channel and range in Table 4.
4. When applicable, configure the Fluke 5700/5720A calibrator to the *Resistance* value listed for the corresponding channel and range in Table 4. Enable external sense (4-wire mode) on the calibrator. Record the resistance value displayed by the calibrator.

5. Place the NI PXI-4130 in delayed configuration mode using the niDCPower Abort VI.

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_Abort` with the following parameter: **vi**: The instrument handle from `niDCPower_InitExtCal`

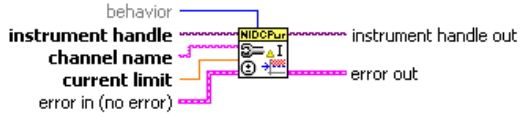
6. Set the output function to DC Voltage using the niDCPower Configure Output Function VI.

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_ConfigureOutput Function` with the following parameters: **vi**: The instrument handle from `niDCPower_InitExtCal` **channelName**: *x* **outputFunction**: `NIDCPOWER_VAL_DCVOLTAGE`

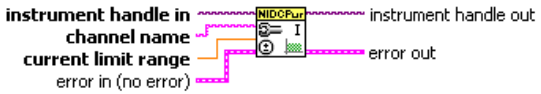
7. Configure the voltage level using the niDCPower Configure Voltage Level VI.

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_ConfigureVoltageLevel` with the following parameters: **vi**: The instrument handle from `niDCPower_InitExtCal` **channelName**: *x* **level**: The *Voltage Level* value for channel *x* in Table 12

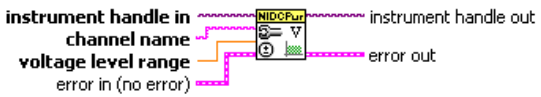
8. Configure the current limit using the niDCPower Configure Current Limit VI.

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_ConfigureCurrentLimit` with the following parameters: **vi**: The instrument handle from `niDCPower_InitExtCal` **channelName**: *x* **behavior**: `NIDCPOWER_VAL_CURRENT_REGULATE` **limit**: The *Output* value for the iteration of channel *x* in Table 12

9. Configure the current limit range using the niDCPower Configure Current Limit Range VI.

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_ConfigureCurrentLimitRange` with the following parameters: **vi**: The instrument handle from `niDCPower_InitExtCal` **channelName**: *x* **currentLimitRange**: The *Current Limit Range* value for the iteration of channel *x* in Table 12

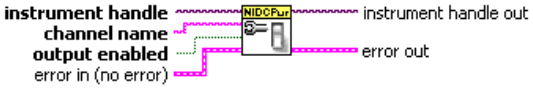
10. Configure the voltage level range using the niDCPower Configure Voltage Level Range VI.

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_ConfigureVoltageLevelRange` with the following parameters: **vi**: The instrument handle from `niDCPower_InitExtCal` **channelName**: *x* **voltageLevelRange**: The *Voltage Level Range* value for the iteration of channel *x* in Table 12

11. Specify the samples to average using the niDCPower property node.

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_SetAttribute ViInt32` with the following parameters: **vi**: The instrument handle from `niDCPower_InitExtCal` **channelName**: *x* **attribute**: `NIDCPOWER_ATTR_SAMPLES_TO_AVERAGE` **value**: `300`

12. Enable the output using the niDCPower Configure Output Enabled VI.

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_ConfigureOutput Enabled` with the following parameters: **vi**: The instrument handle from `niDCPower_InitExtCal` **channelName**: *x* **enabled**: `VI_TRUE`

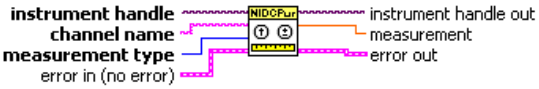
13. Apply the configuration using the niDCPower Initiate VI.

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_Initiate` with the following parameter: **vi**: The instrument handle from `niDCPower_InitExtCal`

14. Wait 3 s for the output of the NI PXI-4130 to settle.
15. If the *Output* value for this iteration of channel *x* is 0, skip to step 16. For all other *Output* values, measure the output voltage or current using the DMM.
16. Record the measurement. If the *Voltage Level* value for this iteration of channel *x* is 0, you did not take a measurement; record “0” in place of the measurement.
17. For the 200 µA, 2 mA, 20 mA, and 200 mA current ranges of channel 1, divide the voltage measurement you recorded in step 16

by the resistance measurement you recorded in step 4 to calculate the output current. For the 1 A current range of channel 0 and the 2 A current range of channel 1, the output current is measured directly by the DMM in step 15.

18. Measure the output current using the niDCPower Measure VI.

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_Measure` with the following parameters: **vi**: The instrument handle from `niDCPower_InitExtCal` **channelName**: *x* **measurementType**: `NIDCPOWER_VAL_MEASURE_CURRENT`

19. Record the measurement.
20. Repeat steps 2 through 19 for all iterations of channel *x* in Table 12 per current range.
21. Set the voltage level to 0 V using the niDCPower Configure Voltage Level VI.

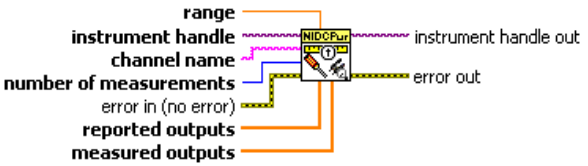
LabVIEW VI	C/C++ Function Call
	Call `niDCPower_ConfigureVoltageLevel` with the following parameters: **vi**: The instrument handle from `niDCPower_InitExtCal` **channelName**: *x* **voltageLevel**: `0`

22. Disable the output using the niDCPower Configure Output Enabled VI.

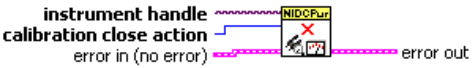
LabVIEW VI	C/C++ Function Call
	Call `niDCPower_ConfigureOutputEnabled` with the following parameters: **vi**: The instrument handle from `niDCPower_InitExtCal` **channelName**: *x* **enabled**: `VI_FALSE`

23. Adjust the current measurement using the niDCPower Cal Adjust Current Measurement VI.

Note When calibrating channel 1, each current limit range requires a unique call to the niDCPower Cal Adjust Current Measurement VI.

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_CalAdjustCurrentMeasurement` with the following parameters: **vi**: The instrument handle from `niDCPower_InitExtCal` **channelName**: *x* **range**: The voltage range of channel *x* **numberOfMeasurements**: 3 **reportedOutputs**: An array composed of the measurements you took with the NI PXI-4130 and recorded in step 19 for each iteration of channel *x* in Table 12 for the range to be calibrated **measuredOutputs**: An array composed of the measurements you took with the DMM and recorded in step 16 for each iteration of channel *x* in Table 12 for the range to be calibrated

24. Disconnect the DMM and the calibrator.
25. Repeat steps 2 through 24 for all unadjusted channels in Table 12. When you have adjusted current measurement on all channels, this part of the adjustment is complete.
26. End the session using the niDCPower Close External Calibration VI.

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_CloseExtCal` with the following parameters: **vi**: The instrument handle from `niDCPower_InitExtCal` **action**: The instrument handle from `niDCPower_VAL_COMMIT`

When you have successfully completed all adjustment tests, adjustment of the NI PXI-4130 is complete. Repeat the *Verification* section to reverify the performance of the NI PXI-4130 post-adjustment. If the NI PXI-4130 successfully passes all verification tests, calibration is complete.

Table 12. NI PXI-4130 Output Parameters and Test Limits for Current Measurement Accuracy Adjustment

Channel	Current Limit Range	Voltage Level Range	Voltage Level	Iteration	Output (A)
0	1 A	6 V	0 V	1	3.50E-01
			6 V	2	3.50E-01
				3	7.00E-01
1	200 µA	20 V	0 V	1	2.00E-04
			20 V	2	
			–20 V	3	
	2 mA		0 V	1	2.00E-03
			20 V	2	
			–20 V	3	
	20 mA		0 V	1	2.00E-02
			20 V	2	
			–20 V	3	
	200 mA		0 V	1	2.00E-01
			20 V	2	
			–20 V	3	
	2 A		0 V	1	8.00E-01
			20 V	2	
			–20 V	3	

Appendix A: Calibration Options

Calibration involves verification, and, if necessary, adjustment, and reverification of the NI PXI-4130.

Verification is the process of testing to ensure that the accuracy of the device is within certain specifications, or calibration test limits. Perform verification to determine if the device requires adjustment, or perform verification post-adjustment to determine if the adjustment was successful.

Note In this document, the calibration test limits are the published *NI PXI-4130 Specifications*.

Adjustment is the process of measuring and compensating for device performance to improve the measurement accuracy. Performing an adjustment updates the calibration date.

The *Complete Calibration* section details the recommended calibration procedure. If the device meets its calibration test limits and you prefer to skip adjustment, the *Optional Calibration* section details alternative calibration procedures.

Complete Calibration

Perform a complete calibration to guarantee that the NI PXI-4130 meets or exceeds its published specifications for a one-year calibration interval. After adjustment, repeat verification to ensure that the device meets the calibration test limits. Figure 5 shows the programming flow for a complete calibration.

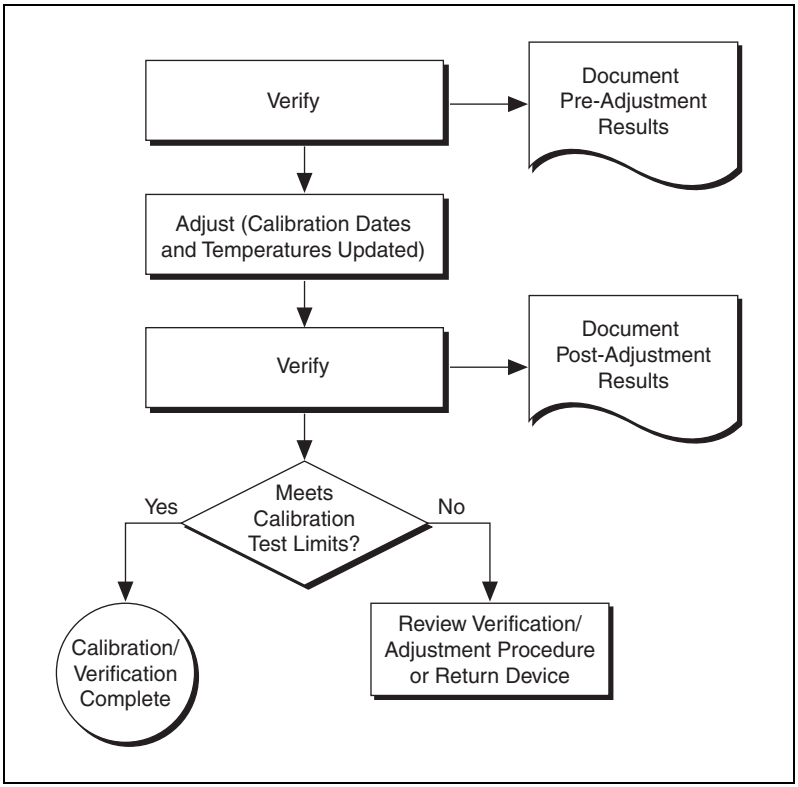


Figure 5. Complete Calibration Programming Flow

Optional Calibration

If the accuracy of the device is within the calibration test limits during the initial verification, the NI PXI-4130 meets its published specifications, and you can skip the adjustment steps of the calibration procedure. If you choose to skip the adjustment, you can update the calibration date, effectively resetting the calibration interval. Refer to the *Adjustment* section for more information.

If you choose to perform an adjustment without verification, you must still verify that the accuracy of the device is within the calibration test limits post-adjustment.

Figure 6 shows the programming flow for the optional calibration.

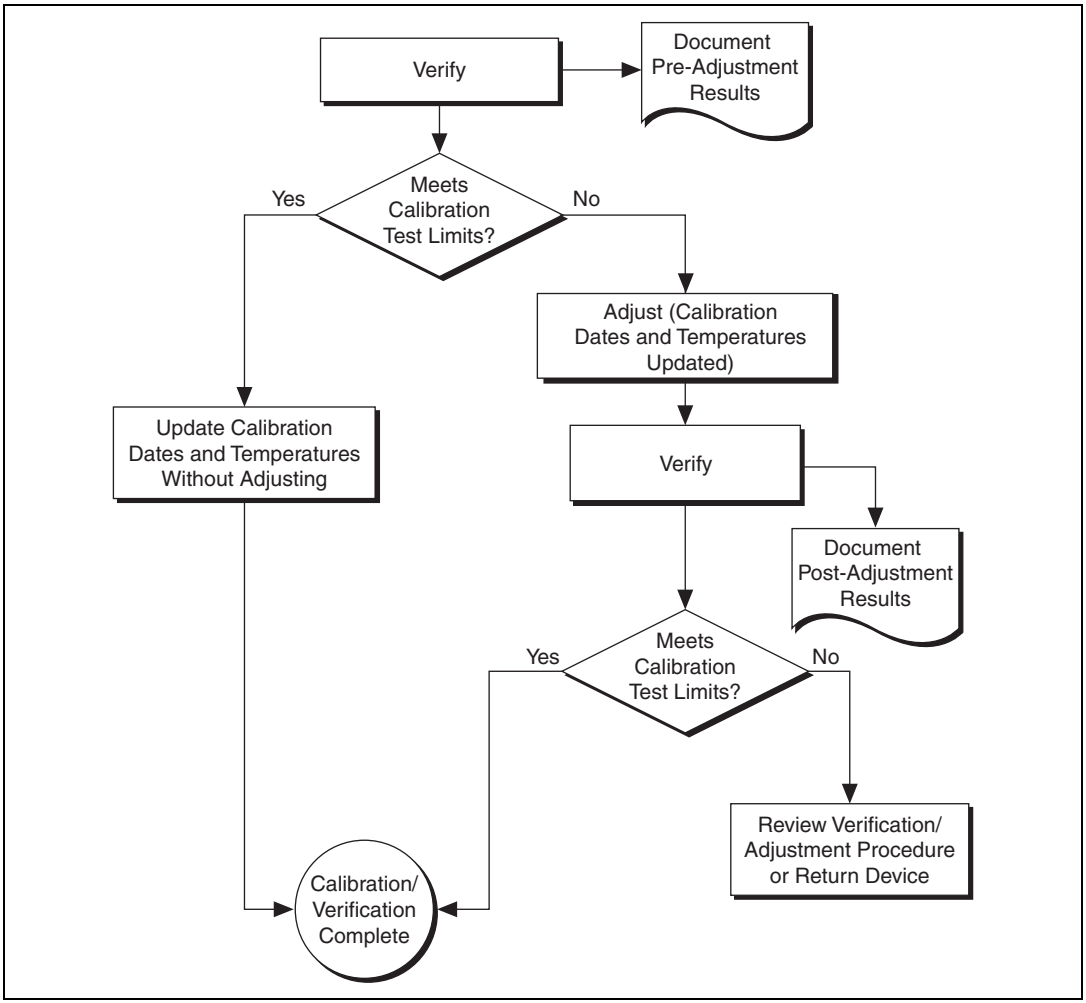


Figure 6. Optional Calibration Programming Flow

Appendix B: Calibration Utilities

NI-DCPower provides a full complement of calibration utility functions and VIs. You can use utility functions and VIs to retrieve information about adjustments performed on the NI PXI-4130, change the calibration password, and store small amounts of information in the onboard EEPROM. Refer to the *NI DC Power Supplies and SMUs Help* for the complete function reference and VI reference. The utility functions include:

- niDCPower Change Ext Cal Password VI (`niDCPower_ChangeExtCalPassword`)
- niDCPower Get Ext Cal Recommended Interval VI (`niDCPower_GetExtCalRecommendedInterval`)
- niDCPower Get Ext Cal Last Date And Time VI (`niDCPower_GetExtCalLastDateAndTime`)
- niDCPower Get Cal User Defined Info Max Size VI (`niDCPower_GetCalUserDefinedInfoMaxSize`)
- niDCPower Set Cal User Defined Info VI (`niDCPower_SetCalUserDefinedInfo`)
- niDCPower Get Cal User Defined Info VI (`niDCPower_GetCalUserDefinedInfo`)
- niDCPower Read Current Temperature VI (`niDCPower_ReadCurrentTemperature`)
- niDCPower Get Ext Cal Last Temp VI (`niDCPower_GetExtCalLastTemp`)

Calibration Function References

The VIs and functions used in this procedure, including all calibration VIs and functions, are documented in the *NI-DCPower VI Reference Help* and the *NI-DCPower Function Reference Help*, both of which you can access from the *NI DC Power Supplies and SMUs Help* at **Start»All Programs» National Instruments»NI-DCPower»Documentation**.

