NI 447*X* Calibration Procedure

This document contains information about calibrating National Instruments PCI/PXI-447*X* devices.

This document does not discuss programming techniques or compiler configuration. The National Instruments DAQmx driver includes help files that have compiler-specific instructions and detailed function explanations. You can install these help files when you install NI-DAQmx on the calibration computer.

Contents

Conventions

The following conventions appear in this document:

» — 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 and hardware labels.

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

`monospace` — Monospace text 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.

NI 447*X* — NI 447*X* refers to the NI PCI/PXI-4472 and NI PCI/PXI-4474 unless otherwise noted.

Number Variable Types — The integer variable type used in this document is represented as a whole number. Any floating point variable type is represented as a number with a decimal point. For example, the number 1,024 is written as `1024` if it is an integer, and as `1024.0` if it is a floating point value.

ReturnValue — A parameter which points to data that is modified by a function call has a variable name in the form <variableTypeReturnValue>, for example, `uInt32PointerReturnValue`. All function calls return a status integer that you must verify. Because all the functions return this status integer, the function descriptions in this document omit the status integer.

Software

Calibration requires the latest NI-DAQmx driver. NI-DAQmx includes high-level function calls to simplify the task of writing software to calibrate devices. The driver supports many programming languages, including LabVIEW, LabWindows™/CVI™, C/C++, C#, and Visual Basic .NET.

If you want to write the calibration procedure using LabVIEW, NI recommends that you use the Full or Professional Development System or the Developer Suite Professional Edition. These editions include frequency-domain analysis tools not included in the Base Development System.

Documentation

If you are using the NI-DAQmx driver, the following documents are your primary references for writing your calibration utility:

- The *NI-DAQmx C Reference Help* includes information about the functions in the driver.
- The *DAQ Getting Started Guide* for NI-DAQ 8.0 or later provides instructions for installing and configuring NI-DAQ devices.
- The *NI-DAQmx Help* includes information about creating applications that use the NI-DAQmx driver.
- *NI 447X Specifications* contains information about specific device features. The limits you use to verify the accuracy of the devices are based on the specifications found in this document.

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

Calibration Interval

NI 447*X* devices should be externally calibrated at a regular interval as defined by the measurement accuracy requirements of your application. NI recommends that you perform a complete calibration at least once every year. Based on your measurement accuracy needs, you can shorten this interval to 90 days or six months. Self-calibration can be performed as needed or when the temperature varies by 5 °C or more from the last external calibration.

Password

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

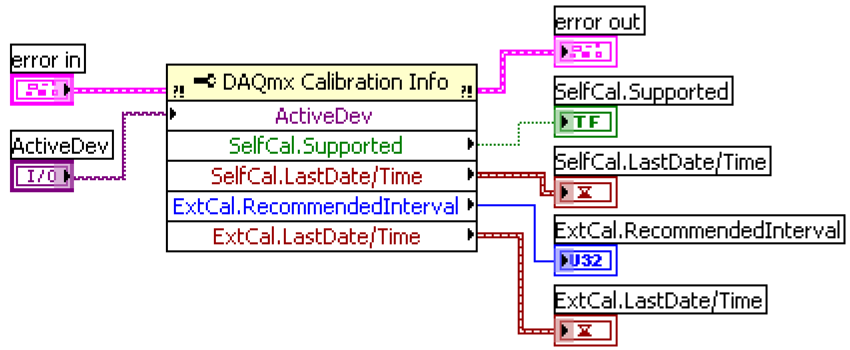
Getting Calibration Information

The NI 447*X* devices contain the following calibration information stored in the EEPROM:

- Whether the device supports self-calibration
- Date and time of the last self-calibration
- Recommended interval for external calibration
- Date and time of the last external calibration

You can access this information in the Measurement & Automation Explorer (MAX) or by using the following LabVIEW VIs.

Note Refer to the NI-DAQmx function parameters for the LabVIEW input values.

LabVIEW Block Diagram

NI-DAQmx Function Call
Call `DAQmxGetSelfCalSupported` with the following parameters: **deviceName**: `dev1` **data**: `bool32PointerReturnValue`
Call `DAQmxGetSelfCalLastDateAndTime` with the following parameters: **deviceName**: `dev1` **year**: `uInt32PointerReturnValue` **month**: `uInt32PointerReturnValue` **day**: `uInt32PointerReturnValue` **hour**: `uInt32PointerReturnValue` **minute**: `uInt32PointerReturnValue`
Call `DAQmxGetExtCalRecommendedInterval` with the following parameters: **deviceName**: `dev1` **data**: `uInt32PointerReturnValue`

Call `DAQmxGetExtCalLastDateAndTime` with the following parameters: **deviceName**: `dev1` **year**: `uInt32PointerReturnValue` **month**: `uInt32PointerReturnValue` **day**: `uInt32PointerReturnValue` **hour**: `uInt32PointerReturnValue` **minute**: `uInt32PointerReturnValue`
Call `DAQmxGetExtCalLastTemp` with the following parameters: **deviceName**: `dev1` **data**: `float64PointerReturnValue`
Call `DAQmxGetCalDevTemp` with the following parameters: **deviceName**: `dev1` **data**: `float64PointerReturnValue`

Test Equipment

Table 1 lists the instruments NI recommends for calibrating the NI 447*X*.

