NI PCI/PXI-4472 CALIBRATION PROCEDURE

Contents

Introduction

This document contains information and step-by-step instructions for verifying and adjusting National Instruments PCI/PXI-4472 devices.

This document, which is intended for metrology laboratories or other facilities that maintain traceable standards, does not cover programming technique or compiler configuration because of the large number of programming languages and styles that can be used to perform calibration.

NATIONAL INSTRUMENTS

This section defines calibration, describes why it is necessary, and explains how frequently you should calibrate the PCI/PXI-4472 device. It also describes internal and external calibration.

What Is Calibration?

Calibration is the process of verifying the measurement accuracy of a device and adjusting for any measurement error. *Verification* consists of measuring the performance of a device and comparing the results to the factory specifications of the device. This document describes self-calibration (or internal calibration) and external calibration.

Why Should You Calibrate?

The accuracy of electronic components drifts with time and temperature, which can affect measurement accuracy as a device ages. Calibration restores the device or module to its specified accuracy and ensures that it still meets NI standards.

How Often Should You Calibrate?

The measurement accuracy requirements of your application determine how often you should calibrate the PCI/PXI-4472 device or module. NI recommends you perform a complete calibration at least once every year. You can shorten this interval to 90 days or six months based on the demands of your application.

Calibration Options: Self-Calibration Versus External Calibration

The PCI/PXI-4472 products have two calibration options: self-calibration and external calibration. Both calibration modes remove error from measurements.

Self-Calibration

Self-calibration is a simple calibration method that does not rely on external standards. In this method, the PCI/PXI-4472 calibration constants are adjusted with respect to a high-precision voltage source on the device. These new calibration constants are defined with respect to the calibration constants created during an external calibration, ensuring that the measurements can be traced back to the external standards. In essence, self-calibration is similar to the auto-zero function found on digital multimeters (DMM).

Self-calibration is performed after the device has been externally calibrated. However, external variables such as temperature may still affect measurements, necessitating external calibration.

External Calibration

External calibration requires the use of high-precision external standards. Metrology laboratories or other facilities that maintain traceable standards in a controlled environment must do external calibration. External standards supply and read a voltage from the PCI/PXI-4472 device. Adjustments are then made to the PCI/PXI-4472 device calibration constants to ensure that the reported voltages fall within specifications. The new calibration constants are then stored in the EEPROM, the non-volatile memory of the device. After the onboard calibration constants have been adjusted, the high-precision voltage source on the device is measured. After you have externally calibrated the device, subsequent internal calibration uses the measured value of the high-precision reference to maintain traceability. The procedures described in the *Calibration Process* section are for external calibration.

Equipment and Other Test Requirements

This section describes the equipment, test conditions, documentation, and software required for calibrating the PCI/PXI-4472.

You must properly configure the device to be calibrated by referring to the *Configuring the PCI/PXI-4472 Device in NI-DAQ* section.

Test Equipment

Table 1 lists the instruments recommended for calibrating the PCI/PXI-4472.

Table 1. PCI/PXI-4472 Calibration Equipment Specifications

Instrument	Critical Specifications	Suggested Model
Calibrator	Frequency Range: 20 Hz–45 kHz Voltage Range: 5 mV$_{rms}$–15 V$_{rms}$ AC Accuracy: ±0.1%[1], 20 Hz–95 kHz DC Accuracy: ±115 ppm[1] at 5 V	Fluke 5700A
Function Generator	Frequency Range: 20 Hz–45 kHz Frequency Accuracy: ±6 ppm[2] Voltage Range: 3 V$_{rms}$	HP 3324A

[1] AC accuracy is the sum of all errors, including percent of reading error and temperature error but excluding percent of range error.

[2] Frequency accuracy is the sum of all errors, including initial accuracy and stability errors.

If the recommended instrument is not available, select a substitute calibration standard that meets the given specifications.

Test Conditions

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

- Keep connections to the PCI/PXI-4472 short. Long cables and wires act as antennae, picking up extra noise that can affect measurements.

- Use a 50 Ω BNC coaxial cable for all connections to the device.

- Keep relative humidity between 10 and 90%, noncondensing, or consult the hardware user manual for the optimum relative humidity.