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, contact your local branch office:

Australia 1800 300 800, Austria 43 662 457990-0,
Belgium 32 (0) 2 757 0020, Brazil 55 11 3262 3599,
Canada 800 433 3488, China 86 21 5050 9800,
Czech Republic 420 224 235 774, Denmark 45 45 76 26 00,
Finland 358 (0) 9 725 72511, France 01 57 66 24 24,
Germany 49 89 7413130, India 91 80 41190000, Israel 972 3 6393737,
Italy 39 02 41309277, Japan 0120-527196, Korea 82 02 3451 3400,
Lebanon 961 (0) 1 33 28 28, Malaysia 1800 887710,
Mexico 01 800 010 0793, Netherlands 31 (0) 348 433 466,
New Zealand 0800 553 322, Norway 47 (0) 66 90 76 60,
Poland 48 22 3390150, Portugal 351 210 311 210, Russia 7 495 783 6851,
Singapore 1800 226 5886, Slovenia 386 3 425 42 00,
South Africa 27 0 11 805 8197, Spain 34 91 640 0085,
Sweden 46 (0) 8 587 895 00, Switzerland 41 56 2005151,
Taiwan 886 02 2377 2222, Thailand 662 278 6777,
Turkey 90 212 279 3031, United Kingdom 44 (0) 1635 523545

374753B Nov08

キャリブレーション手順
NI PXI-4130

このドキュメントには、NI PXI-4130 プログラマブル高出力ソースメジャーユニット（SMU）のキャリブレーションを行うプログラムの記述手順が記載されています。キャリブレーションは通常、ナショナルインスツルメンツ、または外部高精度デジタルマルチメータ（DMM）などのテスト機器を備えた測定室で行われます。キャリブレーションの詳細については、ni.com/calibration（英語）を参照してください。

目次

NATIONAL INSTRUMENTS

表記規則

このドキュメントでは、以下の表記規則を使用します。

→ 矢印（→）は、ネストされたメニュー項目やダイアログボックスのオプションをたどっていくと目的の操作項目を選択できることを示します。たとえば、**ファイル→ページ設定→オプション**となっている場合は、**ファイル**メニューをプルダウンして、**ページ設定**項目を選択し、最後のダイアログボックスから**オプション**を選択します。

このアイコンは、注意すべき重要な情報を示します。

太字 太字のテキストは、メニュー項目やダイアログボックスなど、ソフトウェアでユーザが選択またはクリックする必要のある項目を示します。また、パラメータ名、強調、主要概念を示します。

斜体 斜体のテキストは、変数、強調、相互参照、または重要な概念の説明を示します。また、ユーザが入力する必要がある語または値のプレースホルダも示します。

`monospace` このフォントのテキストは、キーボードから入力する必要があるテキストや文字、コードの一部、プログラムサンプル、構文例を表します。また、ディスクドライブ、パス、ディレクトリ、プログラム、サブプログラム、サブルーチンなどの名称、デバイス名、関数、操作、変数、ファイル名および拡張子の引用にも使用されます。

`monospace 斜体` ユーザが入力する必要がある語または値のプレースホルダを示します。

ソフトウェア要件

NI PXI-4130 のキャリブレーションを行うには、キャリブレーションシステムに NI-DCPower バージョン 1.2 以降をインストールする必要があります。NI-DCPower には、キャリブレーションに必要なすべての関数および VI が含まれています。`ni.com/idnet` から NI-DCPower の最新バージョンをダウンロードすることができます。

NI-DCPower は、C 言語と LabVIEW でのキャリブレーションプログラミングをサポートしています。LabWindows™/CVI™ では、C 言語のキャリブレーション関数は NI-DCPower 関数パネル（`niDCPower.fp`）にインストールされ、この関数パネルからアクセスすることができます。LabVIEW では、キャリブレーション VI は `niDCPower.llb` にインストールされ、LabVIEW の関数パレットからアクセスすることができます。ファイルの場所については、表 1 を参照してください。

このドキュメントでは最初に LabVIEW VI を、その後に対応する C 関数呼び出しを記載しています。C 関数呼び出しは、32 ビット DLL を呼び出すことが可能なすべてのコンパイラにおいて有効です。多くの関数は `niDCPower.h` ファイルで定義された定数を使用します。C 言語でこれら

の定数を使用するには、キャリブレーションプログラムで niDCPower.h を読み込む必要があります。

キャリブレーション VI および関数の詳細については、**スタート→すべてのプログラム→ National Instruments → NI-DCPower →ドキュメント→ NI DC 電源および SMU ヘルプ**から『NI DC 電源および SMU ヘルプ』を参照してください。

表 1 キャリブレーションファイルの場所（NI-DCPower 1.2 以降）

ファイル名および保存場所	説明
IVI¥Bin¥niDCPower_32.dll	キャリブレーション関数など、すべての NI-DCPower API を含む NI-DCPower ドライバ。
IVI¥Lib¥msc¥niDCPower.lib	キャリブレーション関数など、すべての NI-DCPower API を含む Microsoft C 用 NI-DCPower ライブラリ。
<*LabVIEW*>¥instr.lib¥niDCPower Calibrate¥niDCPower.llb	NI-DCPower キャリブレーション API 呼び出し用の VI を含む LabVIEW VI ライブラリ。LabVIEW 関数パレットの、NI-DCPower キャリブレーションパレットからキャリブレーション関数にアクセス可能。
IVI¥Drivers¥niDCPower¥niDCPower.fp	CVI 環境でのキャリブレーション関数プロトタイプ、および NI-DCPower の使用についてのヘルプを含む CVI 関数パネルファイル。
IVI¥Include¥niDCPower.h	キャリブレーション関数を使用するすべての C プログラムに必要な NI-DCPower ヘッダファイル。キャリブレーション関数など、すべての NI-DCPower API を含む。

ドキュメント要件

キャリブレーションプログラムの記述にあたり、必要に応じて以下のドキュメントも参照してください。

- 『NI PXI-4130 仕様』
- 『NI DC 電源および SMU スタートアップガイド』
- 『NI DC 電源および SMU ヘルプ』（LabVIEW VI、C 関数プログラミングリファレンスを含む）

上記のドキュメントは、NI-DCPower と共にインストールされます。また、ni.com/manuals から最新バージョンをダウンロードすることもできます。

パスワード

パスワード保護されている操作のデフォルトのパスワードは、「NI」です。

キャリブレーション間隔

アプリケーションの測定確度要件によって、デバイスのキャリブレーションを行う頻度が決まります。ナショナルインスツルメンツは、1 年に少なくとも 1 回は NI PXI-4130 の完全なキャリブレーションを行うことを推奨しています。アプリケーションで求められる確度に応じて、このキャリブレーション間隔を短くしてください。詳細については、「付録 A: キャリブレーションオプション」を参照してください。

テスト装置

表 2 は、NI PXI-4130 のキャリブレーションに必要な機器を示しています。推奨する機器がない場合は、表 2 に示した仕様を参考に、代わりのキャリブレーション基準を用意してください。

表 2 NI PXI-4130 のキャリブレーションに必要な機器の仕様

必要機器	推奨機器	仕様
デジタルマルチメータ（DMM）	NI 4071	電圧 : 確度 ≤ ±50 ppm、分解能 ≤30 µV。 電流 : 確度 ≤ ±0.04 %、分解能 ≤1 µA。
外部キャリブレータ	Fluke 5700A/5720A	—
補助電源	NI APS-4100	11 V ～ 15.5 V、5 A
ツイストペア、シールドケーブル	Belden 83319E 009100	18 AWG ～ 22 AWG

テスト条件

機器の接続とキャリブレーション実行環境を最適化するため、以下のガイドラインに従ってください。

- ケーブルの長さをできるだけ短くします。長いケーブル / ワイヤはアンテナのような働きをするため、余分なノイズが取り込まれ測定結果に影響します。ノイズをさらに低減するため、信号線と共通線をより合わせます。
- フロントパネルを含むすべての接続が正しく接続されていることを確認します。
- PXI シャーシのファン速度が HIGH に設定され、ファンフィルタが清潔な状態であり、空のスロットにはフィラーパネルが取り付けられていることを確認します。詳細については、`ni.com/manuals` から入手できるドキュメント『強制空冷の維持について』を参照してください。
- 相対湿度を 10% ～ 90%（結露なきこと）に維持します。
- 周囲温度を 23 ℃ ±5 ℃に維持します。
- NI-DCPower ドライバがロードされた後に、少なくとも 30 分間のウォームアップ時間を確保します。明示的に無効にされない限り、NI-DCPower はオペレーティングシステムと共に自動的にロードされ、デバイスを有効にします。ウォームアップ時間を確保することで、NI PXI-4130 の測定回路の動作温度が安定します。

キャリブレーション手順

以下の手順でキャリブレーションを実行します。

1. 「初期設定」: デバイスを取り付け、Measurement & Automation Explorer（MAX）でデバイスを構成します。
2. 「検証」: デバイスの現在の動作を検証します。この手順では、キャリブレーションを行う前にデバイスが指定したレンジで動作していることを確認します。
3. 「調整」: 既知の電圧ソースを基に、キャリブレーション定数を調節するデバイスの外部調整を行います。この調整手順は、キャリブレーション日時を EEPROM に自動的に保存してトレーサビリティを維持します。
4. 再検証 : 上記の調整手順を実行した後、検証手順を繰り返し、デバイスが仕様の範囲内で動作していることを確認します。

以下のセクションでこれらの手順の詳細について説明します。

メモ 完全な外部キャリブレーション手順は、SMU の性能検証、キャリブレーション定数の調整、および調整後の再検証で構成されます。完全なキャリブレーション手順が不要な場合もあります。詳細については、「付録 A: キャリブレーションオプション」を参照してください。

初期設定

ソフトウェアのインストールおよびハードウェアの取り付け方法、MAX でのデバイス構成方法については、『NI DC 電源および SMU スタートアップガイド』を参照してください。

検証

このセクションでは、NI PXI-4130 公開仕様の検証のために記述するプログラムについて説明します。

検証には、NI PXI-4130 を使用した出力の生成 / 測定、DMM での確度検証、その結果とキャリブレーションテスト制限値との比較が含まれます。検証結果がテスト制限値の範囲内で、NI PXI-4130 が公開仕様を満たしている場合、調整手順は任意です。検証結果がテスト制限値の範囲外である場合は、NI PXI-4130 を調整する必要があります。

検証により、以下の NI PXI-4130 仕様をテストします。

- 電圧プログラミング確度
- 電圧測定確度
- 電流プログラミング確度
- 電流測定確度

表 3 と 4 には、検証に必要なキャリブレーション機器の構成情報が記載されています。

表 3 電圧プログラミングと測定検証 / 調整用のキャリブレーション機器構成

NI PXI-4130		DMM*		
チャンネル	**レンジ**	**機能**	**レンジ†**	**入力インピーダンス†**
0	6 V	DC 電圧	10 V	10 GΩ
1	6 V		10 V	10 GΩ
	20 V		100 V	10 MΩ

* DMM で使用可能な最高の分解能を使用。DMM の分解能は 6.5 桁以上である必要あり。

† NI 4071 DMM を想定。その他すべての DMM では、この表の値に最も近いレンジと入力インピーダンスを使用。

表 4 電流プログラミングと測定検証 / 調整用のキャリブレーション機器構成

NI PXI-4130		DMM*				
チャンネル	レンジ	機能	レンジ	入力インピーダンス	分解能（桁）	キャリブレータ抵抗
0	1 A	DC 電流	1 A	なし	6.5	なし
1	200 µA	DC 電圧	10 V	10 GΩ	7.5	10 kΩ
	2 mA	DC 電圧	10 V	10 GΩ	7.5	1 kΩ
	20 mA	DC 電圧	10 V	10 GΩ	7.5	100 Ω
	200 mA	DC 電圧	10 V	10 GΩ	7.5	10 Ω
	2 A	DC 電流	3 A	なし	6.5	なし

* DMM で使用可能な最高の分解能を使用。

メモ 以下の手順において、LabVIEW の入力値については「C/C++ 関数呼び出し」に記載された値を参照してください。

このセクションのすべてのテストが成功して初めて、NI PXI-4130 のすべての検証が完了します。

メモ 調整後に検証が失敗した場合、ナショナルインスツルメンツに NI PXI-4130 の修理を依頼する前に、必要な「テスト条件」を満たしたかどうかを確認してください。

電圧プログラミング確度を検証する

以下の手順を実行して、NI PXI-4130 の電圧プログラミング確度を検証します。表 5 の各反復に対してこのテストを実行します。

1. 「niDCPower 初期化」VI によりセッションを開き、セッションハンドルを取得します。

LabVIEW VI	C/C++ 関数呼び出し
	以下のパラメータを使用して、「niDCPower_init」を呼び出します。 **resourceName**: MAX で割り当てられたデバイス名 **idQuery**: VI_FALSE **resetDevice**: VI_TRUE

2. 図 1 のように、DMM を NI PXI-4130 のチャンネル *x* 出力端子に接続します。

メモ チャンネル *x* は、テスト中のチャンネルを表します。プログラムの変数 *x* を実際のチャンネル名と置き換えてください。

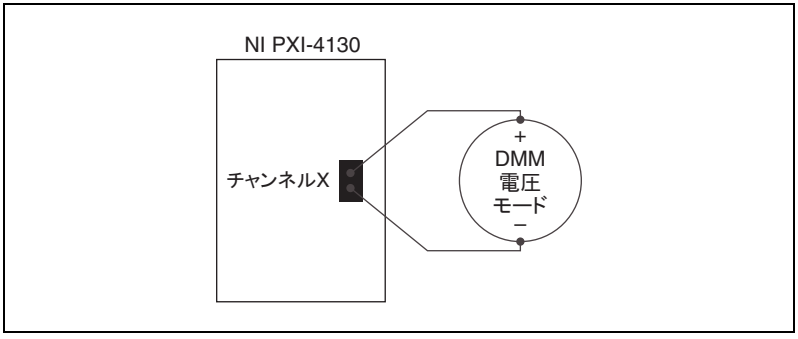


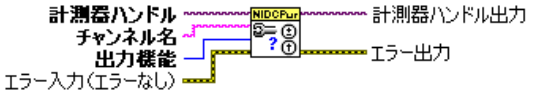
図 1 NI PXI-4130 での電圧確度検証 / 調整設定

3. 表 3 に記載したチャンネルとレンジに対応するように DMM のモードとレンジを構成します。

4. 「niDCPower 中止」VI を使用して、NI PXI-4130 を遅延構成モードにします。

LabVIEW VI	C/C++ 関数呼び出し
	以下のパラメータを使用して、「niDCPower_Abort」を呼び出します。 **vi**: 「niDCPower_init」からの計測器ハンドル

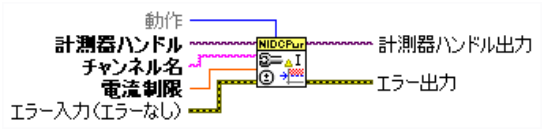
5. 「niDCPower 出力機能を構成」VI を使用して、出力機能を「DC 電圧」に設定します。

LabVIEW VI	C/C++ 関数呼び出し
	以下のパラメータを使用して、「niDCPower_ConfigureOutputFunction」を呼び出します。 **vi**: 「niDCPower_init」からの計測器ハンドル **channelName**: *x* **outputFunction**: NIDCPOWER_VAL_DC_VOLTAGE

6. 「niDCPower 電圧レベルを構成」VI を使用して電圧レベルを構成します。

LabVIEW VI	C/C++ 関数呼び出し
	以下のパラメータを使用して、「niDCPower_ConfigureVoltageLevel」を呼び出します。 **vi**: 「niDCPower_init」からの計測器ハンドル **channelName**: *x* **level**: 表 5 に記載したチャンネル *x* の反復に対応する出力値

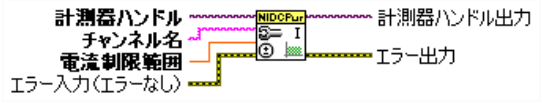
7. 「niDCPower 電流制限を構成」VI を使用して、電流制限を構成します。