Table 1. NI 447*X* Calibration Equipment Specifications

Instrument	Critical Specifications	Suggested Model
Calibrator	Frequency Range: 20 Hz–45 kHz Voltage Range: up to 5 V_{rms} AC Accuracy: ±0.1%*, 20 Hz–95 kHz DC Accuracy: ±115 ppm* at 5 V	Fluke 5700A
Function Generator	Frequency Range: up to 10 kHz Frequency Accuracy: ±2 ppm† Voltage Range: up to 9 V_{pp}	HP/Agilent 33250A

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

† 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 and adjustment:

- Keep connections to the NI 447*X* short. Long cables and wires act as antennae, picking up extra noise that can affect measurements.
- Use SMB coaxial cable for all connections to the device.
- Keep relative humidity between 10 and 80%, noncondensing, or consult the device documentation for the optimum relative humidity.
- Maintain the temperature between 18 and 28 °C, or refer to the device specifications for the optimum temperature range.
- Allow a warm-up time of at least 15 minutes to ensure that the measurement circuitry of the NI 447*X* is at a stable operating temperature.
- Allow a warm-up time for all the instruments and equipment, according to the manufacturers' instructions.

Calibration Procedure

This section provides instructions for verifying and calibrating the NI 447*X* device.

The calibration procedure consists of the following steps:

1. Initial Setup—Configure the device in NI-DAQmx.
2. Self-Calibration—Measure the onboard reference voltage of the device and adjust the self-calibration constants.
3. NI 447*X* Verification Procedure—Verify the accuracy of the device prior to calibration.
4. NI 447*X* Adjustment Procedure—Perform an external calibration that adjusts the device calibration constants with respect to known voltage sources.
5. Perform another verification to ensure that the device operates with the specifications after adjustment.

The initial setup, self-calibration, verification, and adjustment procedures are explained in the following sections.

Note You must compare the verification limits provided in this procedure with the most recent specifications. Refer to the latest *NI 447X Specifications* document at `ni.com/manuals`.

Initial Setup

NI-DAQmx automatically detects all NI 447*X* devices. However, for the driver to communicate with the device, you must configure the device in NI-DAQmx.

Complete the following steps to configure a device in NI-DAQmx:

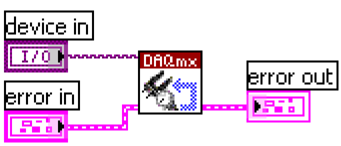
1. Install the NI-DAQmx driver software.
2. Power off the computer that will hold the device and install the device in an available slot.
3. Power on the computer and launch MAX.
4. Right-click the device name and select **Self-Test** to ensure that the device is working properly.

Note When a device is configured with MAX, it is assigned a device name. Each function call uses this device name to determine which DAQ device to calibrate. This document uses `dev1` to refer to the device name. In the following procedures, use the device name as it appears in MAX.

Self-Calibration

Call self-calibration before completing the first verification. The DAQmx Self Calibrate VI, as shown below, measures the onboard reference voltage of the device and adjusts the self-calibration constants to account for any errors caused by short-term fluctuations in the operating environment. When you self-calibrate a device, no external signal connections are necessary.

Note Refer to the NI-DAQmx function parameters for the LabVIEW input values.

<table>
<tr><th>LabVIEW Block Diagram</th><th>NI-DAQmx Function Call</th></tr>
<tr><td>
</td><td>Call <code>DAQmxSelfCal</code> with the following parameters:

deviceName: <code>dev1</code></td></tr>
</table>

Verification Procedure

Verification determines how well the NI 447*X* device is meeting specifications. By completing this procedure, you can see how the device has drifted over time, which helps you determine the appropriate calibration interval for your application.

Analog Input Performance Verification

This section verifies the analog input (AI) performance of all NI 447*X* devices. Refer to the *NI 447X Specifications* for the number of channels and performance specifications.

AI DC Offset Verification

Complete the following steps to verify the AI DC offset:

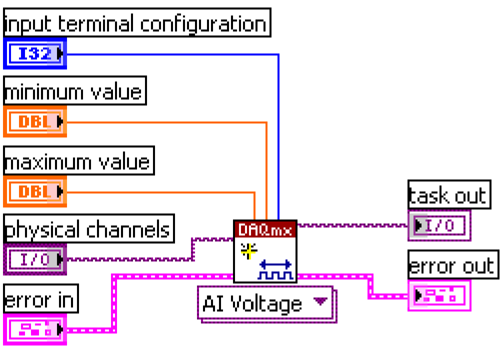
1. Connect a BNC shorting cap to the analog input channel 0 of the device. The terminator grounds the input channel because it is in pseudodifferential configuration.
2. If you use LabVIEW, skip this step. If you use C function calls, create a task using `DAQmxCreateTask`, as shown below. The task is created in step 3 in LabVIEW.

