

CALIBRATION PROCEDURE

NI PXI-4110

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-4110 programmable DC power supply. Calibration is generally performed at National Instruments or a metrology lab with an external high-precision digital multimeter (DMM) and other test equipment. For more information about calibration, visit ni.com/calibration.

Contents

Conventions	2
Software Requirements	2
Documentation Requirements	3
Password	4
Calibration Interval	4
Test Equipment	4
Test Conditions	5
Calibration Procedures	5
Initial Setup	6
Verification	6
Verifying Voltage Programming Accuracy	7
Verifying Voltage Measurement Accuracy	11
Verifying Current Output Accuracy	16
Verifying Current Measurement Accuracy	21
Adjustment	27
Adjusting Voltage Programming Accuracy	28
Adjusting Voltage Measurement Accuracy	32
Adjusting Current Output Accuracy	37
Adjusting Current Measurement Accuracy	42
Appendix A: Calibration Options	47
Complete Calibration	48
Optional Calibration	49

Appendix B: Calibration Utilities	50
Calibration Function References	50
Where to Go for Support	51

Conventions

»	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.
<i>italic</i>	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.
<i>monospace italic</i>	Italic text in this font denotes text that is a placeholder for a word or value that you must supply.

Software Requirements

To calibrate the NI PXI-4110, you must install NI-DCPower version 1.0 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 in parentheses. 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 1.0 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-4110 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

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-4110 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](#) section for more information.

Test Equipment

Table 2 lists the equipment required to calibrate the NI PXI-4110. 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-4110 Calibration

Required Equipment	Recommended Equipment	Specifications
Digital multimeter (DMM)	NI 4071	Voltage: $\leq \pm 50$ ppm accuracy, $\leq 30 \mu\text{V}$ resolution; Current: $\leq \pm 0.04\%$ accuracy, ≤ 50 nA resolution
External load	Clarostat 240C	Power resistor decade box with a range of 3 to 25,000 Ω and an accuracy of $\pm 10\%$, and minimum power rating of 40 W per decade
Auxiliary power supply	NI APS-4100	11 V to 15.5 V, 5 A
Twisted pair, shielded cabling wire	—	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 $23\text{ }^{\circ}\text{C} \pm 10\text{ }^{\circ}\text{C}$.
- Allow a warm-up time of at least 15 minutes after the NI-DCPower driver is loaded. Unless manually disabled, NI-DCPower automatically loads with the operating system and enables the device. The warm-up time ensures that the measurement circuitry of the NI PXI-4110 is at a stable operating temperature.

Calibration Procedures

The calibration process includes the following steps:

1. **Initial Setup**—Install the device and configure it in Measurement & Automation Explorer (MAX).
2. **Verification**—Verify the existing operation of the device. This step 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 steps are described in more detail in the following sections.



Note The complete external calibration procedure consists of verifying the performance of the power supply, 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.

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-4110 Channel(s)	DMM*		
	Function	Range [†]	Input Impedance [†]
0	DC Volts	10 V	10 GΩ
1, 2		100 V	10 MΩ
* Use the highest resolution available on the DMM.			
† 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 Output and Measurement Verification/Adjustment

NI PXI-4110 Channel(s)	DMM*		
	Function	Range	Input Impedance
0	DC Current	1 A	10 MΩ
1, 2 (20 mA)		0.1 A	
1, 2 (1 A)		1 A	

* Use the highest resolution available on the DMM.

Verification

This section describes the program you must write to verify the published specifications for the NI PXI-4110.

Verification consists of generating and measuring a series of outputs using the NI PXI-4110, 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-4110 meets its published specifications, and adjustment is optional. If the results fall outside of the test limits, you must adjust the NI PXI-4110.

Verification tests the following NI PXI-4110 specifications:

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



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

Verification of the NI PXI-4110 is complete only after you have successfully completed all tests in this section.

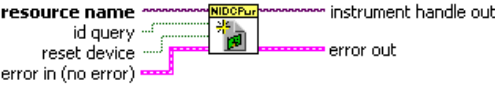


Note If verification fails post-adjustment, confirm that you have met the required *Test Conditions* before you return the NI PXI-4110 to NI for repair.

Verifying Voltage Programming Accuracy

Complete the following steps to verify the voltage programming accuracy of the NI PXI-4110. 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-4110.



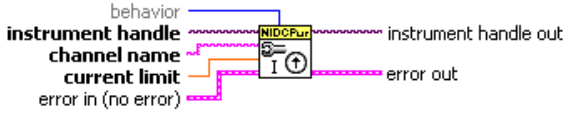
Note Channel *x* represents the channel under test. Replace the variable *x* in the program with the actual channel name.

3. Configure the DMM using the configuration settings for channel *x* in Table 3.

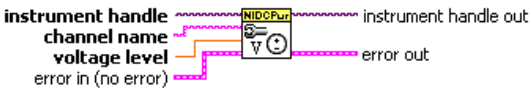
4. Place the NI PXI-4110 in delayed configuration mode using the niDCPower Abort VI.

LabVIEW VI	C/C++ Function Call
	Call <code>niDCPower_Abort</code> with the following parameter: vi: The instrument handle from <code>niDCPower_init</code>

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

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

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

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

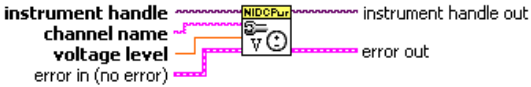
7. Enable the output using the niDCPower Configure Output Enabled VI.

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

8. 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

9. Wait 50 ms for the output of the NI PXI-4110 to settle.
10. Measure the output voltage with the DMM.
11. Record the measurement.
12. To calculate the output error, subtract the measurement you recorded in step 11 from the *Output* value for the iteration of channel x .
13. Compare the output error to the *Test Limit* for the iteration of channel x in Table 5. If the output error is outside of the test limit, adjustment of the NI PXI-4110 is necessary.
14. Repeat steps 4 through 13 for each iteration of channel x in Table 5. When you have verified all iterations of channel x , proceed to step 15.
15. 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: 0

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

LabVIEW VI	C/C++ Function Call
	Call <code>niDCPower_ConfigureOutputEnabled</code> with the following parameters: vi: The instrument handle from <code>niDCPower_init</code> channelName: <i>x</i> enabled: <code>VI_FALSE</code>

17. Disconnect the DMM.
18. Repeat steps 2 through 17 for all unverified channels in Table 5. When you have verified all iterations per channel, this part of the verification is complete.
19. End the session using the niDCPower Close VI.

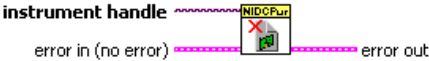
LabVIEW VI	C/C++ Function Call
	Call <code>niDCPower_close</code> with the following parameter: vi: The instrument handle from <code>niDCPower_init</code>

Table 5. NI PXI-4110 Output Parameters and Test Limits for Voltage Programming Accuracy Verification

Channel	Iteration	Output (V)	Test Limit (V)
0	1	0	± 0.00400
	2	1.5	± 0.00475
	3	3	± 0.00550
	4	4.5	± 0.00625
	5	6	± 0.00700

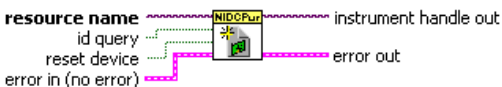
Table 5. NI PXI-4110 Output Parameters and Test Limits for Voltage Programming
Accuracy Verification (Continued)

Channel	Iteration	Output (V)	Test Limit (V)
1	1	0	±0.01000
	2	5	±0.01250
	3	10	±0.01500
	4	15	±0.01750
	5	20	±0.02000
2	1	0	±0.01000
	2	–5	±0.01250
	3	–10	±0.01500
	4	–15	±0.01750
	5	–20	±0.02000

Verifying Voltage Measurement Accuracy

Complete the following steps to verify the voltage measurement accuracy of the NI PXI-4110. 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-4110.



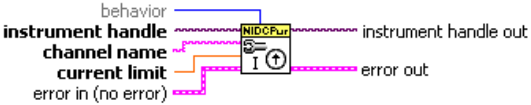
Note Channel *x* represents the channel under test. Replace the variable *x* in the program with the actual channel name.

3. Configure the DMM using the configuration settings for channel *x* in Table 3.

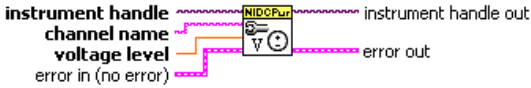
4. Place the NI PXI-4110 in delayed configuration mode using the niDCPower Abort VI.

LabVIEW VI	C/C++ Function Call
	Call <code>niDCPower_Abort</code> with the following parameter: vi: The instrument handle from <code>niDCPower_init</code>

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

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

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

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

7. 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: <i>x</i> enabled: VI_TRUE

8. 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 attribute: NIDCPOWER_ATTR_SAMPLES_ TO_AVERAGE channelName: <i>x</i> value: 300

9. Apply the configuration using the niDCPower Initiate VI.

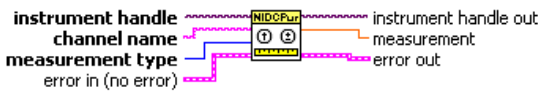
LabVIEW VI	C/C++ Function Call
	Call niDCPower_Initiate with the following parameter: vi: The instrument handle from niDCPower_init

10. Wait 50 ms for the output of the NI PXI-4110 to settle.

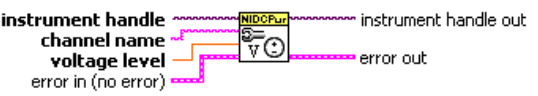
11. Measure the output voltage with the DMM.

12. Record the measurement.

13. Measure the output voltage using the niDCPower Measure VI.

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

14. Record the measurement.
15. To calculate the measurement error, subtract the measurement you recorded in step 12 from the measurement you recorded in step 14.
16. Verify that the measurement error falls between the tolerances listed in the *Test Limit* (V) column for the iteration of channel x in Table 6. Tolerances are provided instead of absolute limits because the DMM measures a unique value. If the measurement error is outside of the test limit, adjustment of the NI PXI-4110 is necessary.
17. Repeat steps 4 through 16 for each iteration of channel x in Table 6. When you have verified all iterations of channel x , proceed to step 18.
18. Configure the voltage level using the niDCPower Configure Voltage Level VI.

LabVIEW VI	C/C++ Function Call
	Call <code>niDCPower_ConfigureVoltageLevel</code> with the following parameters: vi: The instrument handle from <code>niDCPower_init</code> 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: <i>x</i> enabled: VI_FALSE

20. Disconnect the DMM.
21. Repeat steps 2 through 20 for all unverified channels in Table 6. 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 6. NI PXI-4110 Output Parameters and Test Limits for Voltage Measurement Accuracy Verification

Channel	Iteration	Output (V)	Test Limit (V)
0	1	0	±0.004000
	2	1.5	±0.004750
	3	3	±0.005500
	4	4.5	±0.006250
	5	6	±0.007000

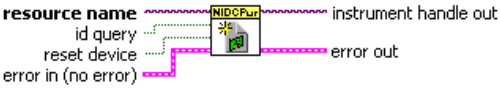
Table 6. NI PXI-4110 Output Parameters and Test Limits for Voltage Measurement Accuracy Verification (Continued)

Channel	Iteration	Output (V)	Test Limit (V)
1	1	0	± 0.005000
	2	5	± 0.007500
	3	10	± 0.010000
	4	15	± 0.012500
	5	20	± 0.015000
2	1	0	± 0.005000
	2	-5	± 0.007500
	3	-10	± 0.010000
	4	-15	± 0.012500
	5	-20	± 0.015000

Verifying Current Output Accuracy

Complete the following steps to verify the current output accuracy of the NI PXI-4110. Complete this test 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 <code>niDCPower_init</code> with the following parameters: resourceName: The device name assigned by MAX idQuery: <code>VI_FALSE</code> resetDevice: <code>VI_TRUE</code>

2. Connect the DMM and the external load to the channel x output terminals of the NI PXI-4110, as illustrated in Figure 1.