- Maintain the temperature between 5 and 40 °C, or consult the hardware user manual for the optimum temperature range.

- Allow a warm-up time of at least 15 minutes to ensure that the measurement circuitry of the PCI/PXI-4472 is at a stable operating temperature.

Documentation

This section describes the documentation you need to calibrate the PCI/PXI-4472. For detailed information on NI-DAQ, refer to the following documents:

- *NI-DAQ Function Reference Manual for PC Compatibles*
- *NI-DAQ User Manual for PC Compatibles*

The function reference manual includes information on the functions in NI-DAQ. The user manual contains instructions on installing and configuring DAQ devices and detailed information on creating applications that use NI-DAQ. These documents are the primary references for writing the verification utility needed for calibration. For further information on the products you are calibrating, refer to the device user manual at ni.com/manuals.

Software

Install NI-DAQ on the calibration computer. Because the PCI/PXI-4472 is a PC-based measurement device, you must have the proper device driver installed in the calibration system before calibrating the device.

Note　NI recommends that you install NI-DAQ driver software before physically installing the PCI/PXI-4472. For the PCI/PXI-4472 calibration procedure, you need NI-DAQ, version 6.9.1 or later, installed on the calibration system. NI-DAQ configures and controls the DAQ device and is available at ni.com/download.

The driver supports a number of programming languages, including LabVIEW, LabWindows/CVI, Microsoft Visual C ++, Microsoft Visual Basic and Borland C ++. When you install the driver, only install support for the programming language that you intend to use.

You also need a copy of the `niDsaCal.dll`, `niDsaCal.lib`, `niDsaCal.fp`, and `niDsaCal.h` files. You can find these files at `ni.com/support/dsa`. The `.dll` provides calibration functionality that does not reside in the standard NI-DAQ driver, including the ability to protect the calibration constants, update the calibration date, and write to the factory calibration area. The factory calibration area and the calibration date should only be modified by a metrology laboratory or another facility that maintains traceable standards. You can access the functions in this `.dll` with any 32-bit compiler.

You can write calibration software to communicate with the device through NI-DAQ function calls. For information on writing software, refer to the *Writing the Calibration Procedure* section.

Writing the Calibration Procedure

The calibration procedure in the *Calibration Process* section provides step-by-step instructions on calling the appropriate calibration functions. These calibration functions are C function calls from NI-DAQ and the calibration `.dll` and are also valid for Visual Basic programs. Although LabVIEW VIs are not discussed in this procedure, you can program in LabVIEW using the VIs with similar names to the NI-DAQ function calls in this procedure. Refer to the *Externally Adjusting the PCI/PXI-4472 Device* section for illustrations of the code used at each step of the calibration procedure.

Often you must follow a number of compiler-specific steps to create an application that uses NI-DAQ. The *NI-DAQ User Manual for PC Compatibles* details the required steps for each of the supported compilers.

✎ **Note** Many of the functions listed in the calibration procedure use variables that are defined in the `nidaqcns.h` file. To use these variables, you must include the `nidaqcns.h` file in the code. If you do not want to use these variable definitions, you can examine the function call listings in the NI-DAQ documentation and the `nidaqcns.h` file to determine what input values are required.

Configuring the PCI/PXI-4472 Device in NI-DAQ

NI-DAQ automatically detects the PCI/PXI-4472 device. However, to communicate with the PCI/PXI-4472, you must configure it in NI-DAQ. The following steps briefly explain how to configure the device in NI-DAQ. Refer to the device user manual for detailed installation instructions. You can install this manual when you install NI-DAQ.

1. Power off the computer that will hold the PCI/PXI-4472.

2. Install the device in an available slot.

3. Power on the computer and launch the Measurement & Automation Explorer (MAX) utility.

4. Configure the device number in MAX.

5. Click **Test Resources** to ensure that the device is properly working.

The PCI/PXI-4472 device is now configured.

Note After the PCI/PXI-4472 device is configured in MAX, the device is assigned a device number. Write this number down and use it to verify and calibrate the PCI/PXI-4472 device.