LabVIEW Block Diagram	NI-DAQmx Function Call
LabVIEW does not require this step.	Call `DAQmxCreateTask` with the following parameter: **taskHandle**: `mytaskHandleReturnValue`

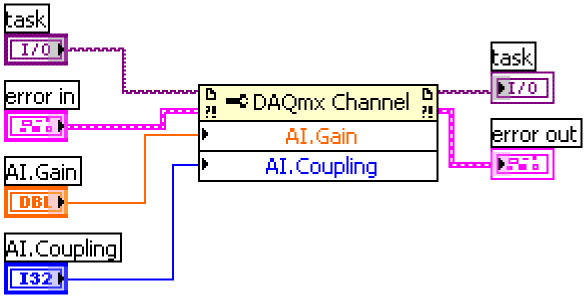
   You use the object `myTaskHandle` in all subsequent NI-DAQmx function calls.

3. Create an AI voltage channel through NI-DAQmx using the DAQmx Create Virtual Channel VI, as shown below.

Note Refer to the NI-DAQmx function parameters for the LabVIEW input values.

LabVIEW Block Diagram	NI-DAQmx Function Call
	Call `DAQmxCreateAIVoltageChan` with the following parameters: **taskHandle**: `myTaskHandle` **physicalChannel**: `dev1/ai0` **nameToAssignToChannel**: `""` **terminalConfig**: `DAQmx_Val_PseudoDiff` **minVal**: `-10.0` **maxVal**: `10.0` **units**: `DAQmx_Val_Volts` **CustomScaleName**: `NULL`

4. Modify the AI voltage channel properties using the following functions.

<table>
<tr><th>LabVIEW Block Diagram</th><th>NI-DAQmx Function Call</th></tr>
<tr><td rowspan="2">
</td><td>Call <code>DAQmxSetAIGain</code> with the following parameters:

taskHandle: <code>myTaskHandle</code>

channel: <code>dev1/ai0</code>

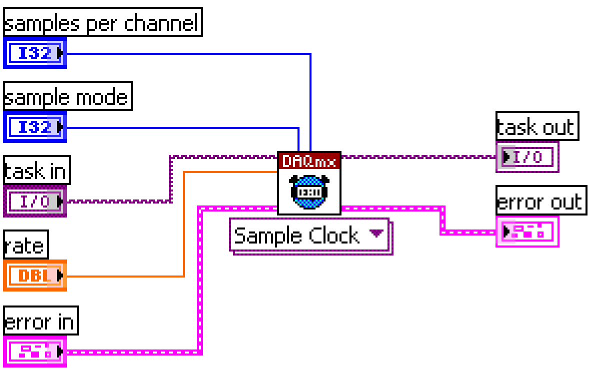
data: <code>0.0</code></td></tr>
<tr><td>Call <code>DAQmxSetAICoupling</code> with the following parameters:

taskHandle: <code>myTaskHandle</code>

channel: <code>dev1/ai0</code>

data: <code>DAQmx_Val_DC</code></td></tr>
</table>

5. Configure the timing properties of the acquisition using the DAQmx Timing VI, as shown below.

<table>
<tr><th>LabVIEW Block Diagram</th><th>NI-DAQmx Function Call</th></tr>
<tr><td>
</td><td>Call <code>DAQmxCfgSampClkTiming</code> with the following parameters:

taskHandle: <code>myTaskHandle</code>

source: <code>NULL</code>

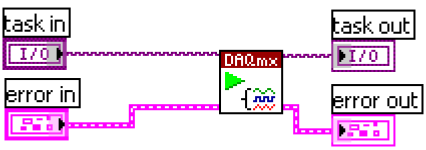
rate: <code>102400.0</code>

activeEdge: <code>DAQmx_Val_Rising</code>

sampleMode: <code>DAQmx_Val_FiniteSamps</code>

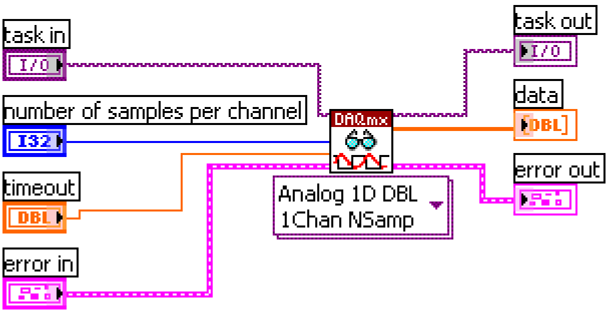
sampsPerChanToAcquire: <code>51200</code></td></tr>
</table>