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

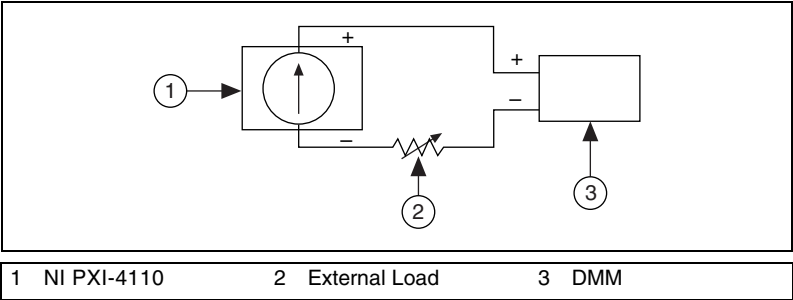


Figure 1. Current Accuracy Verification Test System

3. Configure the DMM using the configuration settings for channel x in Table 4.
4. Configure the external load using the *Equivalent Resistance* value for the iteration of channel x in Table 8.
5. Place the NI PXI-4110 in delayed configuration mode using the niDCPower Abort VI.

LabVIEW VI	C/C++ Function Call
instrument handle instrument handle out error in (no error) error out	Call niDCPower_Abort with the following parameter: vi: The instrument handle from niDCPower_init

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

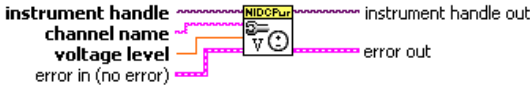
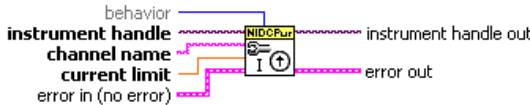
LabVIEW VI	C/C++ Function Call
	Call niDCPower_ConfigureVoltageLevel with the following parameters: vi: The instrument handle from niDCPower_init channelName: <i>x</i> level: The <i>Output</i> value for channel <i>x</i> in Table 7

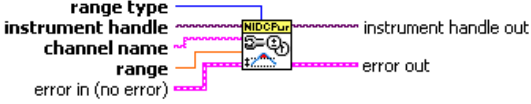
Table 7. Voltage Level Output Parameters

Channel	Output (V)
0	6
1	16
2	-16

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: <i>x</i> behavior: NIDCPOWER_VAL_CURRENT_REGULATE limit: The <i>Output</i> value for the iteration of channel <i>x</i> in Table 8

8. Configure the output range using the niDCPower Configure Output Range VI.

LabVIEW VI	C/C++ Function Call
	Call niDCPower_ConfigureOutputRange with the following parameters: vi: The instrument handle from niDCPower_init channelName: x rangeType: NIDCPOWER_VAL_RANGE_CURRENT range: The <i>Current Range</i> value for the iteration of channel x in Table 8

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

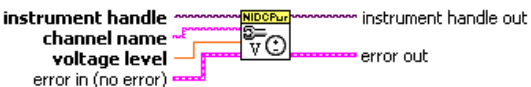
LabVIEW VI	C/C++ Function Call
	Call niDCPower_ConfigureOutputEnabled with the following parameters: vi: The instrument handle from niDCPower_init channelName: x enabled: VI_TRUE

10. 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

11. If the *Equivalent Resistance* value for this iteration of channel x is 25,000 Ω , wait 15 s for the output of the NI PXI-4110 to settle. For all other *Equivalent Resistance* values, wait 3 s for the output of the NI PXI-4110 to settle.
12. Measure the output current using the DMM.
13. Record the measurement.
14. To calculate the output error, subtract the measurement you recorded in step 13 from the *Output* value for the iteration of channel x .

15. Compare the output error to the *Test Limit* for the iteration of channel x in Table 8. If the output error is outside of the test limit, adjustment of the NI PXI-4110 is necessary.
16. Repeat steps 4 through 15 for all iterations of channel x in Table 8 per supported current range. When you have verified all iterations of channel x , proceed to step 17.
17. Configure the voltage level using the niDCPower Configure Voltage Level VI.

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

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

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

19. Disconnect the DMM and the external load.
20. Repeat steps 2 through 19 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.
21. End the session using the niDCPower Close VI.

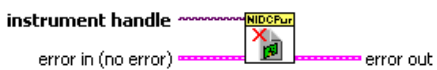
LabVIEW VI	C/C++ Function Call
	Call <code>niDCPower_close</code> with the following parameter: vi: The instrument handle from <code>niDCPower_init</code>

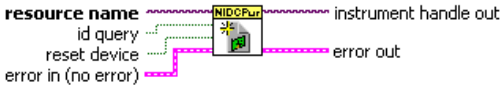
Table 8. NI PXI-4110 Output Parameters and Test Limits for Current Output Accuracy Verification

Channel(s)	Current Range	Iteration	Equivalent Resistance (Ω)	Output (A)	Test Limit (A)
0	1 A	1	150	0.020000	± 0.004030
		2	12	0.250000	± 0.004375
		3	6	0.500000	± 0.004750
		4	4	0.750000	± 0.006063
		5	3	1.000000	± 0.008000
1, 2	1 A	1	500	0.020000	± 0.004030
		2	40	0.250000	± 0.004375
		3	20	0.500000	± 0.004750
		4	13	0.750000	± 0.006063
		5	9	1.000000	± 0.008000
1, 2	20 mA	1	25,000	0.000400	± 0.000060
		2	2000	0.005000	± 0.000068
		3	1000	0.010000	± 0.000075
		4	670	0.015000	± 0.000083
		5	500	0.020000	± 0.000090

Verifying Current Measurement Accuracy

Complete the following steps to verify the current measurement accuracy of the NI PXI-4110. Complete this test for each channel iteration per supported range in Table 9.

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 *Output* value for this iteration of channel *x* is 0, skip to step 5. Do *not* connect the DMM or the external load to the channel *x* output terminals of the NI PXI-4110.

For all other *Output* values, connect the DMM and the external load to the channel *x* output terminals of the NI PXI-4110, as illustrated in Figure 1.



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

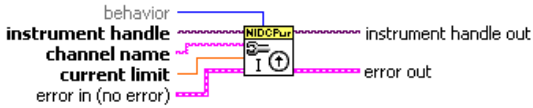
3. Configure the DMM using the configuration settings for channel *x* in Table 4.
4. Configure the external load using the *Equivalent Resistance* value for the iteration of channel *x* in Table 9.
5. Place the NI PXI-4110 in delayed configuration mode using the niDCPower Abort VI.

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

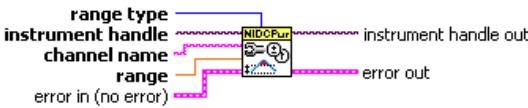
6. Configure the voltage level using the niDCPower Configure Voltage Level VI.

LabVIEW VI	C/C++ Function Call
	Call niDCPower_ConfigureVoltageLevel with the following parameters: vi: The instrument handle from niDCPower_init channelName: <i>x</i> level: The <i>Output</i> value for the iteration of channel <i>x</i> 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_init channelName: x behavior: NIDCPOWER_VAL_CURRENT_REGULATE limit: The <i>Current Range</i> value for the iteration of channel x in Table 9

8. Configure the output range using the niDCPower Configure Output Range VI.

LabVIEW VI	C/C++ Function Call
	Call niDCPower_ConfigureOutputRange with the following parameters: vi: The instrument handle from niDCPower_init channelName: x rangeType: NIDCPOWER_VAL_RANGE_CURRENT range: The <i>Current Range</i> value for the iteration of channel x in Table 9

9. 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

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_init attribute: NIDCPOWER_ATTR_SAMPLES_TO_AVERAGE channelName: <i>x</i> value: 300

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-4110 to settle.
13. If the *Output* value for this iteration of channel *x* is 0, skip to step 14. For all other *Output* values, measure the output current using the DMM.
14. Record the measurement. If the *Output* value for this iteration of channel *x* is 0, you did not take a measurement; record “0” in place of the measurement.



Note Measure the output current using the niDCPower Measure VI. To facilitate an accurate measurement using an *Output* value of 0, verify that the DMM and the external load are disconnected from the NI PXI-4110. For all other *Output* values, the DMM and the external load are connected to the NI PXI-4110.

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

15. Record the measurement.
16. To calculate the measurement error, subtract the measurement you recorded in step 14 from the measurement you recorded in step 15.
17. Verify that the measurement error falls between the tolerances listed in the *Test Limit* for the iteration of channel *x* in Table 9. Tolerances are provided instead of absolute limits because the DMM measures a unique value. If the measurement error is outside of the test limit, adjustment of the NI PXI-4110 is necessary.
18. Repeat steps 4 through 17 for all iterations of channel *x* per supported current range in Table 9. When you have verified all iterations of channel *x*, proceed to step 19.
19. 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: <i>x</i> 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: <i>x</i> enabled: VI_FALSE

21. Disconnect the DMM and the external load.
22. Repeat steps 1 through 21 for all unverified channels in Table 9. 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

If the device has successfully passed all verification tests, the NI PXI-4110 is within the published specifications, and adjustment is optional.

Table 9. NI PXI-4110 Output Parameters and Test Limits for Current Measurement Accuracy Verification

Channel(s)	Current Range	Iteration	Equivalent Resistance (Ω)	Output (V)*	Test Limit (A)	Published Specification (A)
0	1 A	1	open	0	± 0.002000	± 0.004
		2	5	1.25	± 0.002050	± 0.004375
		3	5	2.5	± 0.002500	± 0.00475
		4	5	3.75	± 0.004000	± 0.006063
		5	5	5	± 0.005500	± 0.008000

Table 9. NI PXI-4110 Output Parameters and Test Limits for Current Measurement Accuracy Verification (Continued)

Channel(s)	Current Range	Iteration	Equivalent Resistance (Ω)	Output (V)*	Test Limit (A)	Published Specification (A)
1, 2	1 A	1	open	0	±0.002000	±0.004
		2	20	±5	±0.002050	±0.004375
		3	20	±10	±0.002500	±0.00475
		4	20	±15	±0.004000	±0.006063
		5	20	±20	±0.005500	±0.008000
1, 2	20 mA	1	open	0	±0.000012	±0.000035
		2	1,000	±5	±0.000020	±0.000043
		3	1,000	±10	±0.000027	±0.000050
		4	1,000	±15	±0.000035	±0.000058
		5	1,000	±20	±0.000042	±0.000065
* When verifying channel 1, the <i>Output</i> is positive. When verifying channel 2, the <i>Output</i> is negative.						

Adjustment

Adjustment improves the accuracy of the NI PXI-4110 and updates the calibration date and temperature in the EEPROM. Perform an adjustment once a year or when the accuracy of NI PXI-4110 is outside of the calibration test limits.

Adjustment corrects the following NI PXI-4110 specifications:

- Voltage output accuracy
- Voltage measurement accuracy
- Current output 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-4110 has passed initial verification and is within all calibration 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 (niDCPower_InitExtCal) and end with the niDCPower Close External Calibration VI (niDCPower_CloseExtCal with **action** set to niDCPower_VAL_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-4110. 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-4110.