Calibration Process

To calibrate the PCI/PXI-4472 device, follow these main steps:

1. Verify the existing operation of the PCI/PXI-4472 device. This step, which is covered in the *Verifying the Operation of the PCI/PXI-4472* section, confirms whether the device is in specification prior to calibration.

2. Externally adjust the device calibration constants with respect to a known voltage source. This step is described in the *Externally Adjusting the PCI/PXI-4472 Device* section.

3. Verify again to ensure that the PCI/PXI-4472 device is operating within its specifications after adjustment.

Verifying the Operation of the PCI/PXI-4472

Verification determines how well the PCI/PXI-4472 device meets its specifications. By performing this procedure, you can see how the device has operated over time, which helps determine the appropriate calibration interval for your application.

Note You need the device number for some of the function calls. To get this number, refer to the *Configuring the PCI/PXI-4472 Device in NI-DAQ* section.

Verifying DC Offset

The following flowchart shows the programming code needed to verify the AI DC offset for the PCI/PXI-4472 device.

```
Determine the device channel from
the verification tables.
          │
          ▼
AI_Change_Parameter (deviceNumber,
    channel, ND_AI_COUPLING, ND_DC)
          │
          ▼
DAQ_Op (deviceNumber, channel, 0,
    binaryData, 131072, 102400)
          │
          ▼
DAQ_Vscale (deviceNumber, channel, 0,
1, 0, 131072, binaryData, voltageData)
          │
          ▼
DcOffset (voltageData, 131072, offset)
          │
          ▼
Record measurement.
          │
          ▼
Repeat entire procedure for each input channel.
```

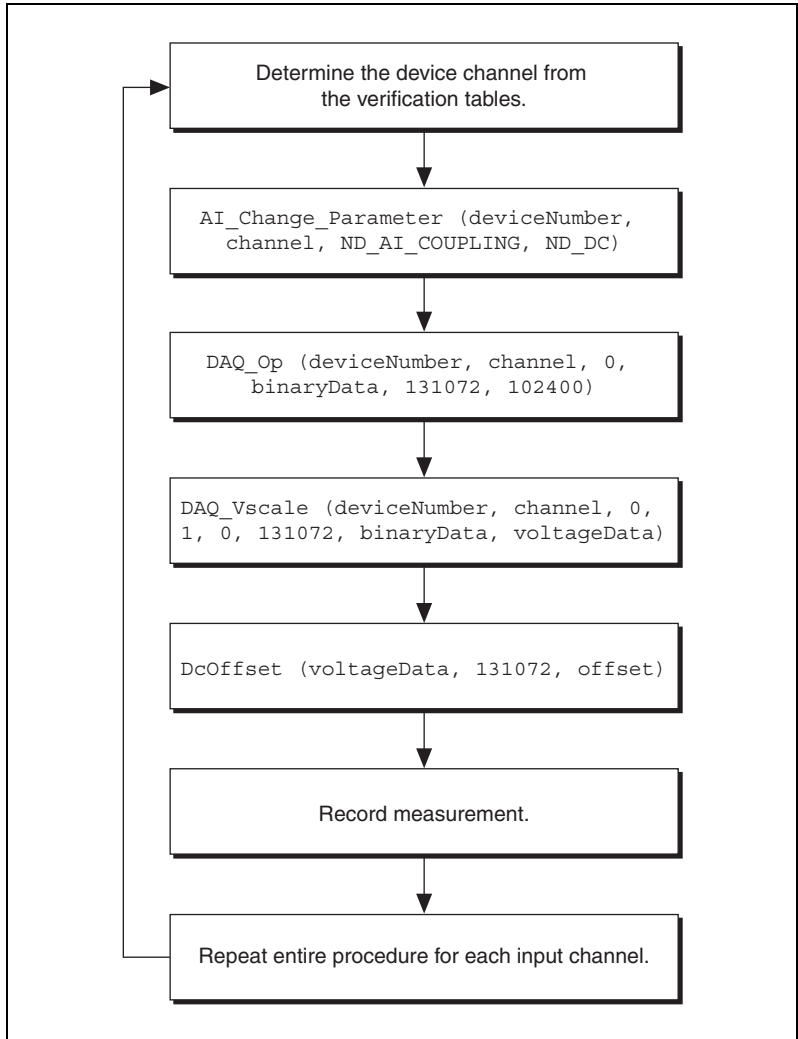**Figure 1.** Verifying AI DC Offset

Table 2. AI DC Offset for PCI/PXI-4472

Device Gain (dB)	Device Source	Device Input Amplitude		
		Min (mV)	**Actual (mV)**	**Max (mV)**
0	Terminated	–3		3

This procedure verifies the DC offset. To complete this procedure, refer to Table 2.

