ARTISAN®
TECHNOLOGY GROUP

Your **definitive** source for quality pre-owned equipment.

Artisan Technology Group
(217) 352-9330 | sales@artisantg.com | artisantg.com

Full-service, independent repair center
with experienced engineers and technicians on staff.

We buy your excess, underutilized, and idle equipment
along with credit for buybacks and trade-ins.

Custom engineering
so your equipment works exactly as you specify.

- Critical and expedited services
- In stock / Ready-to-ship
- Leasing / Rentals / Demos
- ITAR-certified secure asset solutions

Expert team | Trust guarantee | 100% satisfaction

All trademarks, brand names, and brands appearing herein are the property of their respective owners.

Find the *National Instruments PCI-4451* at our website: *Click HERE*

NI 445*X*/455*X* Calibration Procedure

Contents

Introduction 2
- What Is Calibration? 2
- Why Should You Calibrate? 2
- How Often Should You Calibrate? 2
- Calibration Options: External Calibration Versus Self-Calibration 3
  - Self-Calibration 3
  - External Calibration 3

Equipment and Other Test Requirements 4
- Test Equipment 4
  - Connectors 5
- Test Conditions 5
- Documentation 5
- Software 6
- Writing the Calibration Procedure 6
- Configuring the NI 445*X*/455*X* in NI-DAQ 7

Calibration Process 7
- Verifying the Operation of the NI 445*X*/455*X* 8
  - Verifying Analog Input 8
  - Verifying Analog Output 21
  - Verifying the Counter 32
  - Verifying the NI-TIO Counter (NI 4551 and NI 4552) 33
  - Verifying the DAQ-STC Counter (NI 4451 and NI 4452) 34
- Externally Adjusting the NI 445*X*/455*X* 37

Specifications 38
- NI 4451 38
- NI 4452 39
- NI 4454 40
- NI 4551 40
- NI 4552 42

CVI™, DAQ-STC™, LabVIEW™, National Instruments™, NI™, ni.com™, and NI-DAQ™ are trademarks of National Instruments Corporation. Product and company names mentioned herein are trademarks or trade names of their respective companies. For patents covering National Instruments products, refer to the appropriate location: **Help»Patents** in your software, the `patents.txt` file on your CD, or `ni.com/patents`.

September 2002
370333A-01

© 2002 National Instruments Corp. All rights reserved.

NATIONAL INSTRUMENTS

Introduction

This document contains information and step-by-step instructions for verifying and adjusting National Instruments PCI-445*X*/455*X* (NI 445*X*/455*X*) devices. The NI 445*X*/455*X* families include the following devices:

- NI PCI-4451 (NI 4451)
- NI PCI-4452 (NI 4452)
- NI PCI-4454 (NI 4454)
- NI PCI-4551 (NI 4551)
- NI PCI-4552 (NI 4552)

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. This section defines calibration, describes why it is necessary, and explains how frequently you should calibrate the NI 445*X*/455*X*. It also explains internal and external calibration.

What Is Calibration?

Calibration is the process of verifying the measurement accuracy of a device and adjusting for 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 verification and adjustment.

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 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 NI 445*X*/455*X*. 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 application demands.

Calibration Options: External Calibration Versus Self-Calibration

NI 445*X*/455*X* products have two calibration options: an internal, or self-calibration, and an external calibration. Both calibration modes remove measurement error.

Self-Calibration

Self-calibration is a simple calibration method that does not rely on external standards. In this method, the NI 445*X*/455*X* 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 should be performed after the device has been externally calibrated. Self-calibration compensates for temperature drift.

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 NI 445*X*/455*X*. Adjustments are then made to the device calibration constants to ensure that the reported voltages fall within specifications. The new calibration constants are 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. Once 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 later in this document are for external calibration.

Equipment and Other Test Requirements

This section describes the equipment, test conditions, documentation, and software required for calibrating the NI 445*X*/455*X*.

Test Equipment

Table 1 lists the instruments recommended for calibrating the NI 445*X*/455*X*.