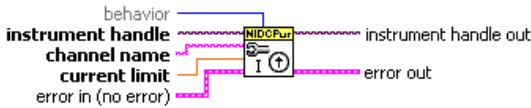


Note Channel x represents the channel under test. Replace the variable x in the program with the actual channel name.

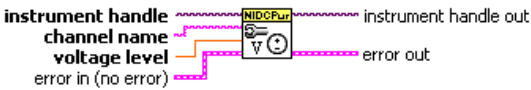
3. Configure the DMM using the configuration settings for channel x in Table 3.
4. Place the NI PXI-4110 in delayed configuration mode using the niDCPower Abort VI.

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

- 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: <i>x</i> behavior: NIDCPOWER_VAL_CURRENT_REGULATE limit: 0.5

- 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: <i>x</i> level: The <i>Output</i> value for the iteration of channel <i>x</i> in Table 10

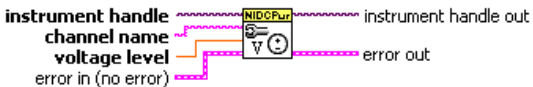
- Enable the output using the niDCPower Configure Output Enabled VI.

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

8. Apply the configuration using the niDCPower Initiate VI.

LabVIEW VI	C/C++ Function Call
	Call <code>niDCPower_Initiate</code> with the following parameter: vi: The instrument handle from <code>niDCPower_InitExtCal</code>

9. Wait 50 ms for the output of the NI PXI-4110 to settle.
10. Measure the output voltage with the DMM.
11. Record the measurement.
12. Repeat steps 4 through 11 for each iteration of channel x in Table 10. When you have measured all iterations of channel x , proceed to step 13.
13. Configure the voltage level using the niDCPower Configure Voltage Level VI.

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

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

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

15. Adjust the voltage level using the niDCPower Cal Adjust Voltage Level VI.

LabVIEW VI	C/C++ Function Call
	Call <code>niDCPower_CalAdjustVoltageLevel</code> with the following parameters: vi: The instrument handle from <code>niDCPower_InitExtCal</code> channelName: x range: The voltage range of channel x numberOfMeasurements: 3 measuredOutputs: An array composed of the measurements you recorded in step 11 for each iteration of channel x requestedOutputs: An array composed of the <i>Output</i> values for each iteration of channel x in Table 10

16. Disconnect the DMM.
17. Repeat steps 2 through 16 for all unadjusted channels in Table 10.
When you have adjusted all channels, this part of the adjustment is complete.
18. End the session using the niDCPower Close External Calibration VI.

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

Table 10. NI PXI-4110 Output Parameters for Voltage Programming Accuracy Adjustment

Channel	Iteration	Output (V)
0	1	0
	2	3
	3	6
1	1	0
	2	10
	3	20
2	1	0
	2	–10
	3	–20

Adjusting Voltage Measurement Accuracy

Complete the following steps to adjust the voltage measurement accuracy of the NI PXI-4110. Complete this test for each 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 <code>niDCPower_InitExtCal</code> 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-4110.



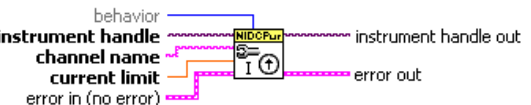
Note Channel *x* represents the channel under test. Replace the variable *x* in the program with the actual channel name.

3. Configure the DMM using the configuration settings for channel *x* in Table 3.

4. Place the NI PXI-4110 in delayed configuration mode using the niDCPower Abort VI.

LabVIEW VI	C/C++ Function Call
	Call <code>niDCPower_Abort</code> with the following parameter: vi: The instrument handle from <code>niDCPower_InitExtCal</code>

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

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

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

LabVIEW VI	C/C++ Function Call
	Call <code>niDCPower_ConfigureVoltageLevel</code> with the following parameters: vi: The instrument handle from <code>niDCPower_InitExtCal</code> channelName: <i>x</i> level: The <i>Output</i> value for the iteration of channel <i>x</i> in Table 11

7. 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: <i>x</i> enabled: VI_TRUE

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

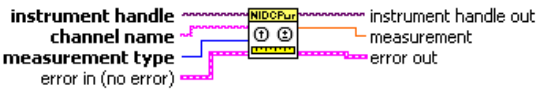
LabVIEW VI	C/C++ Function Call
	Call niDCPower_SetAttributeViInt32 with the following parameters: vi: The instrument handle from niDCPower_InitExtCal attribute: NIDCPOWER_ATTR_SAMPLES_ TO_AVERAGE channelName: <i>x</i> value: 300

9. 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

10. Wait 50 ms for the output of the NI PXI-4110 to settle.
11. Measure the output voltage with the DMM.
12. Record the measurement.

13. Measure the output voltage using the niDCPower Measure VI.

LabVIEW VI	C/C++ Function Call
	Call <code>niDCPower_Measure</code> with the following parameters: vi: The instrument handle from <code>niDCPower_InitExtCal</code> channelName: <i>x</i> measurementType: <code>NIDCPOWER_VAL_MEASURE_VOLTAGE</code>

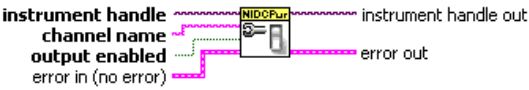
14. Record the measurement.

15. Repeat steps 4 through 14 for each iteration of channel *x* in Table 11. When you have measured all iterations of channel *x*, proceed to step 16.

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

LabVIEW VI	C/C++ Function Call
	Call <code>niDCPower_ConfigureVoltageLevel</code> with the following parameters: vi: The instrument handle from <code>niDCPower_InitExtCal</code> channelName: <i>x</i> level: 0

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

LabVIEW VI	C/C++ Function Call
	Call <code>niDCPower_ConfigureOutputEnabled</code> with the following parameters: vi: The instrument handle from <code>niDCPower_InitExtCal</code> channelName: <i>x</i> enabled: <code>VI_FALSE</code>

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

LabVIEW VI	C/C++ Function Call
	Call <code>niDCPower_CalAdjustVoltageMeasurement</code> with the following parameters: vi: The instrument handle from <code>niDCPower_InitExtCal</code> channelName: x range: The voltage range of channel x numberOfMeasurements: 3 measuredOutputs: An array composed of the measurements you took with the DMM and recorded in step 12 for each iteration of channel x reportedOutputs: An array composed of the measurements you took with the NI PXI-4110 and recorded in step 14 for each iteration of channel x

19. Disconnect the DMM.
20. Repeat steps 2 through 19 for all unadjusted channels in Table 11. When you have adjusted 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 <code>niDCPower_CloseExtCal</code> with the following parameters: vi: The instrument handle from <code>niDCPower_InitExtCal</code> action: <code>niDCPower_VAL_COMMIT</code>

Table 11. NI PXI-4110 Output Parameters for Voltage Measurement Accuracy Adjustment

Channel	Iteration	Output (V)
0	1	0
	2	3
	3	6
1	1	0
	2	10
	3	20
2	1	0
	2	–10
	3	–20

Adjusting Current Output Accuracy

Complete the following steps to adjust the current output accuracy of the NI PXI-4110. Complete this test for each channel iteration per supported range 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
A LabVIEW VI icon for the niDCPower_ExtCal.vi. It features a yellow background with a black border. The text 'resource name' and 'password' are in black, with 'error in (no error)' below them. A red starburst icon is in the center, and a red 'X' icon is at the bottom right. Wavy lines connect the text to the starburst icon. The text 'instrument handle out' is to the right of the starburst icon, and 'error out' is to the right of the 'X' icon.	Call niDCPower_InitExtCal with the following parameters: resourceName: The device name assigned by MAX password: NI

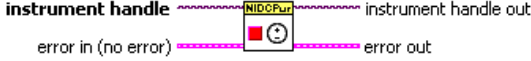
2. Connect the DMM and the external load to the channel *x* output terminals of the NI PXI-4110, as illustrated in Figure 1.



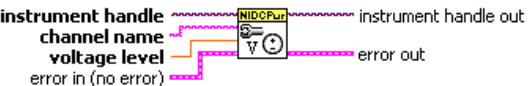
Note Channel *x* represents the channel under test. Replace the variable *x* in the program with the actual channel name.

3. Configure the DMM using the configuration settings for channel *x* in Table 4.
4. Configure the external load using the *Equivalent Resistance* value for the iteration of channel *x* in Table 12.

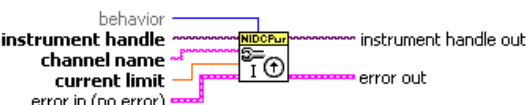
5. Place the NI PXI-4110 in delayed configuration mode using the niDCPower Abort VI.

LabVIEW VI	C/C++ Function Call
	Call <code>niDCPower_Abort</code> with the following parameter: vi: The instrument handle from <code>niDCPower_InitExtCal</code>

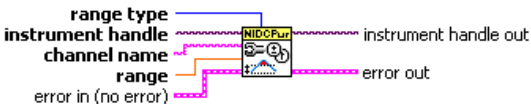
6. Configure the voltage level using the niDCPower Configure Voltage Level VI.

LabVIEW VI	C/C++ Function Call
	Call <code>niDCPower_ConfigureVoltageLevel</code> with the following parameters: vi: The instrument handle from <code>niDCPower_InitExtCal</code> channelName: x level: The <i>Output</i> value for channel x in Table 7

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

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

8. Configure the output range using the niDCPower Configure Output Range VI.

LabVIEW VI	C/C++ Function Call
	Call niDCPower_ConfigureOutputRange with the following parameters: vi: The instrument handle from niDCPower_InitExtCal channelName: x rangeType: NIDCPOWER_VAL_RANGE_CURRENT range: The <i>Current Range</i> value for the iteration of channel x in Table 12

9. 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

10. 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

11. If the *Equivalent Resistance* value for this iteration of channel x is 25,000 Ω , wait 15 s for the output of the NI PXI-4110 to settle. For all other *Equivalent Resistance* values, wait 3 s for the output of the NI PXI-4110 to settle.
12. Measure the output current using the DMM.
13. Record the measurement.

14. Repeat steps 4 through 13 for each iteration of channel x of this current range. When you have measured all iterations of channel x in this current range, proceed to step 14.
15. Adjust the current limit using the niDCPower Cal Adjust Current Limit VI.

LabVIEW VI	C/C++ Function Call
	Call niDCPower_CalAdjustCurrentLimit with the following parameters: vi: The instrument handle from niDCPower_InitExtCal channelName: x range: The current range of channel x numberOfMeasurements: 3 measuredOutputs: An array composed of the measurements you recorded in step 13 for each iteration of channel x requestedOutputs: An array composed of the <i>Output</i> values for each iteration of channel x in Table 12

16. Repeat steps 3 through 15 for all current ranges of channel x in Table 12. When you have adjusted all current ranges of channel x , proceed to step 17.
17. 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: 0

18. 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: <i>x</i> enabled: VI_FALSE

19. Disconnect the DMM.
20. Repeat steps 2 through 19 for all unadjusted channels per current range in Table 12. When you have adjusted all channels per range, this part of the adjustment is complete.
21. End the session using the niDCPower Close External Calibration VI.

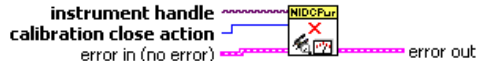
LabVIEW VI	C/C++ Function Call
	Call niDCPower_CloseExtCal with the following parameters: vi: The instrument handle from niDCPower_InitExtCal action: niDCPower_VAL_COMMIT

Table 12. NI PXI-4110 Output Parameters for Current Output Accuracy Adjustment

Channel(s)	Current Range	Iteration	Equivalent Resistance (Ω)	Output (A)
0	1 A	1	80	0.050
		2	12	0.375
		3	5	0.750
1, 2	1 A	1	180	0.050
		2	24	0.375
		3	12	0.750
1, 2	20 mA	1	9,600	0.001
		2	900	0.010
		3	480	0.020

Adjusting Current Measurement Accuracy

Complete the following steps to adjust the current measurement accuracy of the NI PXI-4110. Complete this test for each channel iteration per supported range in Table 13.