1. Ground input channel 0 with a 50 Ω terminator.

2. Call `AI_Change_Parameter` with the following parameters:

   - **channel**—`0`
   - **paramID**—`ND_AI_COUPLING`
   - **paramValue**—`ND_DC`

   Calling this function configures the channel for DC coupling.

3. Create an array called **binaryData** containing `131072` elements. Call `DAQ_Op` with the following parameters:

   - **channel**—`0`
   - **count**—`131072`
   - **sampleRate**—`102400`

   Calling this function acquires data from the PCI/PXI-4472 device.

4. Call `DAQ_VScale` with **channel** set to `0`. Create an array called **voltageData** containing `131072` elements. This step converts the array of binary data obtained from the `DAQ_OP` function in step 3 to voltage values.

5. Call `dcOffset` from `niDsaCal.dll`, and pass in the voltage data obtained from `DAQ_Vscale` with **scans** set to `131072`.

6. Compare the value returned with the minimum and maximum values listed in Table 2. If the value falls between these limits, the device has passed this portion of the DC offset test.

7. Repeat steps 1 through 6 for each input channel. Verify input channels sequentially (0, 1, 2, and so on).

You have now verified the AI DC offset.

Verifying Gain Accuracy

The flowchart in Figure 2 shows the programming code needed to verify the gain accuracy for the PCI/PXI-4472 device.
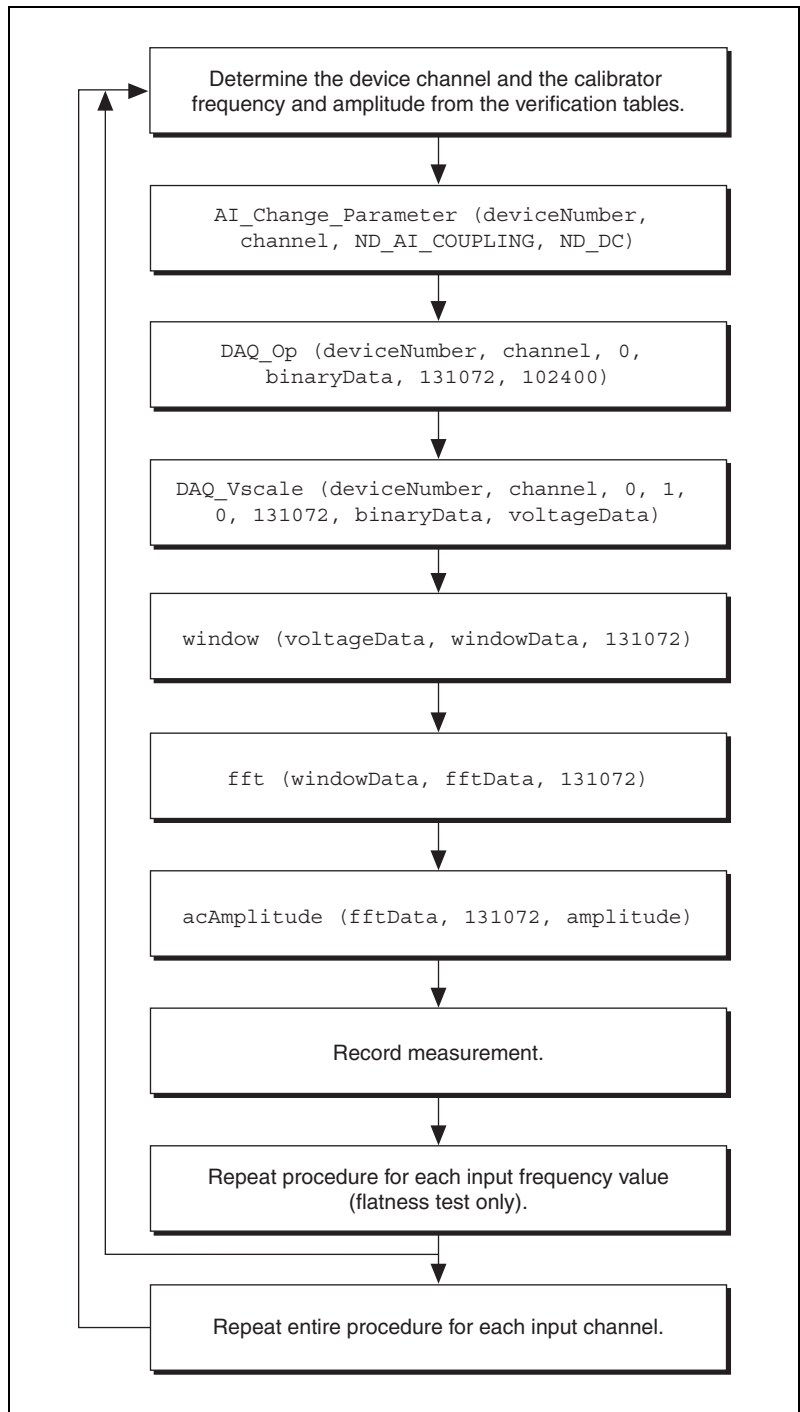
Figure 2. Verifying AI Gain Accuracy and Flatness