LabVIEW VI	C/C++ 関数呼び出し
	以下のパラメータを使用して、「niDCPower_ConfigureCurrentLimit」を呼び出します。 **vi**: 「niDCPower_init」からの計測器ハンドル **channelName**: *x* **behavior**: NIDCPOWER_VAL_CURRENT_REGULATE **limit**: 0.5

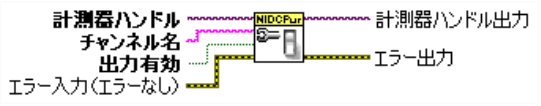
8. 「niDCPower 電圧レベル範囲を構成」VI を使用して、電圧レベル範囲を構成します。

LabVIEW VI	C/C++ 関数呼び出し
	以下のパラメータを使用して、「niDCPower_ConfigureVoltageLevelRange」を呼び出します。 **vi**: 「niDCPower_init」からの計測器ハンドル **channelName**: *x* **voltageLevelRange**: 表 5 に記載したチャンネル *x* の反復に対応する電圧レベル範囲値

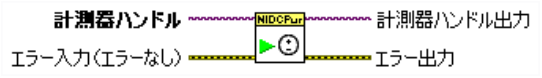
9. 「niDCPower 電流制限範囲を構成」VI を使用して、電流制限範囲を構成します。

LabVIEW VI	C/C++ 関数呼び出し
	以下のパラメータを使用して、「niDCPower_ConfigureCurrentLimitRange」を呼び出します。 **vi**: 「niDCPower_init」からの計測器ハンドル **channelName**: *x* **currentLimitRange**: 1

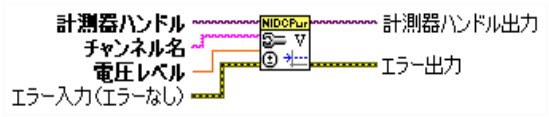
10. 「niDCPower 出力有効化を構成」VI を使用して、出力を有効にします。

LabVIEW VI	C/C++ 関数呼び出し
	以下のパラメータを使用して、「niDCPower_ConfigureOutputEnabled」を呼び出します。 **vi**: 「niDCPower_init」からの計測器ハンドル **channelName**: *x* **enabled**: VI_TRUE

11. 「niDCPower 開始」VI を使用して構成を適用します。

LabVIEW VI	C/C++ 関数呼び出し
	以下のパラメータを使用して、「niDCPower_Initiate」を呼び出します。 **vi**: 「niDCPower_init」からの計測器ハンドル

12. NI PXI-4130 の出力が整定するまで 3 秒間待機します。
13. DMM を使用して出力電圧を測定します。
14. 測定値を記録します。
15. 出力誤差を計算するには、手順 14 で記録した測定値とチャンネル *x* の反復に対応する出力値の差を求めます。
16. 出力誤差と表 5 に記載したチャンネル *x* の反復に対応するテスト制限値を比較します。出力誤差がテスト制限値の範囲外である場合、NI PXI-4130 を調整する必要があります。
17. 表 5 に示したチャンネル *x* の各反復に対して手順 3～16 を繰り返します。
18. 「niDCPower 電圧レベルを構成」VI を使用して、電圧レベルを 0 に設定します。

LabVIEW VI	C/C++ 関数呼び出し
	以下のパラメータを使用して、「niDCPower_ConfigureVoltageLevel」を呼び出します。 **vi**: 「niDCPower_init」からの計測器ハンドル **channelName**: *x* **level**: 0

19. 「niDCPower 出力有効化を構成」VI を使用して、出力を無効にします。

LabVIEW VI	C/C++ 関数呼び出し
	以下のパラメータを使用して、「niDCPower_ConfigureOutputEnabled」を呼び出します。 **vi**: 「niDCPower_init」からの計測器ハンドル **channelName**: *x* **enabled**: VI_FALSE

20. DMM との接続を切断します。
21. 表 5 に示したすべての未検証のチャンネルに対し、手順 2 ～ 20 を繰り返します。各チャンネルにおいてすべての反復を検証すれば、この部分の検証は完了です。
22. 「niDCPower 閉じる」VI を使用してセッションを終了します。

LabVIEW VI	C/C++ 関数呼び出し
	以下のパラメータを使用して、「niDCPower_close」を呼び出します。 **vi**: 「niDCPower_init」からの計測器ハンドル

表 5 NI PXI-4130 出力パラメータとテスト制限値（電圧プログラミング確度検証時）

チャンネル	電圧レベル範囲（V）	反復	出力（V）	テスト制限値（mV）
0	6	1	0	±4.00
		2	1.5	±4.75
		3	3	±5.50
		4	4.5	±6.25
		5	6	±7.00
1	6	1	0	±1.50
		2	1.5	±2.01
		3	3	±2.52
		4	4.5	±3.03
		5	6	±3.54
		6	−1.5	±2.01
		7	−3	±2.52
		8	−4.5	±3.03
		9	−6	±3.54
	20	1	0	±1.80
		2	5	±3.50
		3	10	±5.20
		4	15	±6.90
		5	20	±8.60
		6	−5	±3.50
		7	−10	±5.20
		8	−15	±6.90
		9	−20	±8.60

電圧測定確度を検証する

以下の手順を実行して、NI PXI-4130 の電圧測定確度を検証します。表 6 の各反復に対してこのテストを実行します。

1. 「niDCPower 初期化」VI によりセッションを開き、セッションハンドルを取得します。

LabVIEW VI	C/C++ 関数呼び出し
	以下のパラメータを使用して、「niDCPower_init」を呼び出します。 **resourceName**: MAX で割り当てられたデバイス名 **idQuery**: VI_FALSE **resetDevice**: VI_TRUE

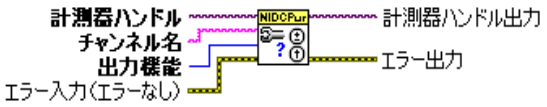
2. 図 1 に示すように、DMM を NI PXI-4130 のチャンネル *x* 出力端子に接続します。

メモ チャンネル *x* は、テスト中のチャンネルを表します。プログラムの変数 *x* を実際のチャンネル名と置き換えてください。

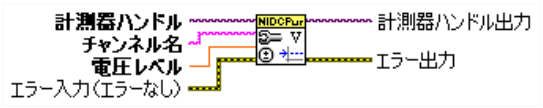
3. 表 3 に記載したチャンネルとレンジに対応するように DMM のモードとレンジを構成します。
4. 「niDCPower 中止」VI を使用して、NI PXI-4130 を遅延構成モードにします。

LabVIEW VI	C/C++ 関数呼び出し
	以下のパラメータを使用して、「niDCPower_Abort」を呼び出します。 **vi**: 「niDCPower_init」からの計測器ハンドル

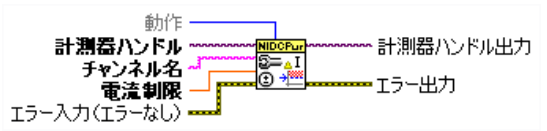
5. 「niDCPower 出力機能を構成」VI を使用して、出力機能を「DC 電圧」に設定します。

LabVIEW VI	C/C++ 関数呼び出し
	以下のパラメータを使用して、「niDCPower_ConfigureOutputFunction」を呼び出します。 **vi**: 「niDCPower_init」からの計測器ハンドル **channelName**: *x* **outputFunction**: NIDCPOWER_VAL_DC_VOLTAGE

6. 「niDCPower 電圧レベルを構成」VI を使用して電圧レベルを構成します。

LabVIEW VI	C/C++ 関数呼び出し
	以下のパラメータを使用して、「niDCPower_ConfigureVoltageLevel」を呼び出します。 **vi**: 「niDCPower_init」からの計測器ハンドル **channelName**: *x* **level**: 表 6 に記載したチャンネル *x* の反復に対応する出力値

7. 「niDCPower 電流制限を構成」VI を使用して、電流制限を構成します。

LabVIEW VI	C/C++ 関数呼び出し
	以下のパラメータを使用して、「niDCPower_ConfigureCurrentLimit」を呼び出します。 **vi**: 「niDCPower_init」からの計測器ハンドル **channelName**: *x* **behavior**: NIDCPOWER_VAL_CURRENT_REGULATE **limit**: 0.5

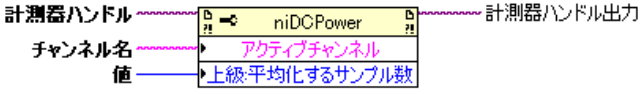
8. 「niDCPower 電圧レベル範囲を構成」VI を使用して、電圧レベル範囲を構成します。

LabVIEW VI	C/C++ 関数呼び出し
	以下のパラメータを使用して、「niDCPower_ConfigureVoltageLevelRange」を呼び出します。 **vi**: 「niDCPower_init」からの計測器ハンドル **channelName**: *x* **voltageLevelRange**: 表 6 に記載したチャンネル *x* の反復に対応する電圧レベル範囲値

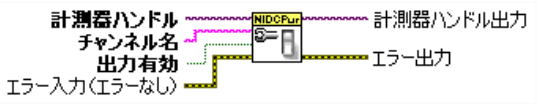
9. 「niDCPower 電流制限範囲を構成」VI を使用して、電流制限範囲を構成します。

LabVIEW VI	C/C++ 関数呼び出し
	以下のパラメータを使用して、「niDCPower_ConfigureCurrentLimitRange」を呼び出します。 **vi**: 「niDCPower_init」からの計測器ハンドル **channelName**: *x* **currentLimitRange**: 1

10. niDCPower プロパティノードを使用して「平均化するサンプル数」を指定します。

LabVIEW VI	C/C++ 関数呼び出し
	以下のパラメータを使用して、「niDCPower_SetAttributeViInt32」を呼び出します。 **vi**: 「niDCPower_init」からの計測器ハンドル **channelName**: *x* **attribute**: NIDCPOWER_ATTR_SAMPLES_TO_AVERAGE **value**: 300

11. 「niDCPower 出力有効化を構成」VI を使用して、出力を有効にします。

LabVIEW VI	C/C++ 関数呼び出し
	以下のパラメータを使用して、「niDCPower_ConfigureOutputEnabled」を呼び出します。 **vi**: 「niDCPower_init」からの計測器ハンドル **channelName**: *x* **enabled**: VI_TRUE

12. 「niDCPower 開始」VI を使用して構成を適用します。

LabVIEW VI	C/C++ 関数呼び出し
	以下のパラメータを使用して、「niDCPower_Initiate」を呼び出します。 **vi**: 「niDCPower_init」からの計測器ハンドル

13. NI PXI-4130 の出力が整定するまで 3 秒間待機します。
14. DMM を使用して出力電圧を測定します。
15. 測定値を記録します。
16. 「niDCPower 測定」VI を使用して出力電圧を測定します。

LabVIEW VI	C/C++ 関数呼び出し
	以下のパラメータを使用して、「niDCPower_Measure」を呼び出します。 **vi**: 「niDCPower_init」からの計測器ハンドル **channelName**: *x* **measurementType**: NIDCPOWER_VAL_MEASURE_VOLTAGE

17. 測定値を記録します。
18. 測定誤差を計算するには、手順 17 で記録した測定値と、手順 15 で記録した測定値の差を求めます。
19. 表 6 の「テスト制限値」列に記載されたゲインおよびオフセット値を使用して、チャンネル *x* の反復に対応するテスト制限値の上下限を計算します。DMM は固有値を測定するため、絶対制限値の代わりに許容範囲が記載されています。各制限値は、DMM の測定値のパーセ

ントとオフセット電圧の和です。測定誤差が計算した制限値の範囲内に入ることを確認します。測定誤差がテスト制限値の範囲外である場合、NI PXI-4130 を調整する必要があります。

20. 表 6 に示したチャンネル *x* の各反復に対して手順 4～19 を繰り返します。
21. 「niDCPower 電圧レベルを構成」VI を使用して、電圧レベルを 0 に設定します。

LabVIEW VI	C/C++ 関数呼び出し
	以下のパラメータを使用して、「niDCPower_ConfigureVoltageLevel」を呼び出します。 **vi**: 「niDCPower_init」からの計測器ハンドル **channelName**: *x* **level**: 0

22. 「niDCPower 出力有効化を構成」VI を使用して、出力を無効にします。

LabVIEW VI	C/C++ 関数呼び出し
	以下のパラメータを使用して、「niDCPower_ConfigureOutputEnabled」を呼び出します。 **vi**: 「niDCPower_init」からの計測器ハンドル **channelName**: *x* **enabled**: VI_FALSE

23. DMM との接続を切断します。
24. 表 6 に示したすべての未検証のチャンネルに対し、手順 2～23 を繰り返します。各チャンネルにおいてすべての反復を検証すれば、この部分の検証は完了です。
25. 「niDCPower 閉じる」VI を使用してセッションを終了します。

LabVIEW VI	C/C++ 関数呼び出し
	以下のパラメータを使用して、「niDCPower_close」を呼び出します。 **vi**: 「niDCPower_init」からの計測器ハンドル

表 6 NI PXI-4130 出力パラメータとテスト制限値（電圧測定確度の検証時）

チャンネル	電圧レベル範囲（V）	反復	出力（V）	テスト制限値（V）
0	6	1	0	0.05% + 4 mV
		2	1.5	
		3	3	
		4	4.5	
		5	6	
1	20	1	0	0.03% + 1.5 mV
		2	5	
		3	10	
		4	15	
		5	20	
		6	−5	
		7	−10	
		8	−15	
		9	−20	

電流プログラミング確度を検証する

以下の手順を実行して、NI PXI-4130 の電流プログラミング確度を検証します。表 7 に記載した各チャンネルの反復に対してこの手順を実行します。抵抗の発熱による影響を最小限に抑えるため、表 7 に示した順番通りに出力確度を検証してください。

1. 「niDCPower 初期化」VI によりセッションを開き、セッションハンドルを取得します。

LabVIEW VI	C/C++ 関数呼び出し
	以下のパラメータを使用して、「niDCPower_init」を呼び出します。 **resourceName**: MAX で割り当てられたデバイス名 **idQuery**: VI_FALSE **resetDevice**: VI_TRUE

2. 図 2 または図 3 のように、NI PXI-4130 のチャンネル x を DMM または Fluke 5700A/5720A キャリブレータに接続します。

   図 2 は、チャンネル 1 の電流範囲が 200 µA、2 mA、20 mA、200 mA の場合の設定を示し、図 3 はチャンネル 0 の電流範囲が 1 A、あるいはチャンネル 1 の電流範囲が 2 A の場合の設定を示しています。

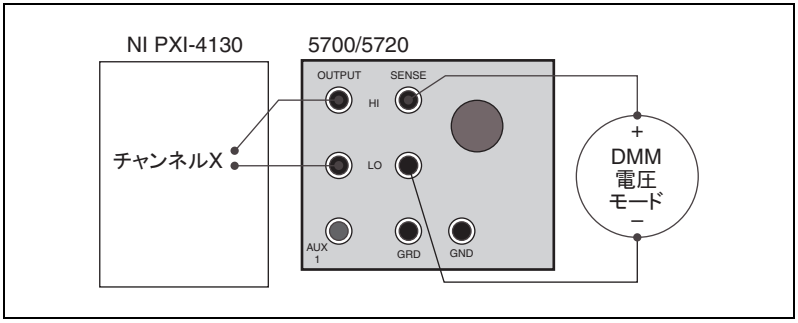


図 2 電流プログラミング確度検証の設定
（チャンネル 1 の電流範囲が 200 µA、2 mA、20 mA、200 mA の場合）

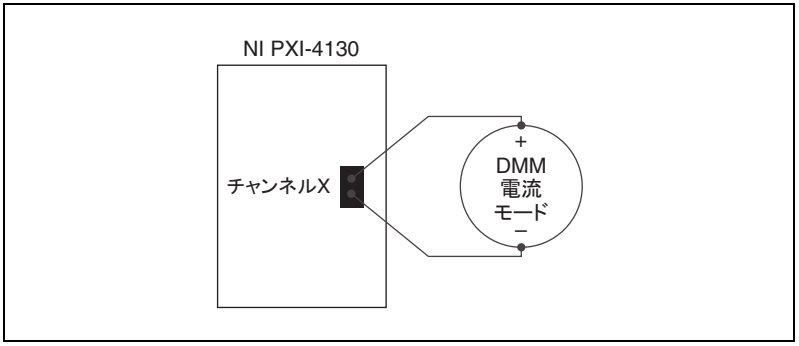


図 3　電流プログラミング確度検証の設定
（チャンネル 0 の電流範囲が 1 A、チャンネル 1 の電流範囲が 2 A の場合）

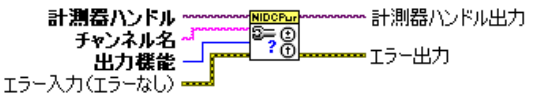
3. 表 4 に記載したチャンネルとレンジに対応するように DMM のモードとレンジを構成します。
4. 表 4 に記載したチャンネルとレンジに対する抵抗値を Fluke 5700/5720A キャリブレータに設定します。キャリブレータの外部センス（4 線式モード）を有効にします。キャリブレータに表示される実際の抵抗値を記録します。