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 *Output* value for this iteration of channel *x* is 0, skip to step 5. Do *not* connect the DMM or the external load to the channel *x* output terminals of the NI PXI-4110.

For all other *Output* values, connect the DMM and the external load to the channel *x* output terminals of the NI PXI-4110, as illustrated in Figure 1.

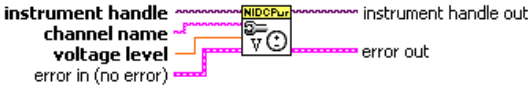


Note Channel *x* represents the channel under test. Replace the variable *x* in the program with the actual channel name.

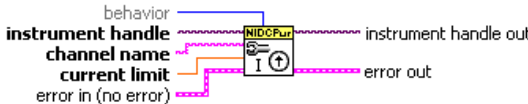
3. Configure the DMM using the configuration settings for channel *x* in Table 4.
4. Configure the external load using the *Equivalent Resistance* value for the iteration of channel *x* in Table 13.
5. Place the NI PXI-4110 in delayed configuration mode using the niDCPower Abort VI.

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

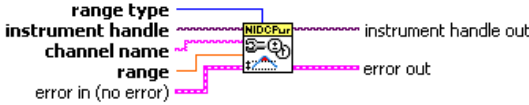
6. Configure the voltage level using the niDCPower Configure Voltage Level VI.

LabVIEW VI	C/C++ Function Call
	Call niDCPower_ConfigureVoltageLevel with the following parameters: vi: The instrument handle from niDCPower_InitExtCal channelName: <i>x</i> level: The <i>Output</i> value for channel <i>x</i> in Table 13

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: <i>x</i> behavior: NIDCPOWER_VAL_CURRENT_REGULATE limit: The <i>Current Range</i> value for the iteration of channel <i>x</i> in Table 13

8. Configure the output range using the niDCPower Configure Output Range VI.

LabVIEW VI	C/C++ Function Call
	Call niDCPower_ConfigureOutputRange with the following parameters: vi: The instrument handle from niDCPower_InitExtCal channelName: <i>x</i> rangeType: NIDCPOWER_VAL_RANGE_CURRENT range: The <i>Current Range</i> value for the iteration of channel <i>x</i> in Table 13

9. 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

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 attribute: NIDCPOWER_ATTR_SAMPLES_TO_AVERAGE channelName: x value: 300

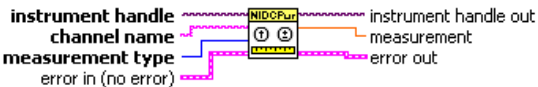
11. Apply the configuration using the niDCPower Initiate VI.

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

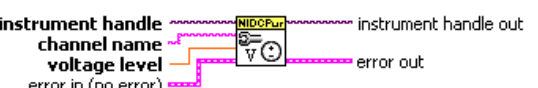
12. Wait 3 s for the output of the NI PXI-4110 to settle.
13. If the *Output* value for this iteration of channel x is 0, skip to step 14. For all other *Output* values, measure the output current using the DMM.
14. Record the measurement. If the *Output* value for this iteration of channel x is 0, you did not take a measurement; record “0” in place of the measurement.



Note Measure the output current using the niDCPower Measure VI. To facilitate an accurate measurement using an *Output* value of 0, verify that the DMM and the external load are disconnected from the NI PXI-4110. For all other *Output* values, the DMM and the external load are connected to the NI PXI-4110.

LabVIEW VI	C/C++ Function Call
	Call <code>niDCPower_Measure</code> with the following parameters: vi: The instrument handle from <code>niDCPower_InitExtCal</code> channelName: <i>x</i> measurementType: <code>NIDCPOWER_VAL_MEASURE_CURRENT</code>

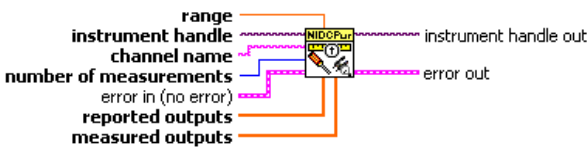
15. Record the measurement.
16. Repeat steps 4 through 15 for each iteration of channel *x* of this current range in Table 13. When you have measured all iterations of channel *x* in this current range, proceed to step 17.
17. Configure the voltage level using the niDCPower Configure Voltage Level VI.

LabVIEW VI	C/C++ Function Call
	Call <code>niDCPower_ConfigureVoltageLevel</code> with the following parameters: vi: The instrument handle from <code>niDCPower_InitExtCal</code> channelName: <i>x</i> level: 0

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

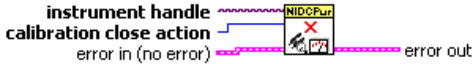
LabVIEW VI	C/C++ Function Call
	Call niDCPower_ConfigureOutputEnabled with the following parameters: vi: The instrument handle from niDCPower_InitExtCal channelName: x enabled: VI_FALSE

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

LabVIEW VI	C/C++ Function Call
	Call niDCPower_CalAdjustCurrent Measurement with the following parameters: vi: The instrument handle from niDCPower_InitExtCal channelName: x range: The current range of channel x numberOfMeasurements: 3 measuredOutputs: An array composed of the measurements you took with the DMM and recorded in step 14 for each iteration of channel x reportedOutputs: An array composed of the measurements you took with the NI PXI-4110 and recorded in step 15 for each iteration of channel x

20. Disconnect the DMM.
21. Repeat steps 2 through 20 for all unadjusted channels in Table 13.
When you have adjusted all channels, this part of the adjustment is complete.

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

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

When you have successfully completed all adjustment tests, adjustment of the NI PXI-4110 is complete. Repeat the [Verification](#) section to reverify the performance of the NI PXI-4110 post-adjustment. If the NI PXI-4110 successfully passes all verification tests, calibration is complete.

Table 13. NI PXI-4110 Output Parameters for Current Measurement Accuracy Adjustment

Channel(s)	Current Range	Iteration	Equivalent Resistance (Ω)	Output (V)*
0	1 A	1	open	0
		2	5	1.875
		3	5	3.75
1, 2	1 A	1	open	0
		2	20	± 7.5
		3	20	± 15
1, 2	20 mA	1	open	0
		2	1,000	± 10
		3	1,000	± 20

* When adjusting channel 1, the *Output* is positive. When adjusting channel 2, the *Output* is negative.

Appendix A: Calibration Options

Calibration involves verification, and, if necessary, adjustment, and reverification of the NI PXI-4110.

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-4110 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-4110 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 2 shows the programming flow for a complete calibration.

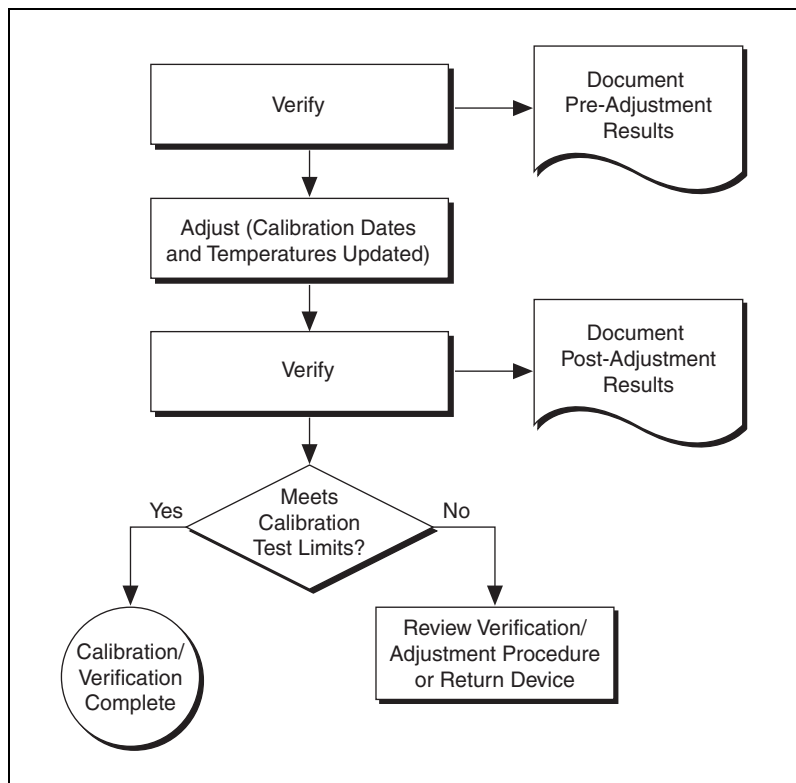


Figure 2. Complete Calibration Programming Flow

Optional Calibration

If the accuracy of the device is within the calibration test limits during the initial verification, the NI PXI-4110 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 3 shows the programming flow for the optional calibration.

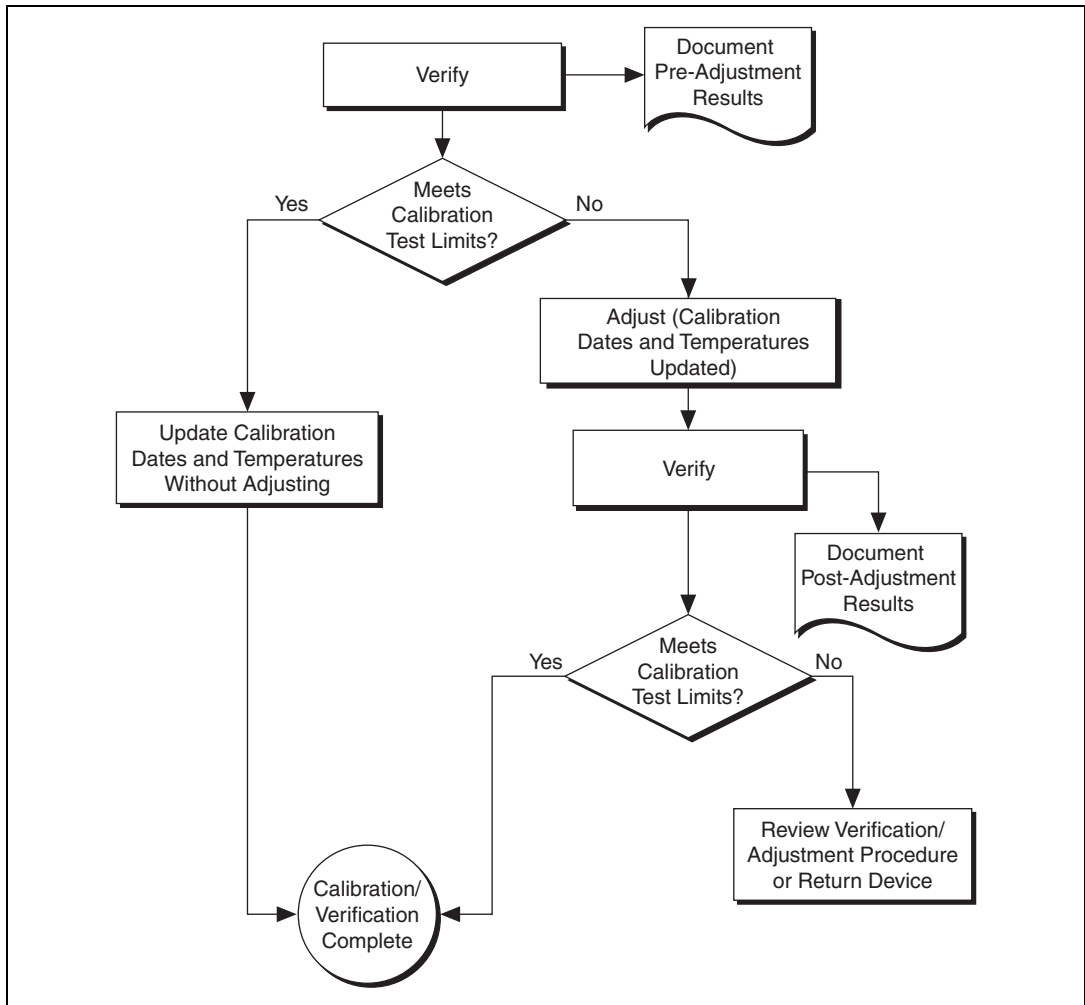


Figure 3. Optional Calibration Programming Flow

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-4110, 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 electronic 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 22 328 90 10, 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

National Instruments, NI, ni.com, and LabVIEW are trademarks of National Instruments Corporation. Refer to the *Terms of Use* section on ni.com/legal for more information about National Instruments trademarks. Other product and company names mentioned herein are trademarks or trade names of their respective companies. For patents covering National Instruments products/technology, refer to the appropriate location: **Help»Patents** in your software, the `patents.txt` file on your media, or the *National Instruments Patent Notice* at ni.com/patents.