Table 3. AI Gain Accuracy

Device Gain (dB)	Calibrator Output		Device Input Amplitude		
	Amplitude (V_{rms})	Frequency (Hz)	Min (V_{rms})	Actual (V_{rms})	Max (V_{rms})
0	5	1000	4.943	A1:	5.058

This procedure verifies AI gain accuracy. To complete this procedure, refer to Table 3.

1. Connect the calibrator to input channel 0. Program the calibrator amplitude and frequency listed in Table 3.

2. Call `AI_Change_Parameter` with the following parameters:
   - **channel**—`0`
   - **paramID**—`ND_AI_COUPLING`
   - **paramValue**—`ND_DC`

   Calling this function configures the channel for DC coupling.

3. Create an array called **binaryData** containing `131072` elements. Call `DAQ_Op` with the following parameters:
   - **channel**—`0`
   - **count**—`131072`
   - **sampleRate**—`102400`

   Calling this function acquires data from the PCI/PXI-4472 device.

4. Call `DAQ_VScale` with **channel** set to `0`. Create an array called **voltageData** containing `131072` elements. This step converts the array of binary data obtained from the `DAQ_OP` function in step 3 to voltage values.

5. Call `window` from niDsaCal.dll. This function passes the data through a time-domain Hanning window. Pass in the voltage data obtained from `DAQ_VScale`. Set **scans** to `131072`, and create an array called **windowData** containing `131072` elements. When the function completes execution, **windowData** contains the **voltageData** array with a general cosine window applied to it.

6. Call `fft` from niDsaCal.dll. Pass the array of **windowData** to the function. Set **scans** to `131072`, and pass in an array called **fftData** containing `131072` elements. When the function completes execution, **fftData** contains the frequency spectrum of the acquired data.

7. Call `acAmplitude` from niDsaCal.dll. Pass in the **fftData** obtained from the `fft` function. Set **scans** to `131072`. This function returns the amplitude of the fundamental harmonic. Record this value in Table 3.

8. Repeat steps 1 through 7 for each input channel. Verify input channels sequentially (0, 1, 2, and so on).

After you verify each gain setting for all the channels, you have verified the AI gain accuracy for the device.

Verifying Flatness

The flowchart in Figure 2 shows the programming code needed to verify the AI flatness for the PCI/PXI-4472 device.

Table 4. AI Flatness

Device Gain (dB)	Calibrator Output		Device Input Amplitude		
	Amplitude (V_{rms})	Frequency (Hz)	Min (V_{rms})	Actual (V_{rms})	Max (V_{rms})
0	5	20	4.9827 V% A1		5.0173 V% A1
0	5	2,000	4.9827 V% A1		5.0173 V% A1
0	5	20,000	4.9827 V% A1		5.0173 V% A1
0	5	45,000	4.945 V% A1		5.058 V% A1

This procedure verifies AI flatness. To complete this procedure, refer to Table 4.