メモ　チャンネル *x* は、テスト中のチャンネルを表します。プログラムの変数 *x* を実際のチャンネル名と置き換えてください。

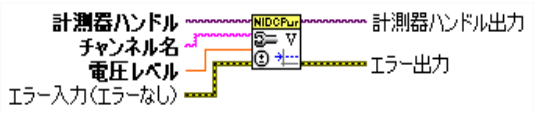
5. 「niDCPower 中止」VI を使用して、NI PXI-4130 を遅延構成モードにします。

LabVIEW VI	C/C++ 関数呼び出し
	以下のパラメータを使用して、「niDCPower_Abort」を呼び出します。 **vi**: 「niDCPower_init」からの計測器ハンドル

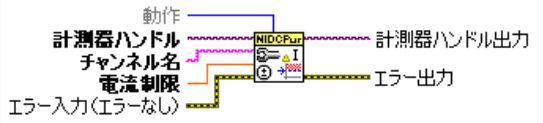
6. 「niDCPower 出力機能を構成」VI を使用して、出力機能を「DC 電圧」に設定します。

LabVIEW VI	C/C++ 関数呼び出し
	以下のパラメータを使用して、「niDCPower_ConfigureOutputFunction」を呼び出します。 **vi**: 「niDCPower_init」からの計測器ハンドル **channelName**: *x* **outputFunction**: NIDCPOWER_VAL_DC_VOLTAGE

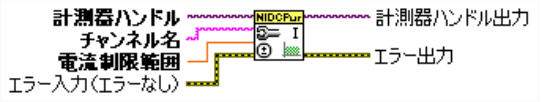
7. 「niDCPower 電圧レベルを構成」VI を使用して電圧レベルを構成します。

LabVIEW VI	C/C++ 関数呼び出し
	以下のパラメータを使用して、「niDCPower_ConfigureVoltageLevel」を呼び出します。 **vi**: 「niDCPower_init」からの計測器ハンドル **channelName**: *x* **voltageLevel**: 表 7 に記載したチャンネル *x* の反復に対応する電圧レベル値

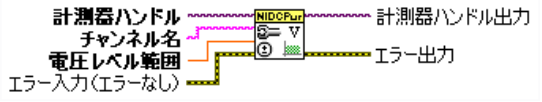
8. 「niDCPower 電流制限を構成」VI を使用して、表 7 に示したチャンネルおよび反復に対応する電流制限を構成します。

LabVIEW VI	C/C++ 関数呼び出し
	以下のパラメータを使用して、「niDCPower_ConfigureCurrentLimit」を呼び出します。 **vi**: 「niDCPower_init」からの計測器ハンドル **channelName**: *x* **currentLimit**: 表 7 に記載したチャンネル *x* の反復に対応する出力値

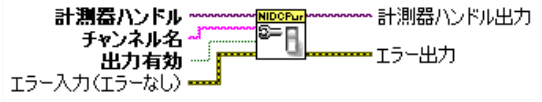
9. 「niDCPower 電流制限範囲を構成」VI を使用して、表 7 に示したチャンネルおよび反復に対応する電流制限範囲を構成します。

LabVIEW VI	C/C++ 関数呼び出し
	以下のパラメータを使用して、「niDCPower_ConfigureCurrentLimitRange」を呼び出します。 **vi**: 「niDCPower_init」からの計測器ハンドル **channelName**: *x* **currentLimitRange**: 表 7 に記載したチャンネル *x* の反復に対応する電流制限範囲値

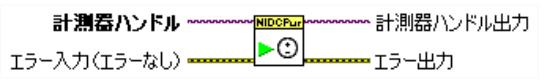
10. 「niDCPower 電圧レベル範囲を構成」VI を使用して、表 7 に示したチャンネルおよび反復に対応する電圧レベル範囲を構成します。

LabVIEW VI	C/C++ 関数呼び出し
	以下のパラメータを使用して、「niDCPower_ConfigureVoltageLevelRange」を呼び出します。 **vi**: 「niDCPower_init」からの計測器ハンドル **channelName**: *x* **voltageLevelRange**: 表 7 に記載したチャンネル *x* の反復に対応する電圧レベル範囲値

11. 「niDCPower 出力有効化を構成」VI を使用して、出力を有効にします。

LabVIEW VI	C/C++ 関数呼び出し
	以下のパラメータを使用して、「niDCPower_ConfigureOutputEnabled」を呼び出します。 **vi**: 「niDCPower_init」からの計測器ハンドル **channelName**: *x* **enabled**: VI_TRUE

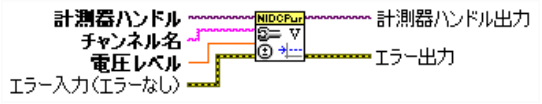
12. 「niDCPower 開始」VI を使用して構成を適用します。

LabVIEW VI	C/C++ 関数呼び出し
	以下のパラメータを使用して、「niDCPower_Initiate」を呼び出します。 **vi**: 「niDCPower_init」からの計測器ハンドル

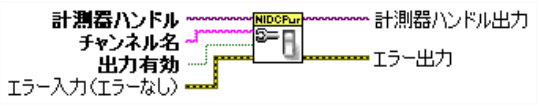
13. NI PXI-4130 の出力が整定するまで 3 秒間待機します。
14. DMM を使用して出力電圧または電流を測定します。
15. 測定値を記録します。
16. チャンネル 1 の電流範囲が 200 µA、2 mA、20 mA、200 mA の場合に出力電流を求めるには、手順 15 で記録した電圧測定値を手順 4 で記録した抵抗測定値で除算します。チャンネル 0 の電流範囲が 1 A、あるいはチャンネル 1 の電流範囲が 2 A の場合、出力電流は DMM から直接測定します。出力誤差を取得するには、上記で求めた出力電流と表 7 のチャンネル *x* の反復に対応する出力値の差を求めます。
17. 出力誤差と表 7 に記載したチャンネル *x* の反復に対応するテスト制限値を比較します。出力誤差がテスト制限値の範囲外である場合、NI PXI-4130 を調整する必要があります。
18. 表 7 に示した、チャンネル *x* の各電流範囲におけるすべての反復に対して手順 2～17 を繰り返します。

メモ チャンネル 1 の場合は、各電流制限範囲で正の電圧レベルと負の電圧レベルに対する反復があります。

19. 「niDCPower 電圧レベルを構成」VI を使用して、電圧レベルを 0 V に設定します。

LabVIEW VI	C/C++ 関数呼び出し
	以下のパラメータを使用して、「niDCPower_ConfigureVoltageLevel」を呼び出します。 **vi**: 「niDCPower_init」からの計測器ハンドル **channelName**: *x* **level**: 0

20. 「niDCPower 出力有効化を構成」VI を使用して、出力を無効にします。

LabVIEW VI	C/C++ 関数呼び出し
	以下のパラメータを使用して、「niDCPower_ConfigureOutputEnabled」を呼び出します。 **vi**: 「niDCPower_init」からの計測器ハンドル **channelName**: *x* **enabled**: VI_FALSE

21. DMM とキャリブレータの接続を切断します。
22. 表 7 に示したすべての未検証のチャンネルに対し、手順 3 ～ 21 を繰り返します。各チャンネルと範囲においてすべての反復を検証すれば、この部分の検証は完了です。
23. 「niDCPower 閉じる」VI を使用してセッションを終了します。

LabVIEW VI	C/C++ 関数呼び出し
	以下のパラメータを使用して、「niDCPower_close」を呼び出します。 **vi**: 「niDCPower_init」からの計測器ハンドル

表 7 NI PXI-4130 出力パラメータとテスト制限値（電流プログラミング確度検証時）

チャンネル	電流制限範囲	電圧レベル範囲	電圧レベル	反復	出力（A）	テスト制限値
0	1 A	6 V	6 V	1	50 mA	±4.08 mA
				2	350 mA	±4.53 mA
				3	700 mA	±5.05 mA
1	200 µA	20 V	20 V	1	4.00 µA	±0.101 µA
				2	50.0 µA	±0.115 µA
				3	0.10 mA	±0.130 µA
				4	0.15 mA	±0.145 µA
				5	0.20 mA	±0.160 µA
			−20 V	6	4.00 µA	±0.101 µA
				7	0.50 mA	±0.115 µA
				8	0.10 mA	±0.130 µA
				9	0.15 mA	±0.145 µA
				10	0.20 mA	±0.160 µA
	2 mA		20 V	1	40.0 µA	±1.01 µA
				2	0.50 mA	±1.15 µA
				3	1.00 mA	±1.30 µA
				4	1.50 mA	±1.45 µA
				5	2.00 mA	±1.60 µA
			−20 V	6	40.0 µA	±1.01 µA
				7	0.50 mA	±1.15 µA
				8	1.00 mA	±1.30 µA
				9	1.50 mA	±1.45 µA
				10	2.00 mA	±1.60 µA
	20 mA		20 V	1	0.40 mA	±10.1 µA
				2	5.00 mA	±11.5 µA
				3	10.0 mA	±13.0 µA
				4	15.0 mA	±14.5 µA

表 7 NI PXI-4130 出力パラメータとテスト制限値（電流プログラミング確度検証時）（続き）

チャンネル	電流制限範囲	電圧レベル範囲	電圧レベル	反復	出力（A）	テスト制限値
1	20 mA	20 V	20 V	5	20.0 mA	±16.0 µA
			−20 V	6	0.40 mA	±10.1 µA
				7	5.00 mA	±11.5 µA
				8	10.0 mA	±13.0 µA
				9	15.0 mA	±14.5 µA
				10	20.0 mA	±16.0 µA
	200 mA		20 V	1	4.00 mA	±0.101 mA
				2	50.0 mA	±0.115 mA
				3	100 mA	±0.130 mA
				4	150 mA	±0.145 mA
				5	200 mA	±0.160 mA
			−20 V	6	4.00 mA	±0.101 mA
				7	50.0 mA	±0.115 mA
				8	100 mA	±0.130 mA
				9	150 mA	±0.145 mA
				10	200 mA	±0.160 mA
	2 A		20 V	1	40.0 mA	±1.05 mA
				2	0.5 A	±1.60 mA
				3	1.0 A	±2.70 mA
				4	1.5 A	±4.68 mA
				5	2.0 A	±8.40 mA
			−20 V	6	40.0 mA	±1.05 mA
				7	0.5 A	±1.60 mA
				8	1.0 A	±2.70 mA
				9	1.5 A	±4.68 mA
				10	2.0 A	±8.40 mA

電流測定確度を検証する

以下の手順を実行して、NI PXI-4130 の電流測定確度を検証します。表 8 に示した範囲における、チャンネルの各反復に対してこのテストを実行します。

1. 「niDCPower 初期化」VI によりセッションを開き、セッションハンドルを取得します。

LabVIEW VI	C/C++ 関数呼び出し
	以下のパラメータを使用して、「niDCPower_init」を呼び出します。 **resourceName**: MAX で割り当てられたデバイス名 **idQuery**: VI_FALSE **resetDevice**: VI_TRUE

2. 表 8 に記載したチャンネル *x* の反復に対応する電圧レベルが 0 の場合、手順 5 に進みます。DMM または Fluke 5700A/5720A キャリブレータを NI PXI-4130 のチャンネル *x* 出力端子に接続しないでください。

   0 以外の出力値の場合、図 2 または図 3 に示すように、NI PXI-4130 のチャンネル *x* を DMM または Fluke 5700A/5720A キャリブレータに接続します。

   図 2 は、チャンネル 1 の電流範囲が 200 µA、2 mA、20 mA、200 mA の場合の設定を示し、図 3 はチャンネル 0 の電流範囲が 1 A、あるいはチャンネル 1 の電流範囲が 2 A の場合の設定を示しています。

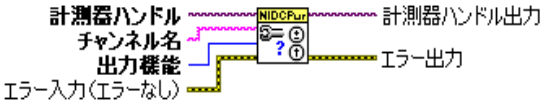
メモ チャンネル *x* は、テスト中のチャンネルを表します。プログラムの変数 *x* を実際のチャンネル名と置き換えてください。

3. 表 4 に記載したチャンネルとレンジに対応するように DMM のモードとレンジを構成します。
4. 表 4 に記載したチャンネルとレンジに対する抵抗値を Fluke 5700/5720A キャリブレータに設定します。キャリブレータの外部センス（4 線式モード）を有効にします。キャリブレータに表示される実際の抵抗値を記録します。

5. 「niDCPower 中止」VI を使用して、NI PXI-4130 を遅延構成モードにします。

LabVIEW VI	C/C++ 関数呼び出し
	以下のパラメータを使用して、「niDCPower_Abort」を呼び出します。 **vi**: 「niDCPower_init」からの計測器ハンドル

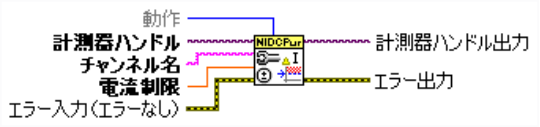
6. 「niDCPower 出力機能を構成」VI を使用して、出力機能を「DC 電圧」に設定します。

LabVIEW VI	C/C++ 関数呼び出し
	以下のパラメータを使用して、「niDCPower_ConfigureOutputFunction」を呼び出します。 **vi**: 「niDCPower_init」からの計測器ハンドル **channelName**: *x* **outputFunction**: NIDCPOWER_VAL_DC_VOLTAGE

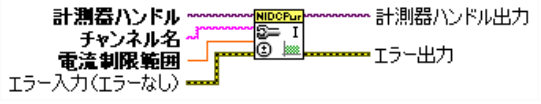
7. 「niDCPower 電圧レベルを構成」VI を使用して電圧レベルを構成します。

LabVIEW VI	C/C++ 関数呼び出し
	以下のパラメータを使用して、「niDCPower_ConfigureVoltageLevel」を呼び出します。 **vi**: 「niDCPower_init」からの計測器ハンドル **channelName**: *x* **level**: 表 8 に記載したチャンネル *x* の反復に対応する電圧レベル値

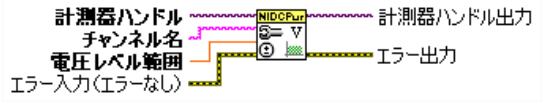
8. 「niDCPower 電流制限を構成」VI を使用して、表 8 に示したチャンネルおよび反復に対応する電流制限を構成します。

LabVIEW VI	C/C++ 関数呼び出し
	以下のパラメータを使用して、「niDCPower_ConfigureCurrentLimit」を呼び出します。 **vi**: 「niDCPower_init」からの計測器ハンドル **channelName**: *x* **currentLimit**: 表 8 に記載したチャンネル *x* の反復に対応する出力値

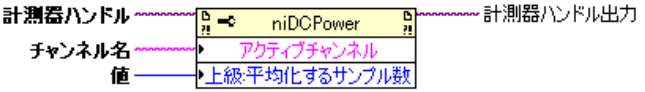
9. 「niDCPower 電流制限範囲を構成」VI を使用して、表 8 に示したチャンネルおよび反復に対応する電流制限範囲を構成します。

LabVIEW VI	C/C++ 関数呼び出し
	以下のパラメータを使用して、「niDCPower_ConfigureCurrentLimitRange」を呼び出します。 **vi**: 「niDCPower_init」からの計測器ハンドル **channelName**: *x* **currentLimitRange**: 表 8 に記載したチャンネル *x* の反復に対応する電流制限範囲値