キャリブレーション手順

NI PXI-4110

このドキュメントには、NI PXI-4110 プログラマブル DC 電源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

表記規則	2
ソフトウェア要件	2
ドキュメント要件	3
パスワード	4
キャリブレーション間隔	4
テスト装置	4
テスト条件	4
キャリブレーション手順	5
初期設定	5
検証	6
電圧プログラミング確度を検証する	7
電圧測定確度を検証する	11
電流出力確度を検証する	15
電流測定確度を検証する	20
調整	26
電圧プログラミング確度を調整する	26
電圧測定確度を調整する	30
電流出力確度を調整する	35
電流測定確度を調整する	40
付録 A: キャリブレーションオプション	45
完全なキャリブレーション	46
オプションのキャリブレーション	46
付録 B: キャリブレーションユーティリティ	48
キャリブレーション関数リファレンス	48
サポート情報	49

表記規則

→

矢印（→）は、ネスト化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。**ファイル→ページ設定→オプション**と表記されている場合は、まず**ファイル**メニューをプルダウンし、次に**ページ設定**項目を選択し、最後にダイアログボックスで**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オプションなど、ソフトウェアでユーザが選択またはクリックする必要がある項目を示します。また、太字のテキストはパラメータ名を示します。

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、斜体のテキストは、ユーザが入力する必要がある語句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-4110 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 1.0 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++ と LabVIEW での「**キャリブレーション手順**」のプログラミングをサポートしています。LabWindows™/CVI™ では、C++ 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++ 関数を括弧内に記載しています。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++ 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 1.0 以降)

ファイル名および保存場所	説明
<code>IVI¥Bin¥niDCPower_32.dll</code>	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。
<code>IVI¥Lib¥msc¥niDCPower.lib</code>	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。
<code><LabVIEW>¥instr.lib¥niDCPowerCalibrate¥niDCPower.llb</code>	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。
<code>IVI¥Drivers¥niDCPower¥niDCPower.fp</code>	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。
<code>IVI¥Include¥niDCPower.h</code>	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-4110 仕様
- NI DC 電源 /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-4110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](#)」セクションを参照してください。

テスト装置

表 2 は、NI PXI-4110 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、代替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-4110 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 $\leq \pm 50$ ppm、分解能 ≤ 30 μ V、 電流: 確度 $\leq \pm 0.04\%$ 、分解能 ≤ 50 nA
外部負荷	Clarostat 240C	範囲が 3 ~ 25,000 Ω および確度が $\pm 10\%$ の ディケード抵抗ボックス (各ディケードの最低定格電力は 40 W)
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	—	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ルタ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 °C ±10 °C 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 15 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-4110 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、電源の性能検証、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。

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-4110 チャンネル	DMM*		
	機能	レンジ†	入力インピーダンス†
0	DC 電圧	10 V	10 GΩ
1, 2		100 V	10 MΩ

* DMM で使用可能な最高の分解能を使用。
† 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 電流出力と測定検証 / 調整用のキャリブレーション機器構成

NI PXI-4110 チャンネル	DMM*		
	機能	レンジ	入力インピーダンス
0	DC 電流	1 A	10 MΩ
1, 2 (20 mA)		0.1 A	
1, 2 (1 A)		1 A	

* DMM で使用可能な最大分解能を使用。

検証

このセクションでは、NI PXI-4110 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-4110 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-4110 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-4110 を調整する必要があります。

検証により、以下の NI PXI-4110 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
- 電流出力確度
- 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。

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-4110 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-4110 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-4110 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. DMM を NI PXI-4110 のチャンネル x 出力端子に接続します。

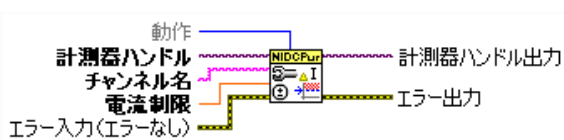


メモ チャンネル x は、テスト中のチャンネルを表します。プログラムの変数 x を実際のチャンネル名と置き換えてください。

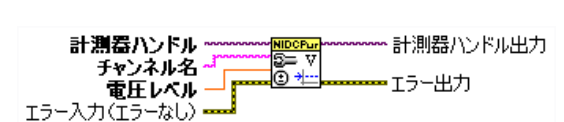
3. 表 3 のチャンネル x の構成設定にしたがって、DMM を構成します。
4. 「niDCPower 中止」VI を使用して、NI PXI-4110 を遅延構成モードにします。

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

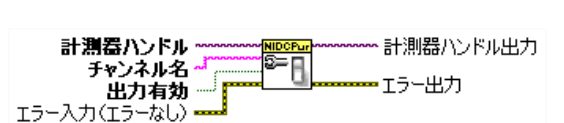
5. 「niDCPower 電流制限を構成」 VI を使用して、電流制限を構成します。

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

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 に対応する出力値

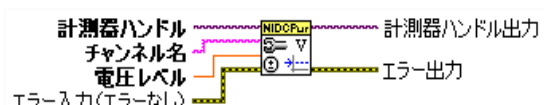
7. 「niDCPower 出力有効化を構成」 VI を使用して、出力を有効にします。

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

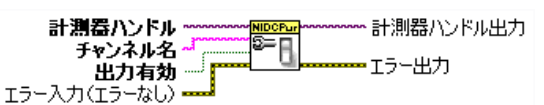
8. 「niDCPower 開始」 VI を使用して構成を適用します。

LabVIEW VI	C/C++ 関数呼び出し
 計測器ハンドル エラー入力(エラーなし) → niDCPower Init → 計測器ハンドル出力 エラー出力	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

9. NI PXI-4110 の出力が安定するまで 50 ms 間待機します。
10. DMM を使用して出力電圧を測定します。
11. 測定値を記録します。
12. 出力誤差を計算するには、チャンネル x の出力値と、手順 11 で記録した測定値の差を求めます。
13. 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-4110 を調整する必要があります。
14. 表 5 に示したチャンネル x の各反復に対して手順 4 ~ 13 を繰り返します。チャンネル x のすべての反復を検証した後、手順 15 に進みます。
15. 「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
 計測器ハンドル チャンネル名 電圧レベル エラー入力(エラーなし) → niDCPower Configure Voltage Level → 計測器ハンドル出力 エラー出力	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

16. 「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
 計測器ハンドル チャンネル名 出力有効 エラー入力(エラーなし) → niDCPower Configure Output Enabled → 計測器ハンドル出力 エラー出力	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

17. DMM との接続を切断します。
18. 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 ～ 17 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。
19. 「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

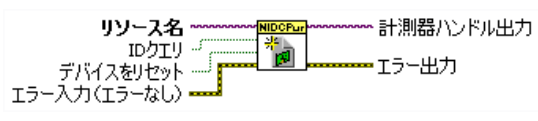
表 5 NI PXI-4110 出力パラメータとテスト制限値（電圧プログラミング確度検証時）

チャンネル	反復	出力 (V)	テスト制限値 (V)
0	1	0	±0.00400
	2	1.5	±0.00475
	3	3	±0.00550
	4	4.5	±0.00625
	5	6	±0.00700
1	1	0	±0.01000
	2	5	±0.01250
	3	10	±0.01500
	4	15	±0.01750
	5	20	±0.02000
2	1	0	±0.01000
	2	-5	±0.01250
	3	-10	±0.01500
	4	-15	±0.01750
	5	-20	±0.02000

電圧測定確度を検証する

以下の手順を実行して、NI PXI-4110 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
 リソース名 IDクエリ デバイスをリセット エラー入力(エラーなし) 計測器ハンドル出力 エラー出力	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. DMM を NI PXI-4110 のチャンネル x 出力端子に接続します。



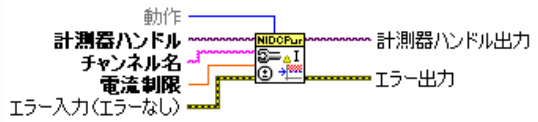
メモ

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。

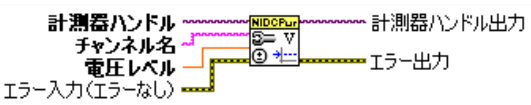
3. 表 3 のチャンネル x の構成設定にしたがって、DMM を構成します。
4. 「niDCPower 中止」VI を使用して、NI PXI-4110 を遅延構成モードにします。

LabVIEW VI	C/C++ 関数呼び出し
 計測器ハンドル エラー入力(エラーなし) 計測器ハンドル出力 エラー出力	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 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
 動作 計測器ハンドル チャンネル名 電流制限 エラー入力(エラーなし) 計測器ハンドル出力 エラー出力	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

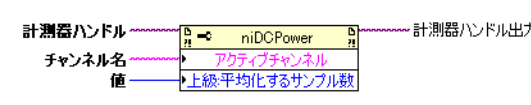
6. 「niDCPower 電圧レベルを構成」VI を使用して電圧レベルを構成します。

LabVIEW VI	C/C++ 関数呼び出し
 計測器ハンドル チャンネル名 電圧レベル エラー入力(エラーなし) 計測器ハンドル出力 エラー出力	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 のチャンネル x の反復に対応する出力値

7. 「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
 計測器ハンドル チャンネル名 出力有効 エラー入力(エラーなし) 計測器ハンドル出力 エラー出力	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

8. 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
 計測器ハンドル チャンネル名 値 計測器ハンドル出力	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 attribute: NIDCPOWER_ATTR_SAMPLES_TO_AVERAGE channelName: x value: 300

9. 「niDCPower 開始」 VI を使用して構成を適用します。

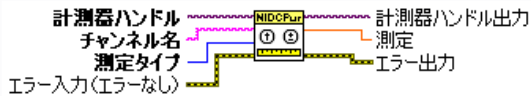
LabVIEW VI	C/C++ 関数呼び出し
	以下のパラメータを使用して、「niDCPower_Initiate」を呼び出します。 vi: 「niDCPower_init」からの計測器ハンドル

10. NI PXI-4110 の出力が安定するまで 50 ms 間待機します。

11. DMM を使用して出力電圧を測定します。

12. 測定値を記録します。

13. 「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: <i>x</i> measurementType: NIDCPOWER_VAL_MEASURE_VOLTAGE