1. Connect the calibrator to input channel 0. Program the calibrator amplitude and frequency listed in Table 4.

✏️ **Note** Use the first row in Table 4 initially. When you repeat this procedure, use the next row (2, 3, 4, and so on).

2. Call `AI_Change_Parameter` with the following parameters:
    - **channel**—`0`
    - **paramID**—`ND_AI_COUPLING`
    - **paramValue**—`ND_DC`

    Calling this function configures the channel for DC coupling.

3. Create an array called **binaryData** containing `131072` elements. Call `DAQ_Op` with the following parameters:
    - **channel**—`0`
    - **count**—`131072`
    - **sampleRate**—`102400`

    This function acquires data from the PCI/PXI-4472 device.

4. Call `DAQ_VScale` with **channel** set to `0`. Create an array called **voltageData** containing `131072` elements, and set **binaryZeroOffset** to `0`. This step converts the array of binary data obtained from the `DAQ_OP` function in step 3 to voltage values.

5. Call `window` from `niDsaCal.dll`. Pass in the voltage data obtained from `DAQ_VScale`. Set **scans** to `131072`, and create an array called **windowData** containing `131072` elements. When the function completes execution, **windowData** contains the **voltageData** array with a general cosine window applied to it.

6. Call `fft` from `niDsaCal.dll`. Pass the array of **windowData** to the function. Set **scans** to `131072`, and pass in an array called **fftData** containing `131072` elements. When the function completes execution, **fftData** contains the frequency spectrum of the acquired data.

7. Call `acAmplitude` from `niDsaCal.dll`. Pass in the **fftData** obtained from the `fft` function. Set **scans** to `131072`. This function returns the amplitude of the fundamental harmonic. Record this value in Table 4. If this value falls between the minimum and maximum values, the device has passed this portion of the flatness test. The minimum and maximum values are the percentages multiplied by the corresponding actual value recorded in Table 4 for the specified gain.

8. Repeat steps 1 through 7 to verify each row in Table 4 (2, 3, 4, and so on).

9. Repeat steps 1 through 8 for each input channel. Verify input channels sequentially (0, 1, 2, and so on).

After you verify each gain setting for all the channels, you have verified the AI flatness for the PCI/PXI-4472 device.