10. 「niDCPower 電圧レベル範囲を構成」VI を使用して、表 8 に示したチャンネルおよび反復に対応する電圧レベル範囲を構成します。

LabVIEW VI	C/C++ 関数呼び出し
	以下のパラメータを使用して、「niDCPower_ConfigureVoltageLevelRange」を呼び出します。 **vi**: 「niDCPower_init」からの計測器ハンドル **channelName**: *x* **voltageLevelRange**: 表 8 に記載したチャンネル *x* の反復に対応する電圧レベル範囲値

11. niDCPower プロパティノードを使用して「平均化するサンプル数」を指定します。

LabVIEW VI	C/C++ 関数呼び出し
	以下のパラメータを使用して、「niDCPower_SetAttributeViInt32」を呼び出します。 **vi**: 「niDCPower_init」からの計測器ハンドル **channelName**: *x* **attribute**: NIDCPOWER_ATTR_SAMPLES_TO_AVERAGE **value**: 300

12. 「niDCPower 出力有効化を構成」VI を使用して、出力を有効にします。

LabVIEW VI	C/C++ 関数呼び出し
	以下のパラメータを使用して、「niDCPower_ConfigureOutputEnabled」を呼び出します。 **vi**: 「niDCPower_init」からの計測器ハンドル **channelName**: *x* **enabled**: VI_TRUE

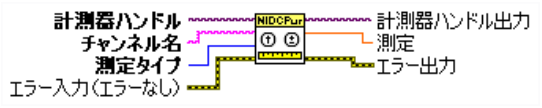
13. 「niDCPower 開始」VI を使用して構成を適用します。

LabVIEW VI	C/C++ 関数呼び出し
	以下のパラメータを使用して、「niDCPower_Initiate」を呼び出します。 **vi**: 「niDCPower_init」からの計測器ハンドル

14. NI PXI-4130 の出力が整定するまで 3 秒間待機します。
15. チャンネル *x* の反復に対応する出力値が 0 の場合、DMM またはキャリブレータの接続が NI PXI-4130 から切断されていることを確認し、手順 16 に進みます。

    出力値がその他の値の場合、DMM を使用して出力電圧または電流を測定します。

16. 測定値を記録します。チャンネル *x* の反復に対応する電圧レベル値が 0 の場合、測定は行いません。測定値の代わりに 0 を記録します。
17. チャンネル 1 の電流範囲が 200 µA、2 mA、20 mA、200 mA の場合に出力電流を求めるには、手順 16 で記録した電圧測定値を手順 4 で記録した抵抗測定値で除算します。チャンネル 0 の電流範囲が 1 A、あるいはチャンネル 1 の電流範囲が 2 A の場合、出力電流は前手順で DMM から直接測定します。
18. 「niDCPower 測定」VI を使用して出力電流を測定します。

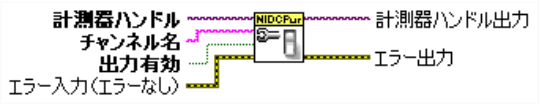
LabVIEW VI	C/C++ 関数呼び出し
	以下のパラメータを使用して、「`niDCPower_Measure`」を呼び出します。 **vi**: 「`niDCPower_init`」からの計測器ハンドル **channelName**: *x* **measurementType**: `NIDCPOWER_VAL_MEASURE_CURRENT`

19. 測定値を記録します。
20. 測定誤差を計算するには、手順 19 で記録した測定値と手順 17 で求めた測定値の差を求めます。
21. 表 8 の「テスト制限値」列に記載されたゲイン / オフセットを使用して、チャンネル *x* の反復に対応するテスト制限値の上下限を計算します。DMM は固有値を測定するため、絶対制限値の代わりに許容範囲が記載されています。各制限値は、DMM の測定値のパーセントとオフセット電圧の和です。手順 17 で求めた測定値を使用して、テスト制限値を計算します。測定誤差が計算した制限値の範囲内に入ることを確認します。測定誤差がテスト制限値の範囲外である場合、NI PXI-4130 を調整する必要があります。
22. 表 8 の電流範囲に対応するチャンネル *x* のすべての反復に対して手順 2 ～ 21 を繰り返します。

23. 「niDCPower 電圧レベルを構成」VI を使用して、電圧レベルを 0 V に設定します。

LabVIEW VI	C/C++ 関数呼び出し
	以下のパラメータを使用して、「niDCPower_ConfigureVoltageLevel」を呼び出します。 **vi**: 「niDCPower_init」からの計測器ハンドル **channelName**: *x* **level**: 0

24. 「niDCPower 出力有効化を構成」VI を使用して、出力を無効にします。

LabVIEW VI	C/C++ 関数呼び出し
	以下のパラメータを使用して、「niDCPower_ConfigureOutputEnabled」を呼び出します。 **vi**: 「niDCPower_init」からの計測器ハンドル **channelName**: *x* **enabled**: VI_FALSE

25. DMM とキャリブレータの接続を切断します。
26. 表 8 に示したすべての未検証のチャンネルに対し、手順 3 ～ 25 を繰り返します。各チャンネルと範囲においてすべての反復を検証すれば、この部分の検証は完了です。
27. 「niDCPower 閉じる」VI を使用してセッションを終了します。

LabVIEW VI	C/C++ 関数呼び出し
	以下のパラメータを使用して、「niDCPower_close」を呼び出します。 **vi**: 「niDCPower_init」からの計測器ハンドル

表 8 NI PXI-4130 出力パラメータとテスト制限値（電流測定確度の検証時）

チャンネル	電流制限範囲	電圧レベル範囲（V）	電圧レベル（V）	反復	出力	テスト制限値 ±（読み取り値の % + オフセット）
0	1 A	6	0	1	350 mA	0.15% + 4 mA[1]
			6	2	350 mA	
				3	700 mA	
1	200 µA	20	0	1	50.0 µA	0.03% + 0.02 µA
			20	2	50.0 µA	
				3	0.10 mA	
				4	0.150 mA	
				5	0.20 mA	
			−20	6	50.0 µA	
				7	0.10 mA	
				8	0.15 mA	
				9	0.20 mA	
	2 mA		0	1	0.50 mA	0.03% + 0.2 µA
			20	2	0.50 mA	
				3	1.00 mA	
				4	1.50 mA	
				5	2.00 mA	
			−20	6	0.50 mA	
				7	1.00 mA	
				8	1.50 mA	
				9	2.00 mA	
	20 mA		0	1	5.00 mA	0.03% + 2.0 µA
			20	2	5.00 mA	
				3	10.0 mA	
				4	15.0 mA	
				5	20.0 mA	
			−20	6	5.00 mA	

表 8 NI PXI-4130 出力パラメータとテスト制限値（電流測定確度の検証時）（続き）

チャンネル	電流制限範囲	電圧レベル範囲（V）	電圧レベル（V）	反復	出力	テスト制限値 ±（読み取り値の % + オフセット）
1	20 mA	20 V	−20	7	10.0 mA	0.03% + 2.0 µA
				8	15.0 mA	
				9	20.0 mA	
	200 mA		0	1	50.0 mA	0.03% + 40 µA
			20	2	50.0 mA	
				3	100 mA	
				4	150 mA	
				5	200 mA	
			−20	6	50.0 mA	
				7	100 mA	
				8	150 mA	
				9	200 mA	
	2 A		0	1	0.5 A	0.12% + 200 µA[1]
			20	2	0.5 A	
				3	1.0 A	
				4	1.5 A	
				5	2.0 A	
			−20 V	6	0.5 A	
				7	1.0 A	
				8	1.5 A	
				9	2.0 A	

[1] 500 mA 以上の電流については、図 4 の「確度低下および負荷電流」を参照してください。

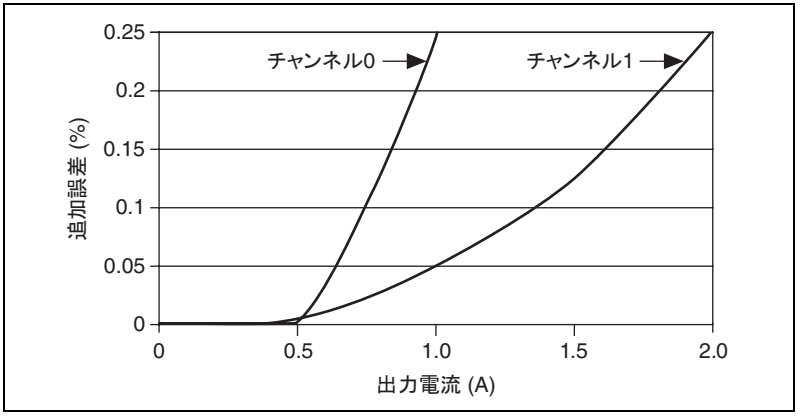


図 4 確度低下および負荷電流

調整

調整により、NI PXI-4130 の確度が向上し、EEPROM 内のキャリブレーション日付および温度情報が更新されます。1 年に 1 回または NI PXI-4130 の確度がキャリブレーションテスト制限値の範囲外となった時には調整を行ってください。

調整により、以下の NI PXI-4130 仕様が補正されます。

- 電圧プログラミング確度
- 電圧測定確度
- 電流プログラミング確度
- 電流測定確度

メモ 以下の手順において、LabVIEW の入力値については「C/C++ 関数呼び出し」に記載された値を参照してください。

メモ NI PXI-4130 が最初の検証テストに合格し、すべてのテスト制限値の範囲内である場合も、NI は公開仕様を保証するために調整を行うことを推奨しています（ただし、必須ではありません）。調整手順を行わない場合、デバイスを調整せずにキャリブレーション日付およびオンボードキャリブレーション温度を更新するには、「niDCPower 外部キャリブレーションを初期化」VI を実行し、**action** を **Commit** に設定して「niDCPower 外部キャリブレーションを閉じる」VI で終了してください。

調整後、「検証」セクションを繰り返して調整が成功したことを確認します。

電圧プログラミング確度を調整する

以下の手順を実行して、NI PXI-4130 の電圧プログラミング確度を調整します。表 9 の各反復に対してこのテストを実行します。

1. 「niDCPower 外部キャリブレーションを初期化」VI によりセッションを開き、セッションハンドルを取得します。

LabVIEW VI	C/C++ 関数呼び出し
	以下のパラメータを使用して、「niDCPower_InitExtCal」を呼び出します。 **resourceName**: MAX で割り当てられたデバイス名 **password**: NI

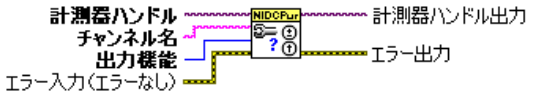
2. 図 1 のように、DMM を NI PXI-4130 のチャンネル *x* 出力端子に接続します。

メモ チャンネル *x* は、テスト中のチャンネルを表します。プログラムの変数 *x* を実際のチャンネル名と置き換えてください。

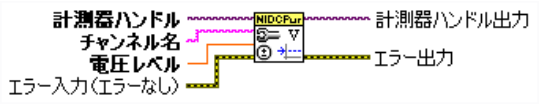
3. 表 3 に記載したチャンネルとレンジに対応するように DMM のモードとレンジを構成します。
4. 「niDCPower 中止」VI を使用して、NI PXI-4130 を遅延構成モードにします。

LabVIEW VI	C/C++ 関数呼び出し
	以下のパラメータを使用して、「niDCPower_Abort」を呼び出します。 **vi**: 「niDCPower_InitExtCal」からの計測器ハンドル

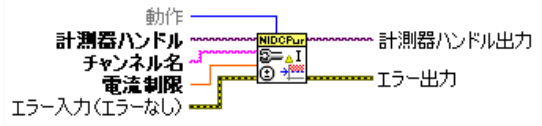
5. 「niDCPower 出力機能を構成」VI を使用して出力機能を構成します。

LabVIEW VI	C/C++ 関数呼び出し
	以下のパラメータを使用して、「niDCPower_ConfigureOutputFunction」を呼び出します。 **vi**: 「niDCPower_InitExtCal」からの計測器ハンドル **channelName**: *x* **outputFunction**: NIDCPOWER_VAL_DC_VOLTAGE

6. 「niDCPower 電圧レベルを構成」VI を使用して電圧レベルを構成します。

LabVIEW VI	C/C++ 関数呼び出し
	以下のパラメータを使用して、「niDCPower_ConfigureVoltageLevel」を呼び出します。 **vi**: 「niDCPower_InitExtCal」からの計測器ハンドル **channelName**: *x* **level**: 表 9 に記載したチャンネル *x* の反復に対応する出力値

7. 「niDCPower 電流制限を構成」VI を使用して、電流制限を構成します。

LabVIEW VI	C/C++ 関数呼び出し
	以下のパラメータを使用して、「niDCPower_ConfigureCurrentLimit」を呼び出します。 **vi**: 「niDCPower_InitExtCal」からの計測器ハンドル **channelName**: *x* **behavior**: NIDCPOWER_VAL_CURRENT_REGULATE **limit**: 0.5

8. 「niDCPower 電圧レベル範囲を構成」VI を使用して、電圧レベル範囲を構成します。

LabVIEW VI	C/C++ 関数呼び出し
	以下のパラメータを使用して、「niDCPower_ConfigureVoltageLevelRange」を呼び出します。 **vi**: 「niDCPower_InitExtCal」からの計測器ハンドル **channelName**: *x* **voltageLevelRange**: 表 9 に記載したチャンネル *x* の反復に対応する電圧レベル範囲値

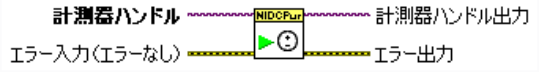
9. 「niDCPower 電流制限範囲を構成」VI を使用して、電流制限範囲を構成します。

LabVIEW VI	C/C++ 関数呼び出し
	以下のパラメータを使用して、「niDCPower_ConfigureCurrentLimitRange」を呼び出します。 **vi**: 「niDCPower_InitExtCal」からの計測器ハンドル **channelName**: *x* **currentLimitRange**: 1

10. 「niDCPower 出力有効化を構成」VI を使用して、出力を有効にします。

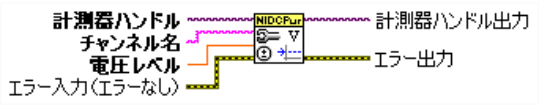
LabVIEW VI	C/C++ 関数呼び出し
	以下のパラメータを使用して、「niDCPower_ConfigureOutputEnabled」を呼び出します。 **vi**: 「niDCPower_InitExtCal」からの計測器ハンドル **channelName**: *x* **enabled**: VI_TRUE

11. 「niDCPower 開始」VI を使用して構成を適用します。

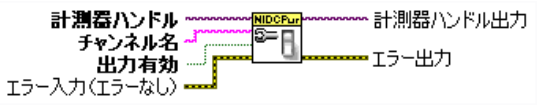
LabVIEW VI	C/C++ 関数呼び出し
	以下のパラメータを使用して、「niDCPower_Initiate」を呼び出します。 **vi**: 「niDCPower_InitExtCal」からの計測器ハンドル

12. NI PXI-4130 の出力が整定するまで 3 秒間待機します。
13. DMM を使用して出力電圧を測定します。
14. 測定値を記録します。
15. 表 9 に示したチャンネル *x* の各反復に対して手順 3～14 を繰り返します。

16. 「niDCPower 電圧レベルを構成」VI を使用して、電圧レベルを 0 V に設定します。

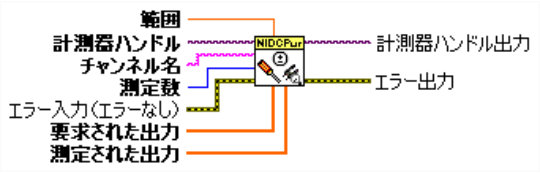
LabVIEW VI	C/C++ 関数呼び出し
	以下のパラメータを使用して、「niDCPower_ConfigureVoltageLevel」を呼び出します。 **vi**: 「niDCPower_InitExtCal」からの計測器ハンドル **channelName**: *x* **level**: 0

17. 「niDCPower 出力有効化を構成」VI を使用して、出力を無効にします。

LabVIEW VI	C/C++ 関数呼び出し
	以下のパラメータを使用して、「niDCPower_ConfigureOutputEnabled」を呼び出します。 **vi**: 「niDCPower_InitExtCal」からの計測器ハンドル **channelName**: *x* **enabled**: VI_FALSE