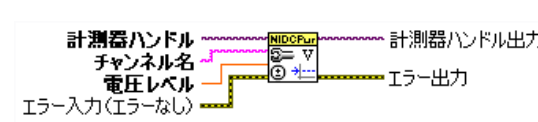
14. 測定値を記録します。

15. 測定誤差を計算するには、手順 14 で記録した測定値と、手順 12 で記録した測定値の差を求めます。

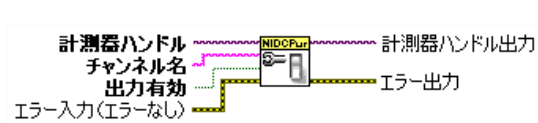
16. 測定誤差が、表 6 のチャンネル *x* の該当する反復のテスト制限値 (V) の許容範囲内であることを確認します。DMM は固有値を測定するため、絶対制限値の代わりに許容範囲が記載されています。測定誤差がテスト制限値の範囲外である場合、NI PXI-4110 を調整する必要があります。

17. 表 6 に示したチャンネル *x* の各反復に対して手順 4 ~ 16 を繰り返します。チャンネル *x* のすべての反復を検証した後、手順 18 に進みます。

18. 「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: <i>x</i> 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: <i>x</i> enabled: VI_FALSE

20. DMM との接続を切断します。
21. 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 ～ 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。

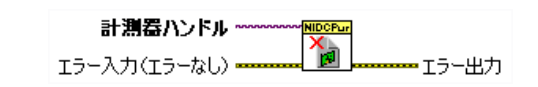
LabVIEW VI	C/C++ 関数呼び出し
	以下のパラメータを使用して、「niDCPower_close」を呼び出します。 vi: 「niDCPower_init」からの計測器ハンドル

表 6 NI PXI-4110 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	反復	出力 (V)	テスト制限値 (V)
0	1	0	±0.004000
	2	1.5	±0.004750
	3	3	±0.005500
	4	4.5	±0.006250
	5	6	±0.007000
1	1	0	±0.005000
	2	5	±0.007500
	3	10	±0.010000
	4	15	±0.012500
	5	20	±0.015000
2	1	0	±0.005000
	2	-5	±0.007500
	3	-10	±0.010000
	4	-15	±0.012500
	5	-20	±0.015000

電流出力確度を検証する

以下の手順を実行して、NI PXI-4110 の電流出力確度を検証します。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
 リソース名 IDクエリ デバイスをリセット エラー入力(エラーなし) 計測器ハンドル出力 エラー出力	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

- 図 1 に示すように、DMM と外部負荷を NI PXI-4110 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。

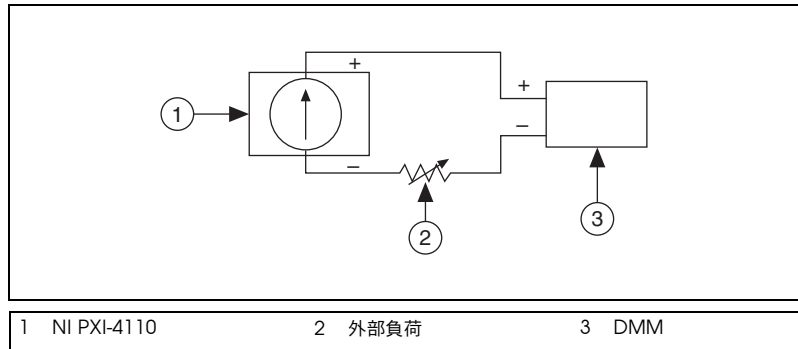


図 1 電流確度検証テストシステム

- 表 4 のチャンネル x の構成設定にしたがって、DMM を構成します。
- 表 8 に示したチャンネル x の反復に対応する等価抵抗値を使用して、外部負荷を構成します。
- 「niDCPower 中止」VI を使用して、NI PXI-4110 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

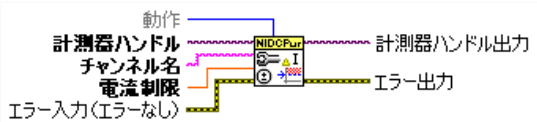
- 「niDCPower 電圧レベルを構成」VI を使用して電圧レベルを構成します。

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 7 のチャンネル x の反復に対応する出力値

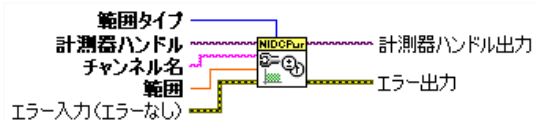
表 7 電圧レベル出力パラメータ

チャンネル	出力 (V)
0	6
1	16
2	-16

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: 表 8 に示したチャンネル x の反復に対応する出力値

8. 「niDCPower 出力範囲を構成」 VI を使用して出力範囲を構成します。

LabVIEW VI	C/C++ 関数呼び出し
	以下のパラメータを使用して、「niDCPower_ConfigureOutputRange」を呼び出します。 vi: 「niDCPower_init」からの計測器ハンドル channelName: x rangeType: NIDCPOWER_VAL_RANGE_CURRENT range: 表 8 に示したチャンネル x の反復に対応する電流範囲値

9. 「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

10. 「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

- チャンネル x の反復の等価抵抗値が 25,000 Ω の場合、NI PXI-4110 の出力が安定するまで 15 秒間待機します。その他の抵抗値に対しては、NI PXI-4110 の出力が安定するまで 3 秒間待機します。
- DMM を使用して出力電流を測定します。
- 測定値を記録します。
- 出力誤差を計算するには、チャンネル x の出力値と、手順 13 で記録した測定値の差を求めます。
- 出力誤差と表 8 に記載したチャンネル x 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-4110 を調整する必要があります。
- 表 8 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 4 ~ 15 を繰り返します。チャンネル x のすべての反復を検証した後、手順 17 に進みます。

17. 「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: 0

18. 「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

19. DMM と外部負荷の接続を切断します。
20. 表 8 に示したすべての未検証のチャンネルに対し、手順 2 ～ 19 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。
21. 「niDCPower 閉じる」 VI を使用してセッションを終了します。

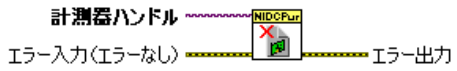
LabVIEW VI	C/C++ 関数呼び出し
	以下のパラメータを使用して、「niDCPower_close」を呼び出します。 vi: 「niDCPower_init」からの計測器ハンドル

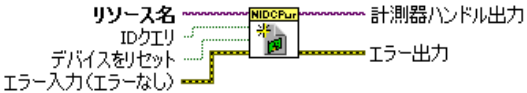
表 8 NI PXI-4110 出力パラメータとテスト制限値（電流出力確度の検証時）

チャンネル	電流範囲	反復	等価抵抗 (Ω)	出力 (A)	テスト制限値 (A)
0	1 A	1	150	0.020000	±0.004030
		2	12	0.250000	±0.004375
		3	6	0.500000	±0.004750
		4	4	0.750000	±0.006063
		5	3	1.000000	±0.008000
1, 2	1 A	1	500	0.020000	±0.004030
		2	40	0.250000	±0.004375
		3	20	0.500000	±0.004750
		4	13	0.750000	±0.006063
		5	9	1.000000	±0.008000
1, 2	20 mA	1	25,000	0.000400	±0.000060
		2	2000	0.005000	±0.000068
		3	1000	0.010000	±0.000075
		4	670	0.015000	±0.000083
		5	500	0.020000	±0.000090

電流測定確度を検証する

以下の手順を実行して、NI PXI-4110 の電流測定確度を検証します。表 9 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. チャンネル x の反復に対応する出力値が 0 の場合、手順 5 に進みます。DMM または外部負荷を NI PXI-4110 のチャンネル x 出力端子に接続しないでください。

その他すべての出力値については、図 1 のように、DMM と外部負荷を NI PXI-4110 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。

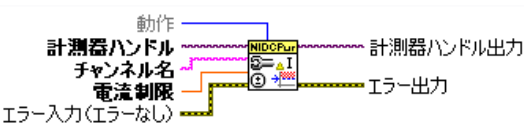
3. 表 4 のチャンネル x の構成設定にしたがって、DMM を構成します。
4. 表 9 に示したチャンネル x の反復に対応する等価抵抗値を使用して、外部負荷を構成します。
5. 「niDCPower 中止」VI を使用して、NI PXI-4110 を遅延構成モードにします。

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

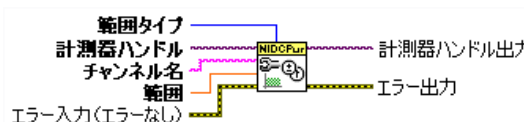
6. 「niDCPower 電圧レベルを構成」VI を使用して電圧レベルを構成します。

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 9 のチャンネル 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: 表 9 に示したチャンネル x の反復に対応する電流範囲値

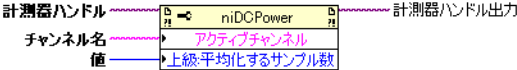
8. 「niDCPower 出力範囲を構成」VI を使用して出力範囲を構成します。

LabVIEW VI	C/C++ 関数呼び出し
	以下のパラメータを使用して、「niDCPower_ConfigureOutputRange」を呼び出します。 vi: 「niDCPower_init」からの計測器ハンドル channelName: x rangeType: NIDCPOWER_VAL_RANGE_CURRENT range: 表 9 に示したチャンネル x の反復に対応する電流範囲値

9. 「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

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 attribute: NIDCPOWER_ATTR_SAMPLES_TO_AVERAGE channelName: x value: 300

11. 「niDCPower 開始」VI を使用して構成を適用します。

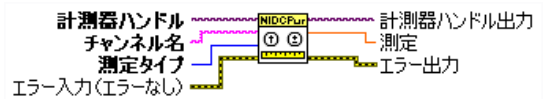
LabVIEW VI	C/C++ 関数呼び出し
	以下のパラメータを使用して、「niDCPower_Initiate」を呼び出します。 vi: 「niDCPower_init」からの計測器ハンドル

12. NI PXI-4110 の出力が安定するまで 3 秒間待機します。
13. チャンネル x の反復に対応する出力値が 0 の場合、手順 14 に進みます。その他の出力値の場合は、DMM を使用して出力電流を測定します。
14. 測定値を記録します。チャンネル x の反復に対応する出力値が 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。



メモ

「niDCPower 測定」VI を使用して出力電流を測定します。出力値が 0 の場合に正確に測定するには、DMM/ 外部負荷と NI PXI-4110 間の接続が切断されていることを確認します。その他すべての出力値の場合は、DMM と外部負荷を NI PXI-4110 に接続します。

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

15. 測定値を記録します。
16. 測定誤差を計算するには、手順 15 で記録した測定値と、手順 14 で記録した測定値の差を求めます。
17. 測定誤差が、表 9 のチャンネル x の該当する反復のテスト制限値の許容範囲内であることを確認します。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。測定誤差がテスト制限値の範囲外である場合、NI PXI-4110 を調整する必要があります。
18. 表 9 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 4 ～ 17 を繰り返します。チャンネル x のすべての反復を検証した後、手順 19 に進みます。
19. 「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: 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 と外部負荷の接続を切断します。
22. 表 9 に示したすべての未検証のチャンネルに対し、手順 1 ～ 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
 計測器ハンドル エラー入力(エラーなし) → NI DC Power → エラー出力	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