Verifying AI Frequency Accuracy

The flowchart in Figure 3 shows the programming code needed to verify the AI frequency accuracy for the PCI/PXI-4472 device.
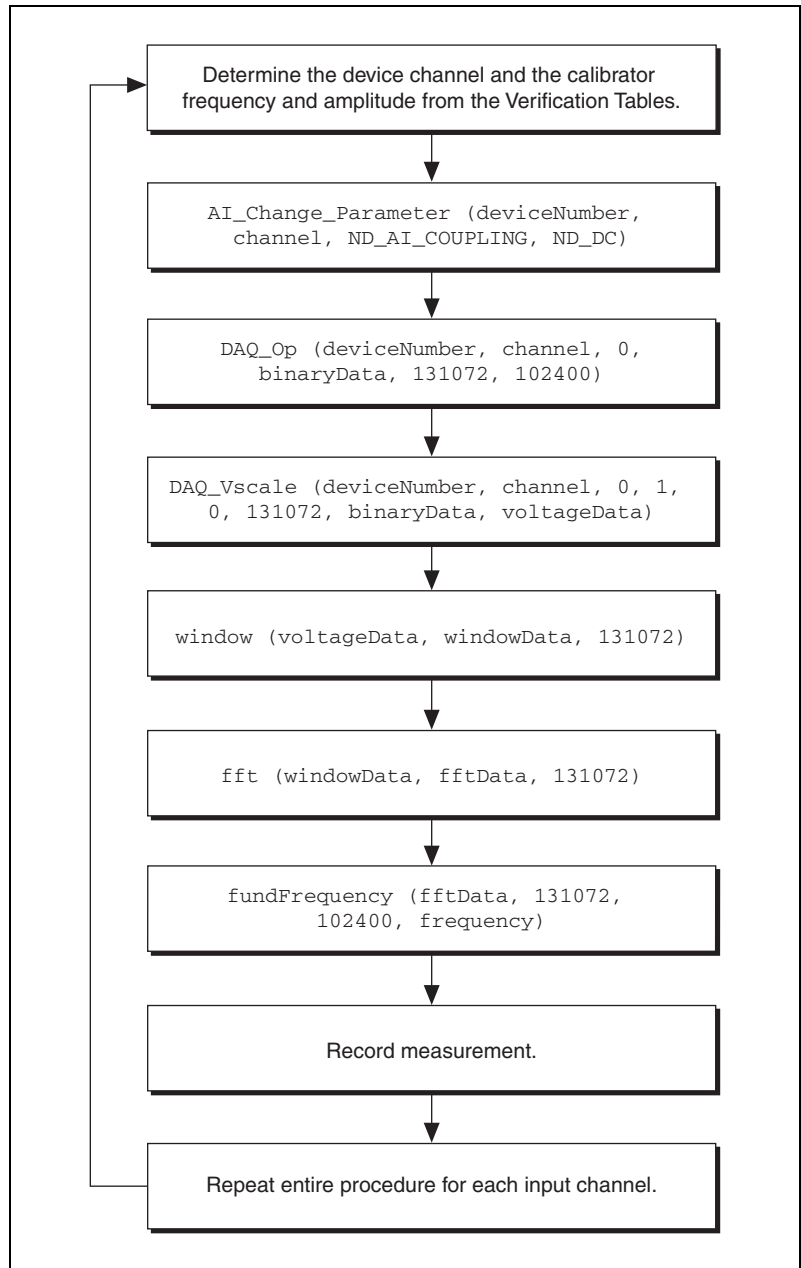
```
Determine the device channel and the calibrator
frequency and amplitude from the Verification Tables.
                        │
                        ▼
AI_Change_Parameter (deviceNumber,
    channel, ND_AI_COUPLING, ND_DC)
                        │
                        ▼
DAQ_Op (deviceNumber, channel, 0,
    binaryData, 131072, 102400)
                        │
                        ▼
DAQ_Vscale (deviceNumber, channel, 0, 1,
    0, 131072, binaryData, voltageData)
                        │
                        ▼
window (voltageData, windowData, 131072)
                        │
                        ▼
fft (windowData, fftData, 131072)
                        │
                        ▼
fundFrequency (fftData, 131072,
        102400, frequency)
                        │
                        ▼
Record measurement.
                        │
                        ▼
Repeat entire procedure for each input channel.
```

Figure 3. Verifying AI Frequency Accuracy

Table 5. AI Frequency Accuracy

Device Gain (dB)	Function Generator Output		Device Input Frequency		
	Amplitude (V_{rms})	Frequency (Hz)	Min (Hz)	Actual (Hz)	Max (Hz)
0	2.5	20	19.9995		20.0005
0	2.5	2,000	1,999.95		2,000.05
0	2.5	20,000	1,999.95		20,000.5
0	2.5	45,000	44,998.88		45,001.12

This procedure verifies AI frequency accuracy. To complete this procedure, refer to Table 5.

1. Connect the function generator to input channel 0 on the BNC-2140. Program the function generator amplitude and frequency listed in Table 5.

✏️ **Note** Use the first row in Table 5 initially. When you repeat this procedure, use the next row (2, 3, 4, and so on).

2. Call `AI_Change_Parameter` with the following parameters:
   - **channel**—`0`
   - **paramID**—`ND_AI_COUPLING`
   - **paramValue**—`ND_DC`

   This function configures the channel for DC coupling.

3. Call `DAQ_Op` with **channel** set to `0`. Create an array called **binaryData** containing `131072` elements with the following parameters:
   - **count**—`131072`
   - **sampleRate**—`102400`

   This function acquires data from the PCI/PXI-4472 device.

4. Call `DAQ_VScale` with **channel** set to `0`.

5. Create an array called **voltageData** containing `131072` elements, and set **binaryZeroOffset** to `0`. This step converts the array of binary data obtained from the `DAQ_OP` function in step 3 to voltage values.