18. 「niDCPower Cal 電圧レベルを調整」VI を使用して、チャンネル *x* の各電圧レベルを調整します。

メモ チャンネル 1 でキャリブレーションを行う際には、各電圧レベル範囲毎に「niDCPower Cal 電圧レベルを調整」VI を呼び出す必要があります。また、同じ範囲内の正および負の出力値のキャリブレーションを行う際にも、正 / 負それぞれに対してこの VI を呼び出す必要があります。

LabVIEW VI	C/C++ 関数呼び出し
	以下のパラメータを使用して、「`niDCPower_CalAdjustVoltageLevel`」を呼び出します。 **vi**: 「`niDCPower_InitExtCal`」からの計測器ハンドル **channelName**: *x* **range**: チャンネル *x* の電圧レベル範囲 **numberOfMeasurements**: 測定総数の整数値。この値は **requestedOutputs** および **measuredOutputs** 配列内の要素数と一致する必要があり。 **requestedOutputs**: キャリブレーション範囲において、チャンネル *x* の各反復（表 9 参照）に対応する出力値の配列 **measuredOutputs**: キャリブレーション範囲において、手順 14 で記録した、チャンネル *x* の各反復（表 9 参照）に対する測定値の配列

19. DMM との接続を切断します。
20. 表 9 に示したすべての未調整チャンネルに対し、手順 2～19 を繰り返します。すべてのチャンネルで電圧測定値を調整すれば、この部分の調整は完了です。

21. 「niDCPower 外部キャリブレーションを閉じる」VI を使用して、セッションを終了します。

LabVIEW VI	C/C++ 関数呼び出し
	以下のパラメータを使用して、「niDCPower_CloseExtCal」を呼び出します。 **vi**: 「niDCPower_InitExtCal」からの計測器ハンドル **action**: NIDCPOWER_VAL_COMMIT

表 9　NI PXI-4130 出力パラメータ（電圧プログラミング確度調整時）

チャンネル	反復	電圧レベル範囲（V）	出力
0	1	6	0 V
	2		3 V
	3		6 V
1	1	6	1 mV
	2		3 V
	3		6 V
	1		−0.1 mV
	2		−3 V
	3		−6 V
	1	20	1 mV
	2		10 V
	3		20 V
	1		−1 mV
	2		−10 V
	3		−20 V

電圧測定確度を調整する

以下の手順を実行して、NI PXI-4130 の電圧測定確度を調整します。表 10 の各反復に対してこのテストを実行します。

1. 「niDCPower 外部キャリブレーションを初期化」VI によりセッションを開き、セッションハンドルを取得します。

LabVIEW VI	C/C++ 関数呼び出し
	以下のパラメータを使用して、「niDCPower_InitExtCal」を呼び出します。 **resourceName**: MAX で割り当てられたデバイス名 **password**: NI

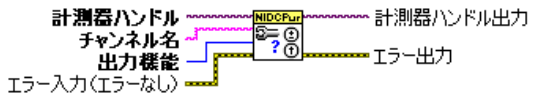
2. 図 1 に示すように、DMM を NI PXI-4130 のチャンネル *x* 出力端子に接続します。

メモ チャンネル *x* は、テスト中のチャンネルを表します。プログラムの変数 *x* を実際のチャンネル名と置き換えてください。

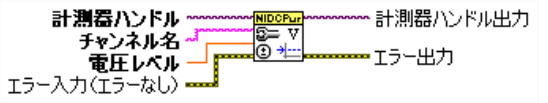
3. 表 3 に記載したチャンネルとレンジに対応するように DMM のモードとレンジを構成します。
4. 「niDCPower 中止」VI を使用して、NI PXI-4130 を遅延構成モードにします。

LabVIEW VI	C/C++ 関数呼び出し
	以下のパラメータを使用して、「niDCPower_Abort」を呼び出します。 **vi**: 「niDCPower_InitExtCal」からの計測器ハンドル

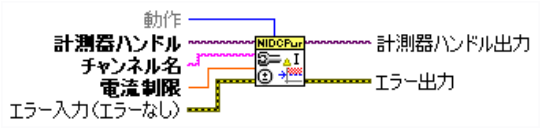
5. 「niDCPower 出力機能を構成」VI を使用して、出力機能を「DC 電圧」に設定します。

LabVIEW VI	C/C++ 関数呼び出し
	以下のパラメータを使用して、「niDCPower_ConfigureOutputFunction」を呼び出します。 **vi**: 「niDCPower_InitExtCal」からの計測器ハンドル **channelName**: *x* **outputFunction**: NIDCPOWER_VAL_DC_VOLTAGE

6. 「niDCPower 電圧レベルを構成」VI を使用して電圧レベルを構成します。

LabVIEW VI	C/C++ 関数呼び出し
	以下のパラメータを使用して、「niDCPower_ConfigureVoltageLevel」を呼び出します。 **vi**: 「niDCPower_InitExtCal」からの計測器ハンドル **channelName**: *x* **level**: 表 10 に記載したチャンネル *x* の反復に対応する出力値

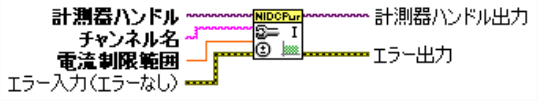
7. 「niDCPower 電流制限を構成」VI を使用して、電流制限を構成します。

LabVIEW VI	C/C++ 関数呼び出し
	以下のパラメータを使用して、「niDCPower_ConfigureCurrentLimit」を呼び出します。 **vi**: 「niDCPower_InitExtCal」からの計測器ハンドル **channelName**: *x* **behavior**: NIDCPOWER_VAL_CURRENT_REGULATE **limit**: 0.5

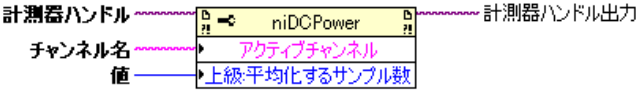
8. 「niDCPower 電圧レベル範囲を構成」VI を使用して、電圧レベル範囲を構成します。

LabVIEW VI	C/C++ 関数呼び出し
	以下のパラメータを使用して、「niDCPower_ConfigureVoltageLevelRange」を呼び出します。 **vi**: 「niDCPower_InitExtCal」からの計測器ハンドル **channelName**: *x* **voltageLevelRange**: 表 10 に記載したチャンネル *x* の反復に対応する電圧レベル範囲値

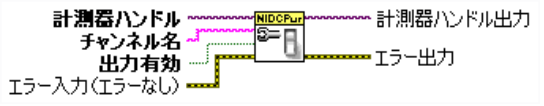
9. 「niDCPower 電流制限範囲を構成」VI を使用して、電流制限範囲を構成します。

LabVIEW VI	C/C++ 関数呼び出し
	以下のパラメータを使用して、「niDCPower_ConfigureCurrentLimitRange」を呼び出します。 **vi**: 「niDCPower_InitExtCal」からの計測器ハンドル **channelName**: *x* **currentLimitRange**: 1

10. niDCPower プロパティノードを使用して「平均化するサンプル数」を指定します。

LabVIEW VI	C/C++ 関数呼び出し
	以下のパラメータを使用して、「niDCPower_SetAttributeViInt32」を呼び出します。 **vi**: 「niDCPower_InitExtCal」からの計測器ハンドル **channelName**: *x* **attribute**: NIDCPOWER_ATTR_SAMPLES_TO_AVERAGE **value**: 300

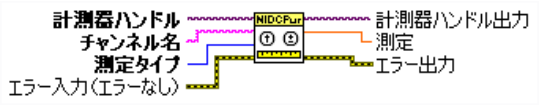
11. 「niDCPower 出力有効化を構成」VI を使用して、出力を有効にします。

LabVIEW VI	C/C++ 関数呼び出し
	以下のパラメータを使用して、「niDCPower_ConfigureOutputEnabled」を呼び出します。 **vi**: 「niDCPower_InitExtCal」からの計測器ハンドル **channelName**: *x* **enabled**: VI_TRUE

12. 「niDCPower 開始」VI を使用して構成を適用します。

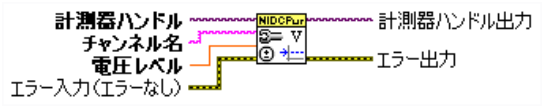
LabVIEW VI	C/C++ 関数呼び出し
	以下のパラメータを使用して、「niDCPower_Initiate」を呼び出します。 **vi**: 「niDCPower_InitExtCal」からの計測器ハンドル

13. NI PXI-4130 の出力が整定するまで 3 秒間待機します。
14. DMM を使用して出力電圧を測定します。
15. 測定値を記録します。
16. 「niDCPower 測定」VI を使用して出力電圧を測定します。

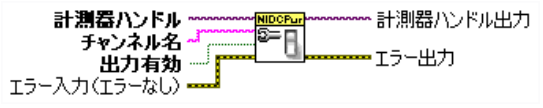
LabVIEW VI	C/C++ 関数呼び出し
	以下のパラメータを使用して、「niDCPower_Measure」を呼び出します。 **vi**: 「niDCPower_InitExtCal」からの計測器ハンドル **channelName**: *x* **measurementType**: NIDCPOWER_VAL_MEASURE_VOLTAGE

17. 測定値を記録します。
18. 表 10 に示したチャンネル *x* の各反復に対して手順 3～17 を繰り返します。

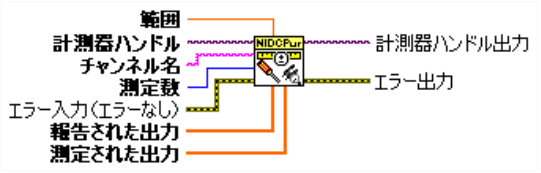
19. 「niDCPower 電圧レベルを構成」VI を使用して、電圧レベルを 0 V に設定します。

LabVIEW VI	C/C++ 関数呼び出し
	以下のパラメータを使用して、「niDCPower_ConfigureVoltageLevel」を呼び出します。 **vi**: 「niDCPower_InitExtCal」からの計測器ハンドル **channelName**: *x* **level**: 0

20. 「niDCPower 出力有効化を構成」VI を使用して、出力を無効にします。

LabVIEW VI	C/C++ 関数呼び出し
	以下のパラメータを使用して、「niDCPower_ConfigureOutputEnabled」を呼び出します。 **vi**: 「niDCPower_InitExtCal」からの計測器ハンドル **channelName**: *x* **enabled**: VI_FALSE

21. 「niDCPower Cal 電圧測定を調整」VI を使用して電圧測定を調整します。

LabVIEW VI	C/C++ 関数呼び出し
	以下のパラメータを使用して、「niDCPower_CalAdjustVoltage Measurement」を呼び出します。 **vi**: 「niDCPower_InitExtCal」からの計測器ハンドル **channelName**: *x* **range**: チャンネル *x* の電圧範囲 **numberOfMeasurements**: 3 **reportedOutputs**: 手順 17 で記録した、チャンネル *x* の各反復に対する NI PXI-4130 の測定値の配列 **measuredOutputs**: 手順 15 で記録した、チャンネル *x* の各反復に対する DMM の測定値の配列

22. DMM との接続を切断します。
23. 表 10 に示したすべての未調整チャンネルに対し、手順 2 ～ 22 を繰り返します。すべてのチャンネルで電圧測定値を調整すれば、この部分の調整は完了です。
24. 「niDCPower 外部キャリブレーションを閉じる」VI を使用して、セッションを終了します。

LabVIEW VI	C/C++ 関数呼び出し
	以下のパラメータを使用して、「niDCPower_CloseExtCal」を呼び出します。 **vi**: 「niDCPower_InitExtCal」からの計測器ハンドル **action**: NIDCPOWER_VAL_COMMIT

表 10 NI PXI-4130 出力パラメータ（電圧測定確度の調整時）

チャンネル	反復	電圧レベル範囲（V）	出力（V）
0	1	6	0
	2		3
	3		6
1	1	20	−20
	2		0
	3		20

電流プログラミング確度を調整する

以下の手順を実行して、NI PXI-4130 の電流プログラミング確度を調整します。表 11 の各チャンネルの反復に対してこのテストを実行します。

1. 「niDCPower 外部キャリブレーションを初期化」VI によりセッションを開き、セッションハンドルを取得します。

LabVIEW VI	C/C++ 関数呼び出し
	以下のパラメータを使用して、「`niDCPower_InitExtCal`」を呼び出します。 **resourceName**: MAX で割り当てられたデバイス名 **password**: `NI`

2. 図 2 または図 3 に示すように、NI PXI-4130 のチャンネル *x* を DMM または Fluke 5700A/5720A キャリブレータに接続します。

   図 2 は、チャンネル 1 の電流範囲が 200 µA、2 mA、20 mA、200 mA の場合の設定を示し、図 3 はチャンネル 0 の電流範囲が 1 A、あるいはチャンネル 1 の電流範囲が 2 A の場合の設定を示しています。

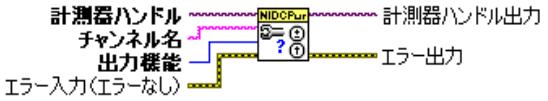
メモ チャンネル *x* は、テスト中のチャンネルを表します。プログラムの変数 *x* を実際のチャンネル名と置き換えてください。

3. 表 4 に記載したチャンネルとレンジに対応するように DMM のモードとレンジを構成します。

4. 表 4 に記載したチャンネルとレンジに対する抵抗値を Fluke 5700/5720A キャリブレータに設定します。キャリブレータの外部センス（4 線式モード）を有効にします。キャリブレータに表示された抵抗値を記録します。
5. 「niDCPower 中止」VI を使用して、NI PXI-4130 を遅延構成モードにします。

LabVIEW VI	C/C++ 関数呼び出し
	以下のパラメータを使用して、「niDCPower_Abort」を呼び出します。 **vi**: 「niDCPower_InitExtCal」からの計測器ハンドル

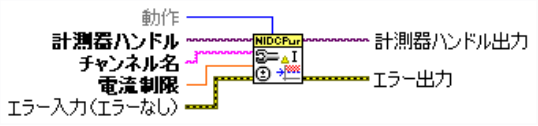
6. 「niDCPower 出力機能を構成」VI を使用して、出力機能を「DC 電圧」に設定します。

LabVIEW VI	C/C++ 関数呼び出し
	以下のパラメータを使用して、「niDCPower_ConfigureOutputFunction」を呼び出します。 **vi**: 「niDCPower_InitExtCal」からの計測器ハンドル **channelName**: *x* **outputFunction**: NIDCPOWER_VAL_DC_VOLTAGE

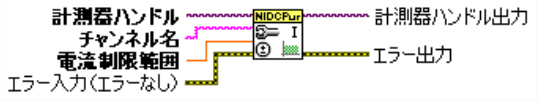
7. 「niDCPower 電圧レベルを構成」VI を使用して電圧レベルを構成します。

LabVIEW VI	C/C++ 関数呼び出し
	以下のパラメータを使用して、「niDCPower_ConfigureVoltageLevel」を呼び出します。 **vi**: 「niDCPower_InitExtCal」からの計測器ハンドル **channelName**: *x* **voltageLevel**: 表 11 に記載したチャンネル *x* の反復に対応する電圧レベル値

8. 「niDCPower 電流制限を構成」VI を使用して、電流制限を構成します。

LabVIEW VI	C/C++ 関数呼び出し
	以下のパラメータを使用して、「niDCPower_ConfigureCurrentLimit」を呼び出します。 **vi**: 「niDCPower_InitExtCal」からの計測器ハンドル **channelName**: *x* **limit**: 表 11 に記載したチャンネル *x* の反復に対応する出力値

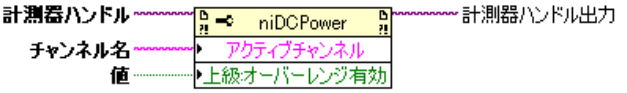
9. 「niDCPower 電流制限範囲を構成」VI を使用して、電流制限範囲を構成します。

LabVIEW VI	C/C++ 関数呼び出し
	以下のパラメータを使用して、「niDCPower_ConfigureCurrentLimitRange」を呼び出します。 **vi**: 「niDCPower_InitExtCal」からの計測器ハンドル **channelName**: *x* **currentLimitRange**: 表 11 に記載したチャンネル *x* の反復に対応する電流制限範囲値

10. 「niDCPower 電圧レベル範囲を構成」VI を使用して、電圧レベル範囲を構成します。

LabVIEW VI	C/C++ 関数呼び出し
	以下のパラメータを使用して、「niDCPower_ConfigureVoltageLevelRange」を呼び出します。 **vi**: 「niDCPower_InitExtCal」からの計測器ハンドル **channelName**: *x* **voltageLevelRange**: 表 11 に記載したチャンネル *x* の反復に対応する電圧レベル範囲値