デバイスがすべての検証テストに合格し、NI PXI-4110 が公開仕様の範囲内である場合、調整手順は省略可能です。

表 9 NI PXI-4110 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	電流範囲	反復	等価抵抗 (Ω)	出力 (V) *	テスト制限値 (A)	公開仕様 (A)
0	1 A	1	オープン	0	±0.002000	±0.004
		2	5	1.25	±0.002050	±0.004375
		3	5	2.5	±0.002500	±0.00475
		4	5	3.75	±0.004000	±0.006063
		5	5	5	±0.005500	±0.008000
1, 2	1 A	1	オープン	0	±0.002000	±0.004
		2	20	±5	±0.002050	±0.004375
		3	20	±10	±0.002500	±0.00475
		4	20	±15	±0.004000	±0.006063
		5	20	±20	±0.005500	±0.008000
1, 2	20 mA	1	オープン	0	±0.000012	±0.000035
		2	1,000	±5	±0.000020	±0.000043
		3	1,000	±10	±0.000027	±0.000050
		4	1,000	±15	±0.000035	±0.000058
		5	1,000	±20	±0.000042	±0.000065

* チャンネル 1 検証時、出力は正の値。チャンネル 2 検証時、出力は負の値。

調整

調整により、NI PXI-4110 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-4110 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-4110 仕様が補正されます。

- 電圧出力確度
- 電圧測定確度
- 電流出力確度
- 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-4110 が最初の検証テストに合格し、キャリブレーション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情報を更新するには、「niDCPower 外部キャリブレーションを初期化」VI (niDCPower_InitExtCal) を実行し、**action** を「niDCPower_VAL_COMMIT」に設定して「niDCPower 外部キャリブレーションを閉じる」VI (niDCPower_CloseExtCal) 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-4110 の電圧プログラミング確度を調整します。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. DMM を NI PXI-4110 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 のチャンネル x の構成設定にしたがって、DMM を構成します。
4. 「niDCPower 中止」 VI を使用して、NI PXI-4110 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 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

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

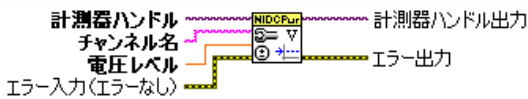
- 「niDCPower 出力有効化を構成」VI を使用して、出力を有効にします。

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

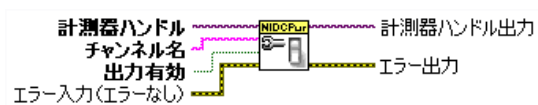
- 「niDCPower 開始」VI を使用して構成を適用します。

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

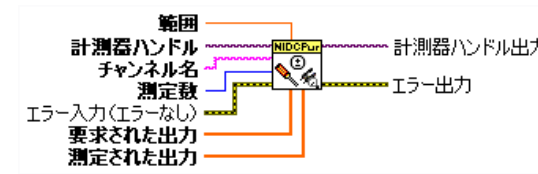
- NI PXI-4110 の出力が安定するまで 50 ms 間待機します。
- DMM を使用して出力電圧を測定します。
- 測定値を記録します。
- 表 10 に示したチャンネル x の各反復に対して手順 4 ~ 11 を繰り返します。チャンネル x のすべての反復を測定した後、手順 13 に進みます。
- 「niDCPower 電圧レベルを構成」VI を使用して電圧レベルを構成します。

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: 0

14. 「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

15. 「niDCPower Cal 電圧レベルを調整」 VI を使用して電圧レベルを調整します。

LabVIEW VI	C/C++ 関数呼び出し
	以下のパラメータを使用して、「niDCPower_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 の電圧範囲 numberOfMeasurements: 3 measuredOutputs: 手順 11 で記録した、チャンネル x の各反復に対する測定値の配列 requestedOutputs: チャンネル x の各反復（表 10 参照）に対応する出力値の配列

16. DMM との接続を切断します。
17. 表 10 に示したすべての未調整チャンネルに対し、手順 2 ～ 16 を繰り返します。すべてのチャンネルを調整すれば、この部分の調整は完了です。
18. 「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。

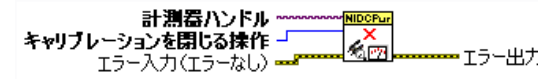
LabVIEW VI	C/C++ 関数呼び出し
	以下のパラメータを使用して、「niDCPower_CloseExtCal」を呼び出します。 vi: 「niDCPower_InitExtCal」からの計測器ハンドル action: niDCPower_VAL_COMMIT

表 10 NI PXI-4110 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	出力 (V)
0	1	0
	2	3
	3	6
1	1	0
	2	10
	3	20
2	1	0
	2	-10
	3	-20

電圧測定確度を調整する

以下の手順を実行して、NI PXI-4110 の電圧測定確度を調整します。表 11 の各反復に対してこのテストを実行します。

- 「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
 リソース名 パスワード エラー入力(エラーなし) niDCPower 計測器ハンドル出力 エラー出力	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

- DMM を NI PXI-4110 のチャンネル x 出力端子に接続します。



メモ

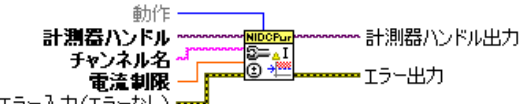
チャンネル x は、テスト中のチャンネルを表します。プログラムの変数 x を実際のチャンネル名と置き換えてください。

- 表 3 のチャンネル x の構成設定にしたがって、DMM を構成します。

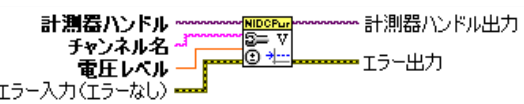
4. 「niDCPower 中止」 VI を使用して、NI PXI-4110 を遅延構成モードにします。

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 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

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 11 に記載したチャンネル x の反復に対応する出力値

7. 「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: <i>x</i> enabled: VI_TRUE

8. 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。

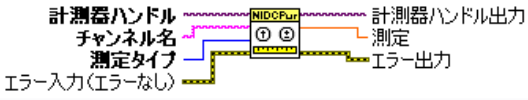
LabVIEW VI	C/C++ 関数呼び出し
	以下のパラメータを使用して、「niDCPower_SetAttributeViInt32」を呼び出します。 vi: 「niDCPower_InitExtCal」からの計測器ハンドル attribute: NIDCPOWER_ATTR_SAMPLES_TO_AVERAGE channelName: <i>x</i> value: 300

9. 「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

10. NI PXI-4110 の出力が安定するまで 50 ms 間待機します。
11. DMM を使用して出力電圧を測定します。
12. 測定値を記録します。

13. 「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

14. 測定値を記録します。

15. 表 11 に示したチャンネル x の各反復に対して手順 4 ～ 14 を繰り返します。チャンネル x のすべての反復を測定した後、手順 16 に進みます。

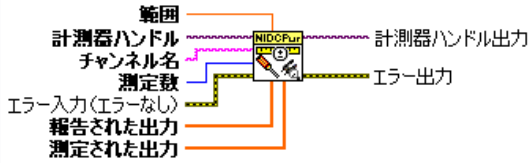
16. 「niDCPower 電圧レベルを構成」 VI を使用して電圧レベルを構成します。

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: 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 measuredOutputs: 手順 12 で記録した、チャンネル x の各反復に対する DMM での測定値の配列 reportedOutputs: 手順 14 で記録した、チャンネル x の各反復に対する NI PXI-4110 での測定値の配列

19. DMM との接続を切断します。
20. 表 11 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。

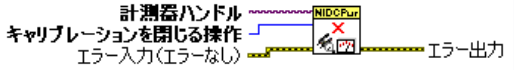
LabVIEW VI	C/C++ 関数呼び出し
	以下のパラメータを使用して、「niDCPower_CloseExtCal」を呼び出します。 vi: 「niDCPower_InitExtCal」からの計測器ハンドル action: niDCPower_VAL_COMMIT

表 11 NI PXI-4110 出力パラメータ（電圧測定確度の調整時）

チャンネル	反復	出力 (V)
0	1	0
	2	3
	3	6
1	1	0
	2	10
	3	20
2	1	0
	2	-10
	3	-20

電流出力確度を調整する

以下の手順を実行して、NI PXI-4110 の電流出力確度を調整します。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. 図 1 に示すように、DMM と外部負荷を NI PXI-4110 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 4 のチャンネル x の構成設定にしたがって、DMM を構成します。
4. 表 12 に示したチャンネル x の反復に対応する等価抵抗値を使用して、外部負荷を構成します。

5. 「niDCPower 中止」 VI を使用して、NI PXI-4110 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 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 7 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: 表 12 に示したチャンネル x の反復に対応する出力値

8. 「niDCPower 出力範囲を構成」 VI を使用して出力範囲を構成します。

LabVIEW VI	C/C++ 関数呼び出し
	以下のパラメータを使用して、「niDCPower_ConfigureOutputRange」を呼び出します。 vi: 「niDCPower_InitExtCal」からの計測器ハンドル channelName: x rangeType: NIDCPOWER_VAL_RANGE_CURRENT range: 表 12 に示したチャンネル x の反復に対応する電流範囲値

9. 「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

10. 「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

- チャンネル x の反復の等価抵抗値が 25,000 Ω の場合、NI PXI-4110 の出力が安定するまで 15 秒間待機します。その他の抵抗値に対しては、NI PXI-4110 の出力が安定するまで 3 秒間待機します。
- DMM を使用して出力電流を測定します。
- 測定値を記録します。
- 同じ電流範囲で、チャンネル x の各反復に対して手順 4 ～ 13 を繰り返します。この電流範囲でチャンネル x のすべての反復を測定した後、手順 14 に進みます。

15. 「niDCPower Cal 電流制限を調整」 VI を使用して、電流制限を調整します。

LabVIEW VI	C/C++ 関数呼び出し
	以下のパラメータを使用して、「niDCPower_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 の電流範囲 numberOfMeasurements: 3 measuredOutputs: 手順 13 で記録した、チャンネル x の各反復に対する測定値の配列 requestedOutputs: チャンネル x の各反復（表 12 参照）に対応する出力値の配列

16. 表 12 に示したチャンネル x のすべての電流範囲に対して、手順 3 ～ 15 を繰り返します。チャンネル x のすべての電流範囲を調整した後、手順 17 に進みます。
17. 「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: 0

18. 「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

19. DMM との接続を切断します。
20. 表 12 に示した電流範囲で、すべての未調整チャンネルに対して手順 2 ～ 19 を繰り返します。すべてのチャンネルの各範囲を調整すれば、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
 計測器ハンドル キャリブレーションを閉じる操作 エラー入力(エラーなし) エラー出力	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 12 NI PXI-4110 出力パラメータ（電流出力確度の調整時）

チャンネル	電流範囲	反復	等価抵抗 (Ω)	出力 (A)
0	1 A	1	80	0.050
		2	12	0.375
		3	5	0.750
1, 2	1 A	1	180	0.050
		2	24	0.375
		3	12	0.750
1, 2	20 mA	1	9,600	0.001
		2	900	0.010
		3	480	0.020

電流測定確度を調整する

以下の手順を実行して、NI PXI-4110 の電流測定確度を調整します。表 13 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。

- 「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

- チャンネル x の反復に対応する出力値が 0 の場合、手順 5 に進みます。DMM または外部負荷を NI PXI-4110 のチャンネル x 出力端子に接続しないでください。
その他すべての出力値については、図 1 のように、DMM と外部負荷を NI PXI-4110 のチャンネル x 出力端子に接続します。