6. Start the task using the DAQmx Start Task VI, as shown below.

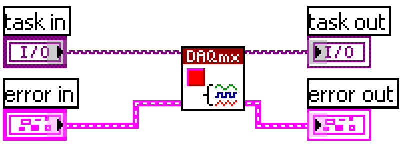
<table>
<tr><th>LabVIEW Block Diagram</th><th>NI-DAQmx Function Call</th></tr>
<tr><td>
</td><td>Call <code>DAQmxStartTask</code> with the following parameters:

taskHandle: <code>myTaskHandle</code></td></tr>
</table>

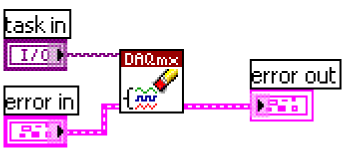
7. Acquire data using the DAQmx Read VI, as shown below.

LabVIEW Block Diagram	NI-DAQmx Function Call
	Call `DAQmxReadAnalogF64` with the following parameters: **taskHandle**: `myTaskHandle` **numSampsPerChan**: `-1` **timeout**: `10.0` **fillMode**: `DAQmx_Val_GroupByChannel` **arraySizeInSamps**: `51200` **reserved**: `NULL` **readArray**: `float64ArrayReturnValue` **sampsPerChanRead**: `int32PointerReturnValue`

8. Stop the task using the DAQmx Stop Task VI, as shown below.

LabVIEW Block Diagram	NI-DAQmx Function Call
	Call `DAQmxStopTask` with the following parameters: **taskHandle**: `myTaskHandle`

9. Clear the task using the DAQmx Clear Task VI, as shown below.

LabVIEW Block Diagram	NI-DAQmx Function Call
	Call `DAQmxClearTask` with the following parameters: **taskHandle**: `myTaskHandle`

10. Average all the values in the array returned from the acquisition. The average is the resulting offset for AI 0. Compare this value to the limits in Table 2.

Table 2. AI Offset Limits

Device Input Offset	
Min (mV)	**Max (mV)**
–3.0	3.0

11. Repeat steps 1 through 10 for all remaining analog input channels of the device. Replace `ai0` with the appropriate channel name in function call parameters `physicalChannel` and `channel`.

AI Gain Accuracy Verification

Complete the following steps to verify AI gain accuracy:

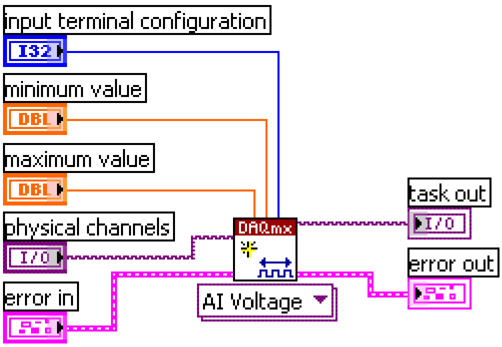
1. Connect the output of the calibrator to the analog input channel 0 of the device.
2. Program the calibrator amplitude and frequency as listed in Table 3.
3. If you use LabVIEW, skip this step. If you use C function calls, create a task using `DAQmxCreateTask`, as shown below. The task is created in step 4 in LabVIEW.

LabVIEW Block Diagram	NI-DAQmx Function Call
LabVIEW does not require this step.	Call `DAQmxCreateTask` with the following parameter: **taskHandle**: `mytaskHandleReturnValue`

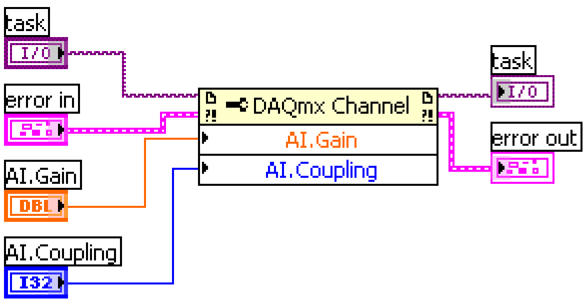
You use the object `myTaskHandle` in all subsequent NI-DAQmx function calls.

4. Create an AI voltage channel through NI-DAQmx using the DAQmx Create Virtual Channel VI, as shown below.

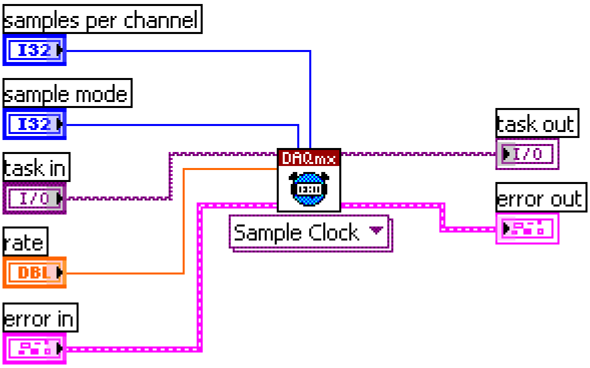
Note Refer to the NI-DAQmx function parameters for the LabVIEW input values.