11. niDCPower プロパティノードを使用して、電流オーバーレンジを有効にします。

LabVIEW VI	C/C++ 関数呼び出し
	以下のパラメータを使用して、「niDCPower_SetAttributeViInt32」を呼び出します。 **vi**: 「niDCPower_InitExtCal」からの計測器ハンドル **channelName**: *x* **attribute**: NIDCPOWER_ATTR_OVERRANGING_ENABLED **value**: VI_TRUE

12. 「niDCPower 出力有効化を構成」VI を使用して、出力を有効にします。

LabVIEW VI	C/C++ 関数呼び出し
	以下のパラメータを使用して、「niDCPower_ConfigureOutputEnabled」を呼び出します。 **vi**: 「niDCPower_InitExtCal」からの計測器ハンドル **channelName**: *x* **enabled**: VI_TRUE

13. 「niDCPower 開始」VI を使用して構成を適用します。

LabVIEW VI	C/C++ 関数呼び出し
	以下のパラメータを使用して、「niDCPower_Initiate」を呼び出します。 **vi**: 「niDCPower_InitExtCal」からの計測器ハンドル

14. NI PXI-4130 の出力が整定するまで 3 秒間待機します。
15. DMM を使用して出力電圧または電流を測定します。
16. 測定値を記録します。

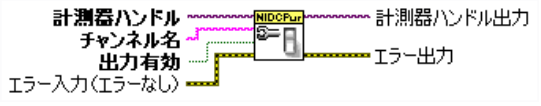
17. チャンネル 1 の電流範囲が 200 µA、2 mA、20 mA、200 mA の場合に出力電流を求めるには、手順 16 で記録した電圧測定値を手順 4 で記録した抵抗測定値で除算します。チャンネル 0 の電流範囲が 1 A、あるいはチャンネル 1 の電流範囲が 2 A の場合、出力電流は DMM から直接測定します。
18. 表 11 に示した、チャンネル *x* の各電流範囲におけるすべての反復に対して手順 2 ～17 を繰り返します。

メモ チャンネル 1 の場合は、各電流制限範囲で正の電圧レベルと負の電圧レベルに対する反復があります。

19. 「niDCPower 電圧レベルを構成」VI を使用して、電圧レベルを 0 V に設定します。

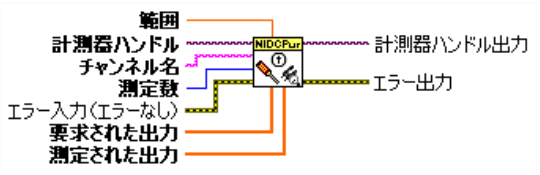
LabVIEW VI	C/C++ 関数呼び出し
	以下のパラメータを使用して、「niDCPower_ConfigureVoltageLevel」を呼び出します。 **vi**: 「niDCPower_InitExtCal」からの計測器ハンドル **channelName**: *x* **voltageLevel**: 0

20. 「niDCPower 出力有効化を構成」VI を使用して、出力を無効にします。

LabVIEW VI	C/C++ 関数呼び出し
	以下のパラメータを使用して、「niDCPower_ConfigureOutputEnabled」を呼び出します。 **vi**: 「niDCPower_InitExtCal」からの計測器ハンドル **channelName**: *x* **enabled**: VI_FALSE

21. 「niDCPower Cal 電流制限を調整」VI を使用して、表 11 に記載した各電流制限範囲を調整します。

メモ チャンネル 1 でキャリブレーションを行う際には、各電流レベル毎に「niDCPower Cal 電流レベルを調整」VI を呼び出す必要があります。また、同じ範囲内の正および負の出力値のキャリブレーションを行う際にも、正 / 負それぞれに対してこの VI を呼び出す必要があります。

LabVIEW VI	C/C++ 関数呼び出し
	以下のパラメータを使用して、「`niDCPower_CalAdjustCurrentLimit`」を呼び出します。 **vi**: 「`niDCPower_InitExtCal`」からの計測器ハンドル **channelName**: *x* **range**: チャンネル *x* の電流制限範囲 **numberOfMeasurements**: 測定総数の整数値。この値は **requestedOutputs** および **measuredOutputs** 配列内の要素数と一致する必要があり。 **requestedOutputs**: キャリブレーション範囲において、チャンネル *x* の各反復（表 11 参照）に対応する出力値の配列 **measuredOutputs**: キャリブレーション範囲において、手順 16 で記録した、チャンネル *x* の各反復（表 11 参照）に対する測定値の配列

22. 表 11 に示したチャンネル *x* のすべての電流範囲で 2 ～ 21 を繰り返します。
23. DMM とキャリブレータの接続を切断します。
24. 表 11 に示したすべての未調整チャンネルで手順 2 ～ 23 を繰り返します。すべてのチャンネルで電流プログラミング値を調整すれば、この部分の調整は完了です。

25. 「niDCPower 外部キャリブレーションを閉じる」VI を使用して、セッションを終了します。

LabVIEW VI	C/C++ 関数呼び出し
	以下のパラメータを使用して、「`niDCPower_CloseExtCal`」を呼び出します。 **vi**: 「`niDCPower_InitExtCal`」からの計測器ハンドル **action**: `NIDCPOWER_VAL_COMMIT`

表 11　NI PXI-4130 出力パラメータとテスト制限値（電流プログラミング確度調整時）

チャンネル	電流制限範囲	電圧レベル範囲	電圧レベル	反復	出力
0	1 A	6 V	6 V	1	20 mA
				2	350 mA
				3	700 mA
1	200 µA	20 V	20 V	1	2.00 µA
				2	24.0 µA
				3	46.0 µA
				4	68.0 µA
				5	90.0 µA
				6	112 µA
				7	134 µA
				8	156 µA
				9	178 µA
				10	200 µA
			−20 V	11	2.00 µA
				12	24.0 µA
				13	46.0 µA
				14	68.0 µA
				15	90.0 µA
				16	112 µA
				17	134 µA

表 11 NI PXI-4130 出力パラメータとテスト制限値（電流プログラミング確度調整時）（続き）

チャンネル	電流制限範囲	電圧レベル範囲	電圧レベル	反復	出力
1	200 µA	20 V	−20 V	18	156 µA
				19	178 µA
				20	200 µA
	2 mA		20 V	1	20.0 µA
				2	0.240 mA
				3	0.460 mA
				4	0.680 mA
				5	0.900 mA
				6	1.12 mA
				7	1.34 mA
				8	1.56 mA
				9	1.78 mA
				10	2.00 mA
			−20 V	11	20.0 µA
				12	0.240 mA
				13	0.460 mA
				14	0.680 mA
				15	0.900 mA
				16	1.12 mA
				17	1.34 mA
				18	1.56 mA
				19	1.78 mA
				20	2.00 mA
	20 mA		20 V	1	0.200 mA
				2	2.40 mA
				3	4.60 mA
				4	6.80 mA
				5	9.00 mA
				6	11.2 mA

表 11 NI PXI-4130 出力パラメータとテスト制限値（電流プログラミング確度調整時）（続き）

チャンネル	電流制限範囲	電圧レベル範囲	電圧レベル	反復	出力
1	20 mA	20 V	20 V	7	13.4 mA
				8	15.6 mA
				9	17.8 mA
				10	20.0 mA
			−20 V	11	0.200 mA
				12	2.40 mA
				13	4.60 mA
				14	6.80 mA
				15	9.00 mA
				16	11.2 mA
				17	13.4 mA
				18	15.6 mA
				19	17.8 mA
				20	20.0 mA
	200 mA		20 V	1	0.200 mA
				2	2.40 mA
				3	4.60 mA
				4	6.80 mA
				5	9.00 mA
				6	112 mA
				7	134 mA
				8	156 mA
				9	178 mA
				10	200 mA
			−20 V	11	2.00 mA
				12	24.0 mA
				13	46.0 mA
				14	68.0 mA
				15	90.0 mA
				16	112 mA

表 11 NI PXI-4130 出力パラメータとテスト制限値（電流プログラミング確度調整時）（続き）

チャンネル	電流制限範囲	電圧レベル範囲	電圧レベル	反復	出力
1	200 mA	20 V	−20 V	17	134 mA
				18	156 mA
				19	178 mA
				20	200 mA
	2 A		20 V	1	20.0 mA
				2	240 mA
				3	460 mA
				4	680 mA
				5	900 mA
				6	1.12 A
				7	1.34 A
				8	1.56 A
				9	1.78 A
				10	2.00 A
			−20 V	11	20.0 mA
				12	240 mA
				13	460 mA
				14	680 mA
				15	900 mA
				16	1.12 A
				17	1.34 A
				18	1.56 A
				19	1.78 A
				20	2.00 A

電流測定確度を調整する

以下の手順を実行して、NI PXI-4130 の電流測定確度を調整します。表 12 の各チャンネルの反復に対してこのテストを実行します。

1. 「niDCPower 外部キャリブレーションを初期化」VI によりセッションを開き、セッションハンドルを取得します。

LabVIEW VI	C/C++ 関数呼び出し
	以下のパラメータを使用して、「`niDCPower_InitExtCal`」を呼び出します。 **resourceName**: MAX で割り当てられたデバイス名 **password**: `NI`

2. チャンネル *x* の反復に対応する電圧レベル値が 0 の場合、手順 5 に進みます。DMM または Fluke 5700A/5720A キャリブレータを NI PXI-4130 のチャンネル *x* 出力端子に接続しないでください。

   0 以外の出力値の場合、図 2 または図 3 に示すように、NI PXI-4130 のチャンネル *x* を DMM または Fluke 5700A/5720A キャリブレータに接続します。

   図 2 は、チャンネル 1 の電流範囲が 200 µA、2 mA、20 mA、200 mA の場合の設定を示し、図 3 はチャンネル 0 の電流範囲が 1 A、あるいはチャンネル 1 の電流範囲が 2 A の場合の設定を示しています。

メモ チャンネル *x* は、テスト中のチャンネルを表します。プログラムの変数 *x* を実際のチャンネル名と置き換えてください。

3. 表 4 に記載したチャンネルとレンジに対応するように DMM のモードとレンジを構成します。
4. 表 4 に記載したチャンネルとレンジに対する抵抗値を Fluke 5700/5720A キャリブレータに設定します。キャリブレータの外部センス（4 線式モード）を有効にします。キャリブレータに表示された抵抗値を記録します。

5. 「niDCPower 中止」VI を使用して、NI PXI-4130 を遅延構成モードにします。

LabVIEW VI	C/C++ 関数呼び出し
	以下のパラメータを使用して、「niDCPower_Abort」を呼び出します。 **vi**: 「niDCPower_InitExtCal」からの計測器ハンドル

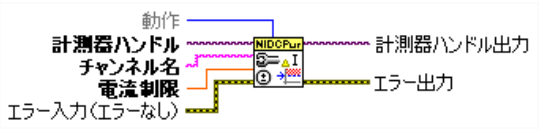
6. 「niDCPower 出力機能を構成」VI を使用して、出力機能を「DC 電圧」に設定します。

LabVIEW VI	C/C++ 関数呼び出し
	以下のパラメータを使用して、「niDCPower_ConfigureOutputFunction」を呼び出します。 **vi**: 「niDCPower_InitExtCal」からの計測器ハンドル **channelName**: *x* **outputFunction**: NIDCPOWER_VAL_DCVOLTAGE

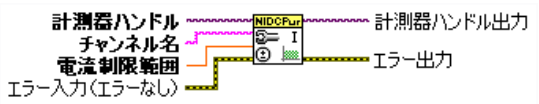
7. 「niDCPower 電圧レベルを構成」VI を使用して電圧レベルを構成します。

LabVIEW VI	C/C++ 関数呼び出し
	以下のパラメータを使用して、「niDCPower_ConfigureVoltageLevel」を呼び出します。 **vi**: 「niDCPower_InitExtCal」からの計測器ハンドル **channelName**: *x* **level**: 表 12 に記載したチャンネル *x* の反復に対応する電圧レベル値

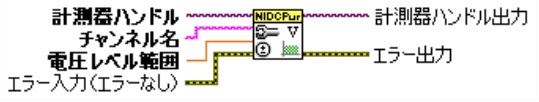
8. 「niDCPower 電流制限を構成」VI を使用して、電流制限を構成します。

LabVIEW VI	C/C++ 関数呼び出し
	以下のパラメータを使用して、「niDCPower_ConfigureCurrentLimit」を呼び出します。 **vi**: 「niDCPower_InitExtCal」からの計測器ハンドル **channelName**: *x* **behavior**: NIDCPOWER_VAL_CURRENT_REGULATE **limit**: 表 12 に記載したチャンネル *x* の反復に対応する出力値

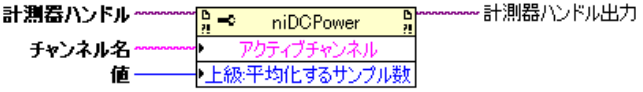
9. 「niDCPower 電流制限範囲を構成」VI を使用して、電流制限範囲を構成します。

LabVIEW VI	C/C++ 関数呼び出し
	以下のパラメータを使用して、「niDCPower_ConfigureCurrentLimitRange」を呼び出します。 **vi**: 「niDCPower_InitExtCal」からの計測器ハンドル **channelName**: *x* **currentLimitRange**: 表 12 に記載したチャンネル *x* の反復に対応する電流制限範囲値

10. 「niDCPower 電圧レベル範囲を構成」VI を使用して、電圧レベル範囲を構成します。

LabVIEW VI	C/C++ 関数呼び出し
	以下のパラメータを使用して、「niDCPower_ConfigureVoltageLevelRange」を呼び出します。 **vi**: 「niDCPower_InitExtCal」からの計測器ハンドル **channelName**: *x* **voltageLevelRange**: 表 12 に記載したチャンネル *x* の反復に対応する電圧レベル範囲値

11. niDCPower プロパティノードを使用して「平均化するサンプル数」を指定します。

LabVIEW VI	C/C++ 関数呼び出し
	以下のパラメータを使用して、「niDCPower_SetAttributeViInt32」を呼び出します。 **vi**: 「niDCPower_InitExtCal」からの計測器ハンドル **channelName**: *x* **attribute**: NIDCPOWER_ATTR_SAMPLES_TO_AVERAGE **value**: 300

12. 「niDCPower 出力有効化を構成」VI を使用して、出力を有効にします。

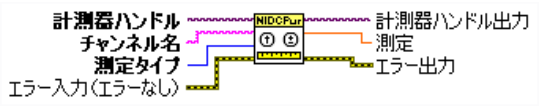
LabVIEW VI	C/C++ 関数呼び出し
	以下のパラメータを使用して、「niDCPower_ConfigureOutputEnabled」を呼び出します。 **vi**: 「niDCPower_InitExtCal」からの計測器ハンドル **channelName**: *x* **enabled**: VI_TRUE

13. 「niDCPower 開始」VI を使用して構成を適用します。

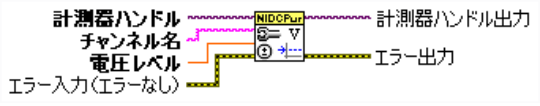
LabVIEW VI	C/C++ 関数呼び出し
	以下のパラメータを使用して、「niDCPower_Initiate」を呼び出します。 **vi**: 「niDCPower_InitExtCal」からの計測器ハンドル

14. NI PXI-4130 の出力が整定するまで 3 秒間待機します。
15. チャンネル *x* の反復に対応する出力値が 0 の場合、手順 16 に進みます。出力値がその他の値の場合、DMM を使用して出力電圧または電流を測定します。
16. 測定値を記録します。チャンネル *x* の反復に対応する電圧レベル値が 0 の場合、測定は行いません。測定値の代わりに 0 を記録します。

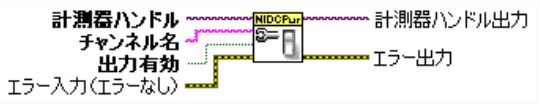
17. チャンネル 1 の電流範囲が 200 µA、2 mA、20 mA、200 mA の場合に出力電流を求めるには、手順 16 で記録した電圧測定値を手順 4 で記録した抵抗測定値で除算します。チャンネル 0 の電流範囲が 1 A、あるいはチャンネル 1 の電流範囲が 2 A の場合、出力電流は手順 15 で DMM から直接測定します。
18. 「niDCPower 測定」VI を使用して出力電流を測定します。