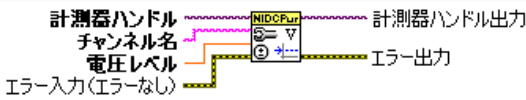


メモ チャンネル x は、テスト中のチャンネルを表します。プログラムの変数 x を実際のチャンネル名と置き換えてください。

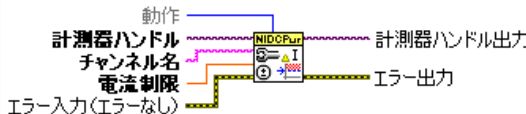
- 表 4 のチャンネル x の構成設定にしたがって、DMM を構成します。
- 表 13 に示したチャンネル x の反復に対応する等価抵抗値を使用して、外部負荷を構成します。
- 「niDCPower 中止」VI を使用して、NI PXI-4110 を遅延構成モードにします。

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

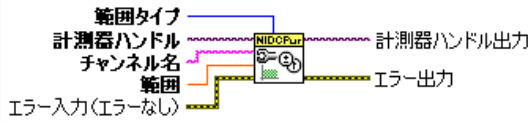
6. 「niDCPower 電圧レベルを構成」 VI を使用して電圧レベルを構成します。

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 13 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: 表 13 に示したチャンネル x の反復に対応する電流範囲値

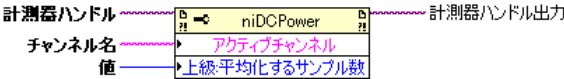
8. 「niDCPower 出力範囲を構成」 VI を使用して出力範囲を構成します。

LabVIEW VI	C/C++ 関数呼び出し
	以下のパラメータを使用して、「niDCPower_ConfigureOutputRange」を呼び出します。 vi: 「niDCPower_InitExtCal」からの計測器ハンドル channelName: x rangeType: NIDCPOWER_VAL_RANGE_CURRENT range: 表 13 に示したチャンネル x の反復に対応する電流範囲値

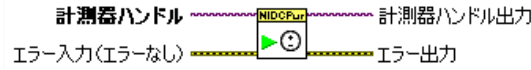
9. 「niDCPower 出力有効化を構成」VI を使用して、出力を有効にします。

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

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 attribute: NIDCPOWER_ATTR_SAMPLES_TO_AVERAGE channelName: x value: 300

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-4110 の出力が安定するまで 3 秒間待機します。
13. チャンネル x の反復に対応する出力値が 0 の場合、手順 14 に進みます。その他の出力値の場合は、DMM を使用して出力電流を測定します。
14. 測定値を記録します。チャンネル x の反復に対応する出力値が 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。



メモ

「niDCPower 測定」VI を使用して出力電流を測定します。出力値が 0 の場合に正確に測定するには、DMM/ 外部負荷と NI PXI-4110 間の接続が切断されていることを確認します。その他すべての出力値の場合は、DMM と外部負荷を NI PXI-4110 に接続します。

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

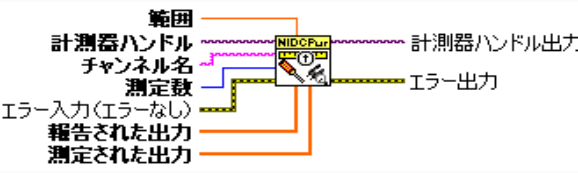
15. 測定値を記録します。
16. 表 13 に示した電流範囲で、チャンネル x の各反復に対して手順 4 ~ 15 を繰り返します。この電流範囲でチャンネル x のすべての反復を測定した後、手順 17 に進みます。
17. 「niDCPower 電圧レベルを構成」VI を使用して電圧レベルを構成します。

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: 0

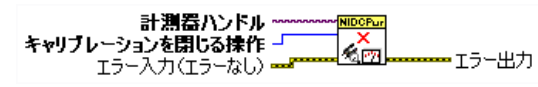
18. 「niDCPower 出力有効化を構成」VI を使用して、出力を無効にします。

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

19. 「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。

LabVIEW VI	C/C++ 関数呼び出し
	以下のパラメータを使用して、「niDCPower_CalAdjustCurrent 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 の電流範囲 numberOfMeasurements: 3 measuredOutputs: 手順 14 で記録した、チャンネル x の各反復に対する DMM での測定値の配列 reportedOutputs: 手順 15 で記録した、チャンネル x の各反復に対する NI PXI-4110 での測定値の配列

20. DMM との接続を切断します。
21. 表 13 に示したすべての未調整チャンネルに対し、手順 2 ～ 20 を繰り返します。すべてのチャンネルを調整すれば、この部分の調整は完了です。
22. 「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-4110 の調整が完了します。NI PXI-4110 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-4110 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 13 NI PXI-4110 出力パラメータ（電流測定精度の調整時）

チャンネル	電流範囲	反復	等価抵抗 (Ω)	出力 (V) *
0	1 A	1	オープン	0
		2	5	1.875
		3	5	3.75
1, 2	1 A	1	オープン	0
		2	20	±7.5
		3	20	±15
1, 2	20 mA	1	オープン	0
		2	1,000	±10
		3	1,000	±20

* チャンネル 1 調整時、出力は正の値。チャンネル 2 調整時、出力は負の値。

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-4110 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精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-4110 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、測定精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-4110 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 2 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。

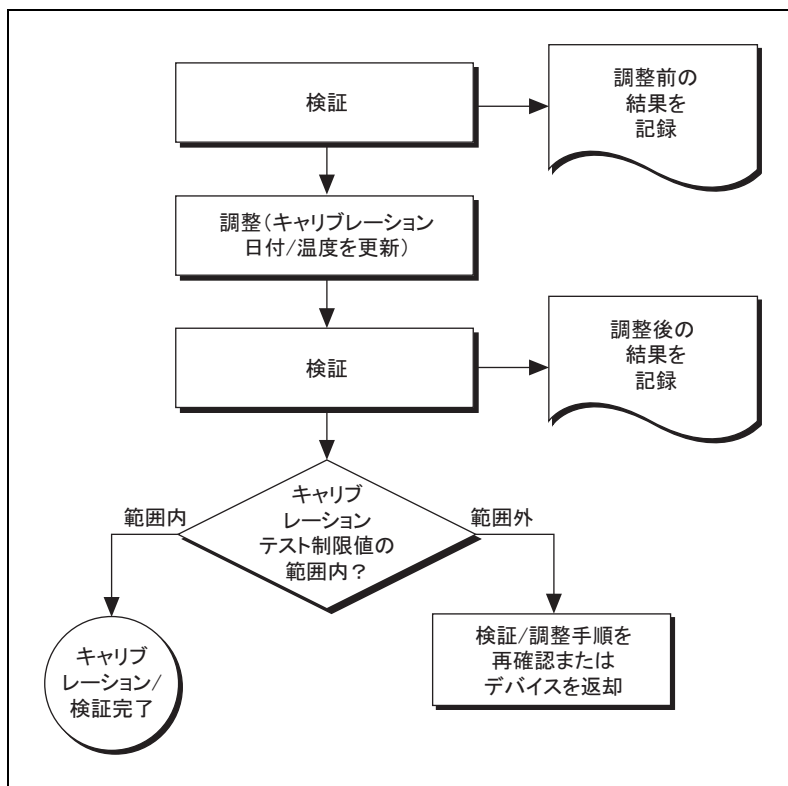


図 2 完全なキャリブレーションのプログラミングフロー

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-4110 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」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 3 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。

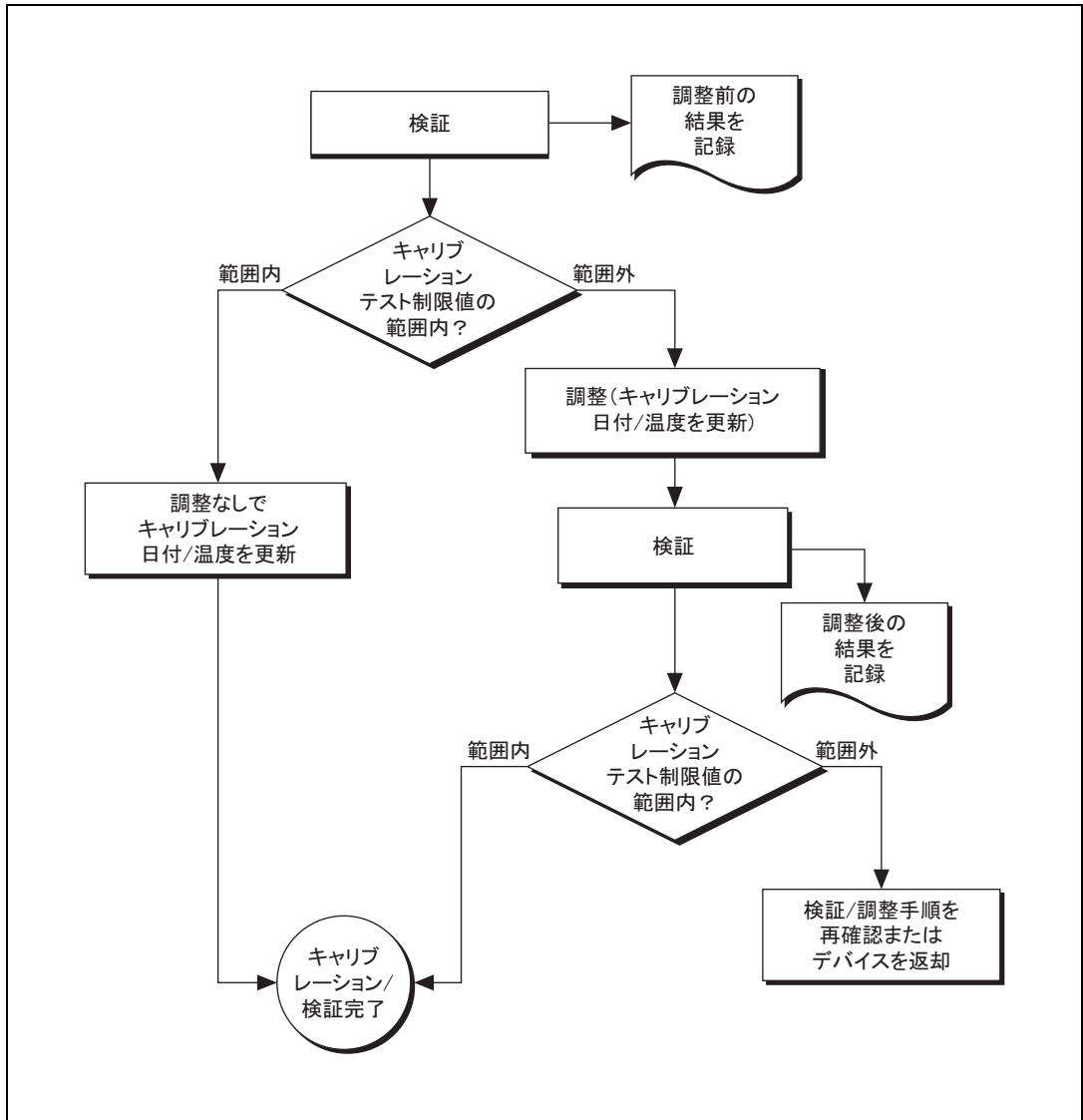


図 3 オプションのキャリブレーションのプログラミングフロー

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-4110 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、電磁適合性 (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、サービスリクエストを ni.com/jp/support で作成するか、0120-527196 (フリーダイヤル) 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

National Instruments, NI, ni.com, および LabVIEW は National Instruments Corporation (米国ナショナルインスツルメンツ社) の商標です。National Instruments の商標の詳細については、ni.com/legal の「Terms of Use」セクションを参照してください。本文中に記載されたその他の製品名および企業名は、それぞれの企業の商標または商号です。National Instruments の製品 / 技術を保護する特許については、ソフトウェアで参照できる特許情報 (**ヘルプ→特許情報**)、メディアに含まれている patents.txt ファイル、または「National Instruments Patent Notice」(ni.com/patents) のうち、該当するリソースから参照してください。