LabVIEW Block Diagram	NI-DAQmx Function Call
	Call `DAQmxCreateAIVoltageChan` with the following parameters: **taskHandle**: `myTaskHandle` **physicalChannel**: `dev1/ai0` **nameToAssignToChannel**: `""` **terminalConfig**: `DAQmx_Val_PseudoDiff` **minVal**: `-10.0` **maxVal**: `10.0` **units**: `DAQmx_Val_Volts` **CustomScaleName**: `NULL`

5. Modify the AI voltage channel properties using the following functions.

LabVIEW Block Diagram	NI-DAQmx Function Call
	Call `DAQmxSetAIGain` with the following parameters: **taskHandle**: `myTaskHandle` **channel**: `dev1/ai0` **data**: `0.0`
	Call `DAQmxSetAICoupling` with the following parameters: **taskHandle**: `myTaskHandle` **channel**: `dev1/ai0` **data**: `DAQmx_Val_DC`

6. Configure the timing properties of the acquisition using the DAQmx Timing VI, as shown below.

<table>
<tr><th>LabVIEW Block Diagram</th><th>NI-DAQmx Function Call</th></tr>
<tr><td>
</td><td>Call <code>DAQmxCfgSampClkTiming</code> with the following parameters:

taskHandle: <code>myTaskHandle</code>

source: <code>NULL</code>

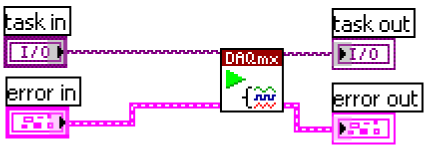
rate: <code>102400.0</code>

activeEdge: <code>DAQmx_Val_Rising</code>

sampleMode: <code>DAQmx_Val_FiniteSamps</code>

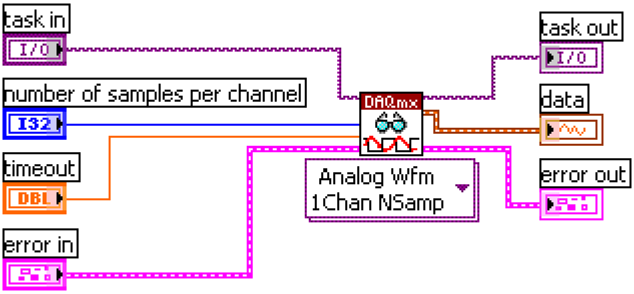
sampsPerChanToAcquire: <code>51200</code></td></tr>
</table>

7. Start the task using the DAQmx Start Task VI, as shown below.

<table>
<tr><th>LabVIEW Block Diagram</th><th>NI-DAQmx Function Call</th></tr>
<tr><td>
</td><td>Call <code>DAQmxStartTask</code> with the following parameters:

taskHandle: <code>myTaskHandle</code></td></tr>
</table>

8. Acquire data using the DAQmx Read VI, as shown below.

<table>
<tr><th>LabVIEW Block Diagram</th><th>NI-DAQmx Function Call</th></tr>
<tr><td>
</td><td>Call <code>DAQmxReadAnalogF64</code> with the following parameters:

taskHandle: <code>myTaskHandle</code>

numSampsPerChan: <code>-1</code>

timeout: <code>10.0</code>

fillMode: <code>DAQmx_Val_GroupByChannel</code>

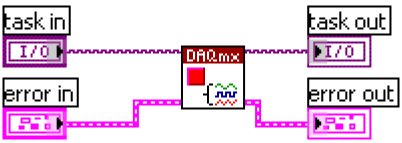
arraySizeInSamps: <code>102400</code>

reserved: <code>NULL</code>

readArray: <code>float64ArrayReturnValue</code>

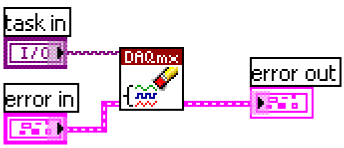
sampsPerChanRead: <code>int32PointerReturnValue</code></td></tr>
</table>

9. Stop the task using the DAQmx Stop Task VI, as shown below.

<table>
<tr><th>LabVIEW Block Diagram</th><th>NI-DAQmx Function Call</th></tr>
<tr><td>
</td><td>Call <code>DAQmxStopTask</code> with the following parameters:

taskHandle: <code>myTaskHandle</code></td></tr>
</table>

10. Clear the task using the DAQmx Clear Task VI, as shown below.

<table>
<tr><th>LabVIEW Block Diagram</th><th>NI-DAQmx Function Call</th></tr>
<tr><td>
</td><td>Call <code>DAQmxClearTask</code> with the following parameters:

taskHandle: <code>myTaskHandle</code></td></tr>
</table>

11. If you use LabVIEW, use the Analog 1D Wfm 1chan Nsamp polymorphic VI to acquire data in step 8. Measure the RMS value of the input using the Amplitude and Level Measurements VI. If you use C function calls, perform an FFT on the array of data and find the RMS amplitude of the fundamental harmonic.