LabVIEW VI	C/C++ 関数呼び出し
	以下のパラメータを使用して、「niDCPower_Measure」を呼び出します。 **vi**: 「niDCPower_InitExtCal」からの計測器ハンドル **channelName**: *x* **measurementType**: NIDCPOWER_VAL_MEASURE_CURRENT

19. 測定値を記録します。
20. 表 12 に示した、チャンネル *x* の各電流範囲におけるすべての反復に対して手順 2～19 を繰り返します。
21. 「niDCPower 電圧レベルを構成」VI を使用して、電圧レベルを 0 V に設定します。

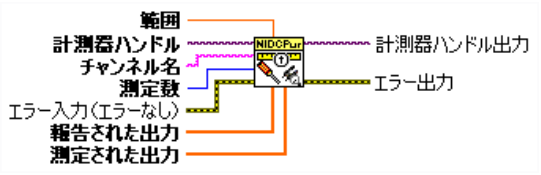
LabVIEW VI	C/C++ 関数呼び出し
	以下のパラメータを使用して、「niDCPower_ConfigureVoltageLevel」を呼び出します。 **vi**: 「niDCPower_InitExtCal」からの計測器ハンドル **channelName**: *x* **voltageLevel**: 0

22. 「niDCPower 出力有効化を構成」VI を使用して、出力を無効にします。

LabVIEW VI	C/C++ 関数呼び出し
	以下のパラメータを使用して、「niDCPower_ConfigureOutputEnabled」を呼び出します。 **vi**: 「niDCPower_InitExtCal」からの計測器ハンドル **channelName**: *x* **enabled**: VI_FALSE

23. 「niDCPower Cal 電流測定を調整」VI を使用して電流測定を調整します。

メモ チャンネル 1 でキャリブレーションを行う際には、各電流制限範囲毎に「niDCPower Cal 電流測定を調整」VI を呼び出す必要があります。

LabVIEW VI	C/C++ 関数呼び出し
	以下のパラメータを使用して、「niDCPower_CalAdjustCurrentMeasurement」を呼び出します。 **vi**: 「niDCPower_InitExtCal」からの計測器ハンドル **channelName**: *x* **range**: チャンネル *x* の電圧範囲 **numberOfMeasurements**: 3 **reportedOutputs**: キャリブレーション範囲において手順 19 で記録した、チャンネル *x* の各反復（表 12 を参照）に対する NI PXI-4130 の測定値の配列 **measuredOutputs**: キャリブレーション範囲において、手順 16 で記録した、チャンネル *x* の各反復（表 12 を参照）に対する DMM の測定値の配列

24. DMM とキャリブレータの接続を切断します。
25. 表 12 に示したすべての未調整チャンネルに対し、手順 2 ～ 24 を繰り返します。すべてのチャンネルで電流測定値を調整すれば、この部分の調整は完了です。
26. 「niDCPower 外部キャリブレーションを閉じる」VI を使用して、セッションを終了します。

LabVIEW VI	C/C++ 関数呼び出し
	以下のパラメータを使用して、「niDCPower_CloseExtCal」を呼び出します。 **vi**: 「niDCPower_InitExtCal」からの計測器ハンドル **action**: NIDCPOWER_VAL_COMMIT

すべての調整テストが成功すれば、NI PXI-4130 の調整が完了します。NI PXI-4130 の調整後の性能を再検証するために、「検証」セクションを繰り返します。NI PXI-4130 がすべての検証テストに合格すれば、キャリブレーションが完了します。

表 12 NI PXI-4130 出力パラメータとテスト制限値（電流測定確度の調整時）

チャンネル	電流制限範囲	電圧レベル範囲	電圧レベル	反復	出力（A）
0	1 A	6 V	0 V	1	3.50E-01
			6 V	2	3.50E-01
				3	7.00E-01
1	200 µA	20 V	0 V	1	2.00E-04
			20 V	2	
			−20 V	3	
	2 mA		0 V	1	2.00E-03
			20 V	2	
			−20 V	3	
	20 mA		0 V	1	2.00E-02
			20 V	2	
			−20 V	3	
	200 mA		0 V	1	2.00E-01
			20 V	2	
			−20 V	3	
	2 A		0 V	1	8.00E-01
			20 V	2	
			−20 V	3	

付録 A: キャリブレーションオプション

キャリブレーション手順では、NI PXI-4130 を検証し、必要に応じてこのデバイスを調整および再検証します。

検証により、デバイスをテストして確度が仕様またはキャリブレーションテスト制限値の範囲内であることを確認します。デバイスの調整が必要かどうかを判断するため、または調整後に手順が成功したかどうかを判断するために検証手順を行います。

メモ このドキュメントでは、キャリブレーションテスト制限値は NI PXI-4130 の公開仕様です。

調整により、デバイスの性能を測定して補正し、測定確度を向上させることができます。調整手順ではキャリブレーション日付情報も更新します。

「完全なキャリブレーション」セクションでは、推奨するキャリブレーション手順について説明します。デバイスがキャリブレーションテスト制限値の範囲内であり、調整手順を省略する場合は、代替のキャリブレーション手順を記載した「オプションのキャリブレーション」セクションを参照してください。

完全なキャリブレーション

NI PXI-4130 がキャリブレーション後の一年間、公開仕様を満たし、仕様要件を上回ることを保証するために、完全なキャリブレーションを行うことを推奨します。調整後に再検証を行い、デバイスがキャリブレーションテスト制限値の範囲内であることを確認します。図 5 は、完全なキャリブレーションのプログラミングフローを示しています。

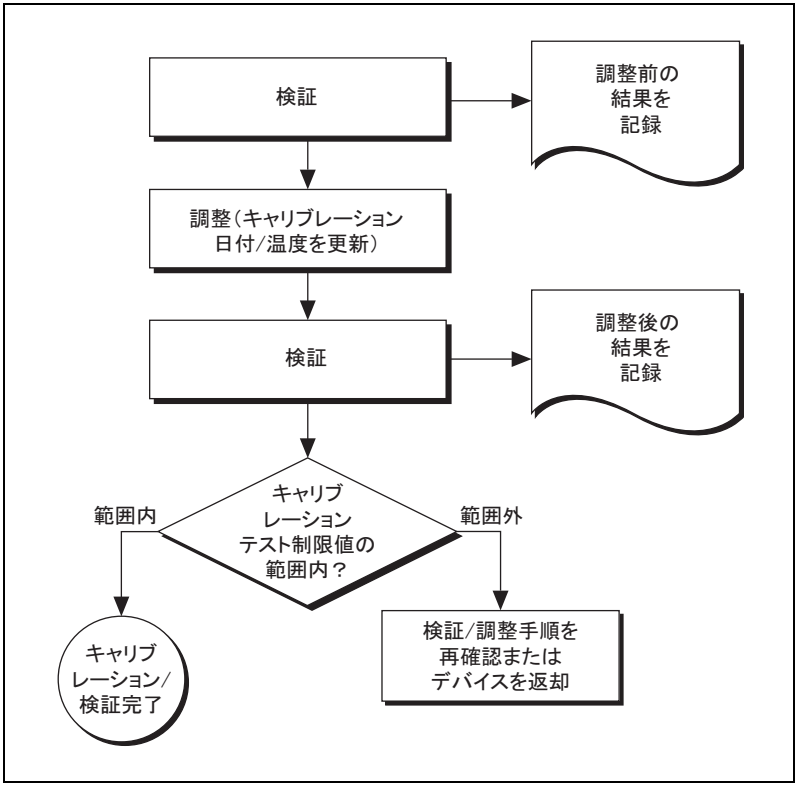


図 5 完全なキャリブレーションのプログラミングフロー

オプションのキャリブレーション

デバイスの確度が最初の検証でキャリブレーションテスト制限値の範囲内であり、NI PXI-4130 が公開仕様を満たしている場合は、キャリブレーション手順の調整を省略することができます。調整手順を省略する場合でも、キャリブレーション日付を更新し、キャリブレーション間隔をリセットすることができます。詳細については、「調整」のセクションを参照してください。

検証せずに調整手順を行う場合も、デバイスの確度が調整後のキャリブレーションテスト制限値の範囲内であることを確認する必要があります。

図 6 は、オプションのキャリブレーションのプログラミングフローを示しています。

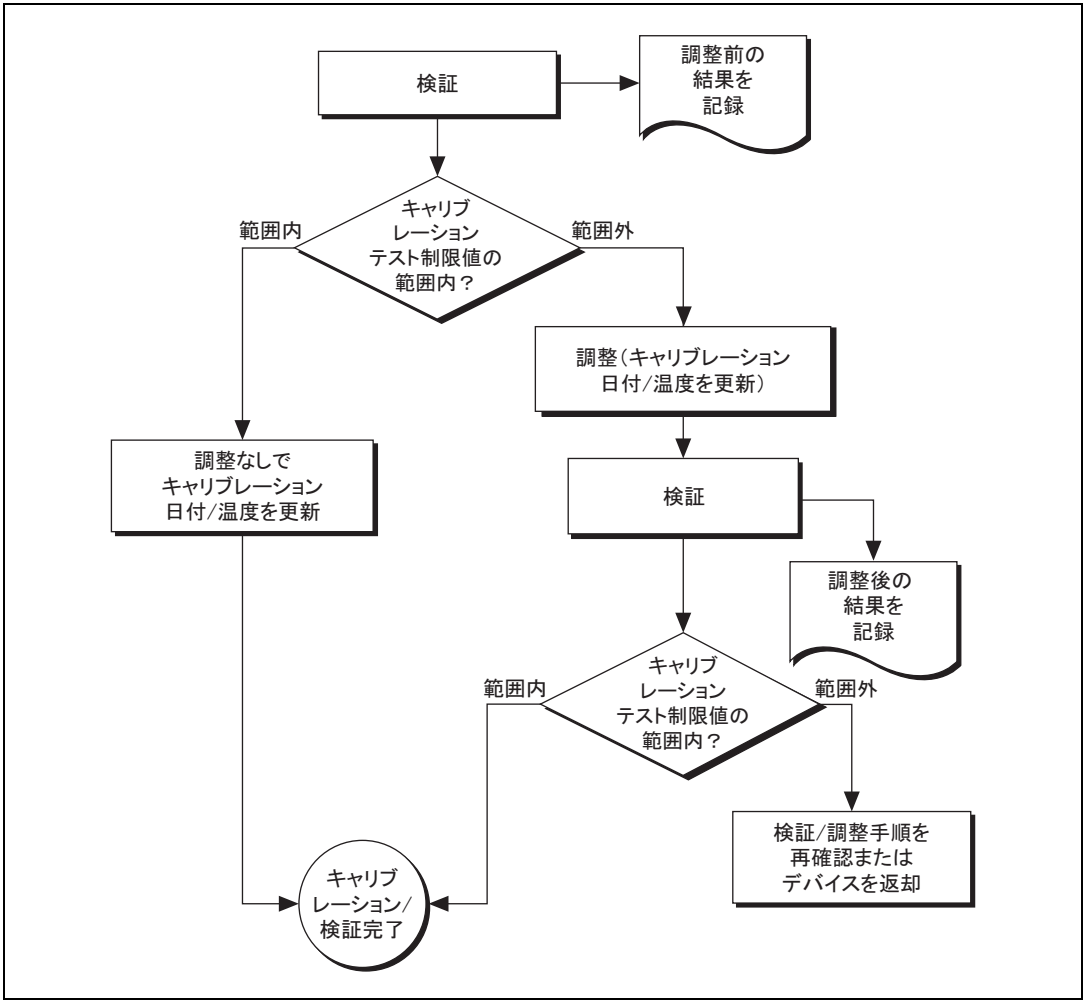


図 6　オプションのキャリブレーションのプログラミングフロー

付録 B: キャリブレーションユーティリティ

NI-DCPower は、キャリブレーションユーティリティ関数および VI のセットを提供します。ユーティリティ関数と VI を使用して、NI PXI-4130 で実行した調整情報の取り出し、キャリブレーションパスワードの変更、オンボード EEPROM への少量の情報の保存が可能です。これらの関数/VI リファレンスについては、『NI DC 電源および SMU ヘルプ』を参照してください。ユーティリティ関数には、以下の関数があります。

- 「niDCPower 外部キャリブレーションパスワードを変更」VI (`niDCPower_ChangeExtCalPassword`)
- 「niDCPower 外部キャリブレーション推奨間隔を取得」VI (`niDCPower_GetExtCalRecommendedInterval`)
- 「niDCPower 最後の外部キャリブレーション日時を取得」VI (`niDCPower_GetExtCalLastDateAndTime`)
- 「niDCPower キャリブレーションのユーザ定義情報の最大サイズを取得」VI (`niDCPower_GetCalUserDefinedInfoMaxSize`)
- 「niDCPower キャリブレーションのユーザ定義情報を設定」VI (`niDCPower_SetCalUserDefinedInfo`)
- 「niDCPower キャリブレーションのユーザ定義情報を取得」VI (`niDCPower_GetCalUserDefinedInfo`)
- 「niDCPower 現在の温度の読み取り」VI (`niDCPower_ReadCurrentTemperature`)
- 「niDCPower 最後の外部キャリブレーション温度を取得」VI (`niDCPower_GetExtCalLastTemp`)

キャリブレーション関数リファレンス

このドキュメントで使用した VI および関数（すべてのキャリブレーション VI と関数を含む）は、『NI-DCPower VI リファレンスヘルプ』と『NI-DCPower Function Reference Help』に記載されています。これらのドキュメントは、**スタート→すべてのプログラム→ National Instruments → NI-DCPower →ドキュメント**にある、『NI DC 電源および SMU ヘルプ』からアクセスすることができます。

サポート情報

技術サポートリソースの一覧は、ナショナルインスツルメンツのウェブサイトでご覧いただけます。ni.com/jp/support では、トラブルシューティングやアプリケーション開発のセルフヘルプリソースから、ナショナルインスツルメンツのアプリケーションエンジニアの E メール / 電話の連絡先まで、あらゆるリソースを参照することができます。

適合宣言（Doc）とは、その会社の自己適合宣言を用いた、さまざまな欧州閣僚理事会指令への適合の宣言のことです。この制度により、電磁両立性（EMC) に対するユーザ保護や製品の安全性に関する情報が提供されます。ご使用の製品の適合宣言は、ni.com/certification（英語）から入手できます。ご使用の製品でキャリブレーションがサポートされている場合、ni.com/calibration からその製品の Calibration Certificate（英語）を入手してご利用になることもできます。

ナショナルインスツルメンツでは、米国本社（11500 North Mopac Expressway, Austin, Texas, 78759-3504）および各国の現地オフィスにてお客様にサポート対応しています。日本国内での電話サポートについては、03-5472-2981（技術サポート直通番号）または 03-5472-2970（大代表）にお電話ください。日本国外での電話サポートについては、各国の営業所にご連絡ください。

イスラエル 972 3 6393737, イタリア 39 02 41309277,
インド 91 80 41190000, 英国 44 (0) 1635 523545,
オーストラリア 1800 300 800, オーストリア 43 662 457990-0,
オランダ 31 (0) 348 433 466, カナダ 800 433 3488,
韓国 82 02 3451 3400, シンガポール 1800 226 5886,
スイス 41 56 2005151, スウェーデン 46 (0) 8 587 895 00,
スペイン 34 91 640 0085, スロベニア 386 3 425 42 00,
タイ 662 278 6777, 台湾 886 02 2377 2222, チェコ 420 224 235 774,
中国 86 21 5050 9800, デンマーク 45 45 76 26 00,
ドイツ 49 89 7413130, トルコ 90 212 279 3031,
ニュージーランド 0800 553 322, ノルウェー 47 (0) 66 90 76 60,
フィンランド 358 (0) 9 725 72511, フランス 01 57 66 24 24,
ブラジル 55 11 3262 3599, 米国 512 683 0100,
ベルギー 32 (0) 2 757 0020, ポーランド 48 22 328 90 10,
ポルトガル 351 210 311 210, マレーシア 1800 887710,
南アフリカ 27 0 11 805 8197, メキシコ 01 800 010 0793,
レバノン 961 (0) 1 33 28 28, ロシア 7 495 783 6851

374753B-01 2008 年 11 月