6. Call `window` from `niDsaCal.dll`. Pass in the voltage data obtained from `DAQ_VScale`. Set **scans** to `131072`, and create an array called **windowData** containing `131072` elements. When the function completes execution, **windowData** contains the **voltageData** array with a general cosine window applied to it.

7. Call fft from the niDsaCAl.dll. Pass the array of **windowData** to the function. Set **scans** to 131072, and pass in an array called **fftData** containing 131072 elements. When the function completes execution, **fftData** contains the frequency spectrum of the acquired data.

8. Call fundFrequency from the niDsaCal.dll. Pass in the **fftData** obtained from the fft function. Set the following parameters:

   - **scans**—131072
   - **sampleRate**—102400

   Pass in a pointer to the frequency. This function returns the frequency of the fundamental harmonic. Record this value in Table 6. If this value falls between the minimum and maximum values, the device has passed this portion of the frequency accuracy test.

9. Repeat steps 1 through 8 to verify each row in Table 5 (2, 3, 4, and so on).

10. Repeat steps 1 through 8 for each input channel. Verify input channels sequentially (0, 1, 2, and so on).

After you verify each gain setting for all the channels, you have verified the AI frequency accuracy for the device.

Externally Adjusting the PCI/PXI-4472 Device

The PCI/PXI-4472 adjustment procedure adjusts the AI calibration constants. At the end of each adjustment procedure, these new constants are stored in the factory area of the device EEPROM, which provides a level of security to ensure that you do not accidentally access or modify any calibration constants adjusted by the metrology laboratory.
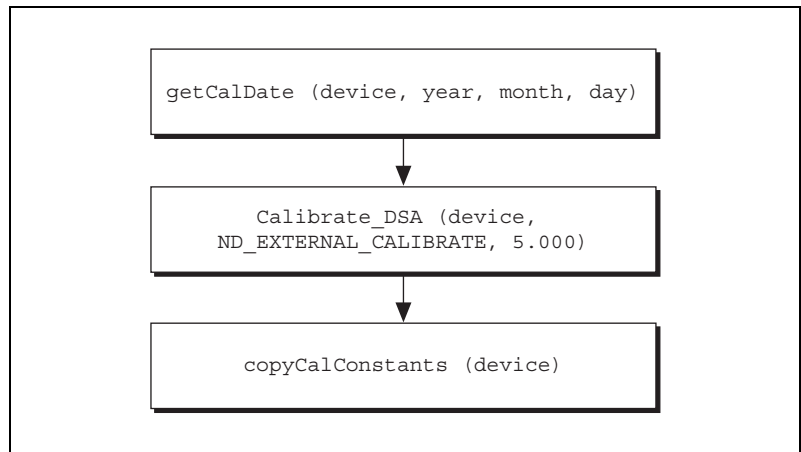

Figure 4. Externally Adjusting the Device

To adjust the device, complete the following steps:

1. Connect the output of the calibrator to all AI channels using eight T-connectors.

2. Set the calibrator to output a voltage of 5.000 V.

3. To find out the date of the last calibration, run the getCalDate function from the calibration library.

4. Call Calibrate_DSA with the following parameters:

   - **calOP**—ND_EXTERNAL_CALIBRATE
   - **setOfCalConst**—ND_USER_EEPROM_AREA
   - **calRefVolts**—5.000

   If the voltage supplied by the calibrator does not maintain a steady 5.000 V, an error is returned.

5. Call copyCalConst to copy the calibration information stored in the user area of the EEPROM to the factory area of the EEPROM.

✎ **Note** After the device is adjusted, you may want to verify the AI and AO operation. To do this, repeat the procedures in the *Verifying the Operation of the PCI/PXI-4472* section.

Specifications

This section lists the specifications for the NI PCI/PXI-4472 devices.

Input

Channels 0, 1, 2, 3, 4, 5, 6, 7

Gain ranges .... 0 dB

Maximum sample rate ....102.4 kS/s

Offset (residual DC) ....±3 mV

Gain (amplitude accuracy).....................±0.1 dB, f_{in} = 1 kHz

Flatness (referenced to 1 kHz)...............±0.03 dB, f_{in} = 0–20 kHz;
±0.1 dB, f_{in} = 20–45 kHz

Frequency accuracy ....±25 ppm

Output

None

Counter

None