12. Compare this RMS value to the limits (V) in Table 3. You will use this value as *x* in Table 4.

Table 3. AI Gain Accuracy Limits

Calibrator Output		Device Input Amplitude	
Amplitude (V_{rms})	**Frequency (Hz)**	**Min (V_{rms})**	**Max (V_{rms})**
5	1000	4.943	5.058

13. Repeat steps 1 through 12 for all remaining analog input channels of the device. Replace `ai0` with the appropriate channel name in function call parameters `physicalChannel` and `channel`.

AI Flatness Verification

Complete the following steps to verify AI flatness:

1. Connect the output of the calibrator to the analog input channel 0 of the device.
2. Program the calibrator amplitude and frequency as listed in Table 4.
3. Repeat steps 3 through 11 of the *AI Gain Accuracy Verification* section and compare the RMS value to the limits (V) in Table 4. The minimum and maximum values are the fractions multiplied by the corresponding measured value, *x*, you determined in the *AI Gain Accuracy Verification* section.
4. Repeat steps 2 and 3 of this section for each row in Table 4.

Table 4. AI Flatness Limits

Calibrator Output		Device Input Amplitude	
Amplitude (V_{rms})	**Frequency (Hz)**	**Min (V_{rms})**	**Max (V_{rms})**
5	20	0.99655 * *x*	1.00346 * *x*
5	2,000	0.99655 * *x*	1.00346 * *x*
5	20,000	0.99655 * *x*	1.00346 * *x*
5	45,000	0.98855 * *x*	1.01158 * *x*

5. Repeat steps 1 through 4 for all remaining analog input channels of the device. Replace `ai0` with the appropriate channel name in function call parameters `physicalChannel` and `channel`.

Timebase Frequency Accuracy Verification

This section describes the verification process for the timebase frequency accuracy on all the NI 447*X* devices. All analog inputs use a single circuit on each device. Therefore, you need to verify only the analog input frequency accuracy on a single channel to verify this circuit accuracy.

Complete the following steps to verify the timebase:

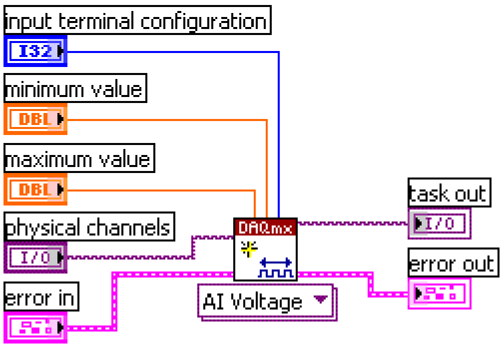
1. Connect the function generator to the analog input channel 0 of the device.
2. Output a sine wave using the function generator. The sine wave should have an amplitude of 9 V_{p-p} with no DC offset and a frequency of 10 kHz.
3. If you use LabVIEW, skip this step. If you use C function calls, create a task using `DAQmxCreateTask`, as shown below. The task is created in step 4 in LabVIEW.

LabVIEW Block Diagram	NI-DAQmx Function Call
LabVIEW does not require this step.	Call `DAQmxCreateTask` with the following parameter: **taskHandle**: `mytaskHandleReturnValue`

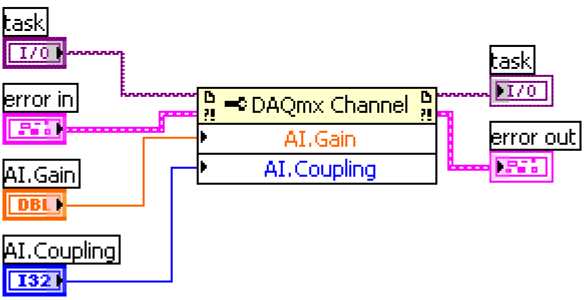
You use the object `myTaskHandle` in all subsequent NI-DAQmx function calls.

4. Create an AI voltage channel through NI-DAQmx using the DAQmx Create Virtual Channel VI, as shown below.

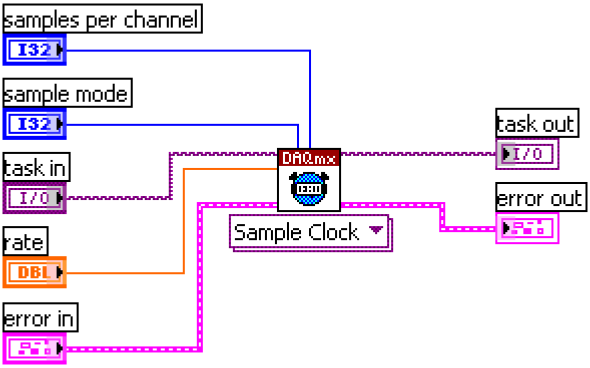
Note Refer to the NI-DAQmx function parameters for the LabVIEW input values.

<table>
<tr><th>LabVIEW Block Diagram</th><th>NI-DAQmx Function Call</th></tr>
<tr><td>
</td><td>Call <code>DAQmxCreateAIVoltageChan</code> with the following parameters:
taskHandle: <code>myTaskHandle</code>
physicalChannel: <code>dev1/ai0</code>
nameToAssignToChannel: <code>""</code>
terminalConfig: <code>DAQmx_Val_PseudoDiff</code>
minVal: <code>-10.0</code>
maxVal: <code>10.0</code>
units: <code>DAQmx_Val_Volts</code>
CustomScaleName: <code>NULL</code></td></tr>
</table>

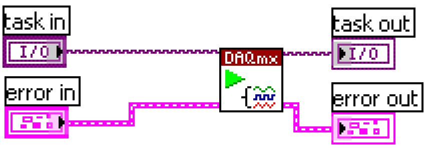
5. Modify the AI voltage channel properties using the following functions.

<table>
<tr><th>LabVIEW Block Diagram</th><th>NI-DAQmx Function Call</th></tr>
<tr><td rowspan="2">
</td><td>Call <code>DAQmxSetAIGain</code> with the following parameters:
taskHandle: <code>myTaskHandle</code>
channel: <code>dev1/ai0</code>
data: <code>0.0</code></td></tr>
<tr><td>Call <code>DAQmxSetAICoupling</code> with the following parameters:
taskHandle: <code>myTaskHandle</code>
channel: <code>dev1/ai0</code>
data: <code>DAQmx_Val_DC</code></td></tr>
</table>

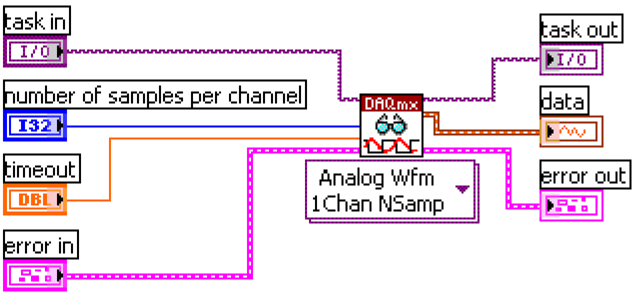
6. Configure the timing properties of the acquisition using the DAQmx Timing VI, as shown below.

LabVIEW Block Diagram	NI-DAQmx Function Call
	Call `DAQmxCfgSampClkTiming` with the following parameters: **taskHandle**: `myTaskHandle` **source**: `NULL` **rate**: `40000.0` **activeEdge**: `DAQmx_Val_Rising` **sampleMode**: `DAQmx_Val_FiniteSamps` **sampsPerChanToAcquire**: `2560000`

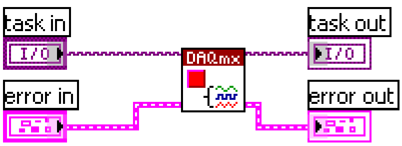
7. Start the task using the DAQmx Start Task VI, as shown below.

LabVIEW Block Diagram	NI-DAQmx Function Call
	Call `DAQmxStartTask` with the following parameters: **taskHandle**: `myTaskHandle`

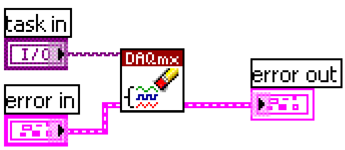
8. If you use LabVIEW, acquire data using the DAQmx Read VI, as shown below. The acquisition takes over 1 minute. If you use C function calls, create an array of double float of size 2560000 that will be filled with data.

LabVIEW Block Diagram	NI-DAQmx Function Call
	Call `DAQmxReadAnalogF64` with the following parameters: **taskHandle**: `myTaskHandle` **numSampsPerChan**: `-1` **timeout**: `100.0` **fillMode**: `DAQmx_Val_GroupByChannel` **arraySizeInSamps**: `2560000` **reserved**: `NULL` **readArray**: `float64ArrayReturnValue` **sampsPerChanRead**: `int32PointerReturnValue`

9. Stop the task using the DAQmx Stop Task VI, as shown below.

LabVIEW Block Diagram	NI-DAQmx Function Call
	Call `DAQmxStopTask` with the following parameters: **taskHandle**: `myTaskHandle`

10. Clear the task using the DAQmx Clear Task VI, as shown below.

LabVIEW Block Diagram	NI-DAQmx Function Call
	Call `DAQmxClearTask` with the following parameters: **taskHandle**: `myTaskHandle`

11. If you use LabVIEW, use the Analog 1D Wfm 1chan Nsamp polymorphic VI to acquire data in step 8. Measure the exact frequency of the peak around 10 kHz using the Extract Single Tone Information VI with the following inputs:
    - **advanced search»approx freq.**: 10000
    - **advanced search»search**: 1
    - **export signals**: 0 (none)

    If you use C function calls, perform an FFT on the array of data.
12. Compare the detected frequency to the limits in Table 5.

Table 5. Timebase Frequency Accuracy Limits

Function Generator Output		Device Input Frequency	
Amplitude (V_{p-p})	**Frequency (Hz)**	**Min (Hz)**	**Max (Hz)**
9.0	10,000.00	9,999.75	10,000.25

Adjustment Procedure

The NI 447*X* adjustment procedure adjusts the analog input calibration constants. At the end of each adjustment procedure, the new constants are stored in the external calibration area of the device EEPROM, which ensures that you do not accidentally access or modify any calibration constants adjusted by the metrology laboratory while performing a self-calibration procedure.

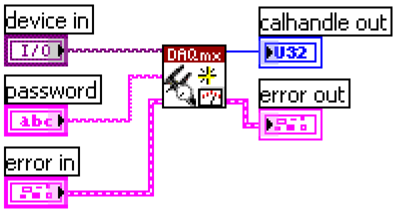
Analog Input Adjustment

Complete the following steps to adjust the analog input:

1. Connect the output of the calibrator to all the analog input channels of the device, using the BNC T-connectors to split the signal in a tree formation to all the inputs.
2. Use the calibrator to output 5.00 VDC.

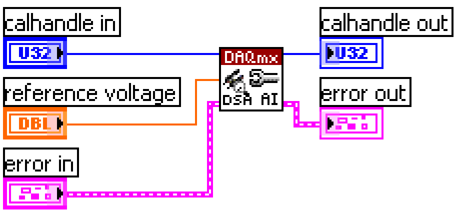
3. Initialize the AI calibration using the DAQmx Initialize External Calibration VI, as shown below.

Note Refer to the NI-DAQmx function parameters for the LabVIEW input values.

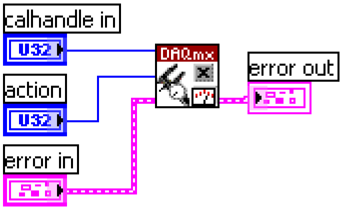
LabVIEW Block Diagram	NI-DAQmx Function Call
	Call `DAQmxInitExtCal` with the following parameters: **deviceName**: `dev1` **password**: `NI` **calHandle**: `myCalHandle`

You use the object `myCalHandle` in all subsequent NI-DAQmx function calls.

4. Perform the AI calibration using the DAQmx Adjust DSA AI Calibration VI, as shown below. The reference voltage parameter is the voltage value output by the calibrator.

LabVIEW Block Diagram	NI-DAQmx Function Call
	Call `DAQmxAdjustDSAAICal` with the following parameters: **calHandle**: `myCalHandle` **referenceVoltage**: `5.0`

5. Finish the AI calibration with the DAQmx Close External Calibration VI, as shown below. Use the action `cancel` if there has been any error during the AI calibration or if you do not want to save the new AI calibration constants in the device EEPROM. Use the action `commit` if you want to save the new AI calibration constants in the device EEPROM.

LabVIEW Block Diagram	NI-DAQmx Function Call
	Call `DAQmxCloseExtCal` with the following parameters: **calHandle**: `myCalHandle` **action**: `DAQmx_Val_Action_Commit` or `DAQmx_Val_Action_Cancel`

The device is now calibrated with respect to your external source. After calibrating the device, verify the analog input operation by repeating the *Verification Procedure* section.

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 385 (0) 9 725 72511, France 33 (0) 1 48 14 24 24,
Germany 49 89 7413130, India 91 80 41190000, Israel 972 3 6393737,
Italy 39 02 413091, Japan 81 3 5472 2970, Korea 82 02 3451 3400,
Lebanon 961 (0) 1 33 28 28, Malaysia 1800 887710,
Mexico 01 800 010 0793, Netherlands 31 (0) 348 433 466,
New Zealand 0800 553 322, Norway 47 (0) 66 90 76 60,
Poland 48 22 3390150, Portugal 351 210 311 210, Russia 7 495 783 6851,
Singapore 1800 226 5886, Slovenia 386 3 425 42 00,
South Africa 27 0 11 805 8197, Spain 34 91 640 0085,
Sweden 46 (0) 8 587 895 00, Switzerland 41 56 2005151,
Taiwan 886 02 2377 2222, Thailand 662 278 6777,
Turkey 90 212 279 3031, United Kingdom 44 (0) 1635 523545

372005A-01 Mar07