Table 1. NI 445*X*/455*X* Calibration Equipment Specifications

Instrument	Critical Specifications	Suggested Model	Required For
Calibrator	Frequency Range: 20 Hz–95 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	NI 4451, NI 4452, NI 4454, NI 4551, NI 4552
Function Generator	Frequency Range: 20 Hz–95 kHz Frequency Accuracy: ±6 ppm[3] Voltage Range: 3 V_{rms}	Agilent (HP) 3324A	NI 4451, NI 4452, NI 4454, NI 4551, NI 4552
Digital Voltmeter	Voltage Range: 5 mV_{rms}–5 V_{rms} AC Accuracy: ±0.2%[1], 20 Hz–95 kHz DC Accuracy: ±500 µV[2] at 0 V	Agilent (HP) 3458A	NI 4451, NI 4551
Frequency Counter	Frequency Range: 20 Hz–5 MHz Frequency Accuracy: ±6 ppm[3] 20 Hz–20 kHz Frequency Accuracy: ±25 ppm[3] 20 Hz–5 MHz Input Voltage: 3 V_{rms}	Agilent (HP) 53131A	NI 4451, NI 4452, NI 4551, NI 4552

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

[2] DC accuracy is the sum of all errors, including percent of reading error, percent of range error, and temperature error.

[3] 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.

Connectors

The following NI breakout boxes and cables are required for calibration:

- BNC-2140 (NI 4451/4452/4551/4552)
- BNC-2142 (NI 4454)
- SHC 68-C68-A1 cable
- SCB-68, LC-68, or CB-68 (NI 4451/4452/4551/4552)
- SHC 50-68 cable (NI 4451/4452/4551/4552)

Note Use the analog breakout boxes, BNC-2140 or BNC-2142, for analog input and analog output verification and the digital breakout boxes, SCB-68, LC-68, or CB-68, for counter verification.

Test Conditions

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

- Keep connections to the NI 445*X*/455*X* 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 digitizer.
- Keep relative humidity between 10 and 90%, noncondensing.
- Maintain the temperature between 5 and 40 °C.
- Allow a warm-up time of at least 15 minutes to ensure that the measurement circuitry of the device is at a stable operating temperature.

Documentation

This section describes the documentation you need to calibrate the NI 445*X*/455*X*. For detailed information about NI-DAQ, refer to the following documents:

- *NI-DAQ Function Reference Help*, available by selecting **Programs»National Instruments»NI-DAQ»NI-DAQ Help** from the Start menu
- *NI-DAQ User Manual for PC Compatibles*, available at `ni.com/manuals`

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 NI 445*X*/455*X* devices are PC-based measurement devices, you must have the proper device driver installed in the calibration system before calibrating or verifying the device.

Note NI highly recommends installing NI-DAQ before installing the NI 445*X*/455*X*. For the NI 445*X*/455*X* 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/downloads`.

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, you only need to 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 the NI-DAQ driver 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 NI 445X/455X* 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 NI 445*X*/455*X* in NI-DAQ

NI-DAQ automatically detects all NI 445*X*/455*X* devices. However, the devices must be configured in NI-DAQ to enable communication. The following steps briefly explain how to configure the devices in NI-DAQ. Refer to the device user manual for detailed installation instructions. You can install this manual when installing NI-DAQ.

1. Power off the computer that will hold the NI 445*X*/455*X*.
2. Install the device in an available slot.
3. Power on the computer and launch 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 NI 445*X*/455*X* is now configured.

Note After the NI 445*X*/455*X* is configured in MAX, the device is assigned a device number. Write this number down and use it to verify and calibrate the device.

Calibration Process

To calibrate the NI 445*X*/455*X*, complete the following steps:

1. Verify the existing operation of the device. This step, which is covered in the *Verifying the Operation of the NI 445X/455X* section, confirms whether the device was 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 NI 445X/455X* section.
3. Perform another verification to ensure that the device is operating within its specifications after adjustment.

Verifying the Operation of the NI 445*X*/455*X*

Verification determines how well the device is meeting its specifications. By performing this procedure, you can see how the device has operated over time. This information helps determine the appropriate calibration interval for your application.

The verification procedure covers AI, AO, and counter performance. Unless otherwise stated, all function calls are part of NI-DAQ or the `niDsaCal.dll`.

Verifying Analog Input

This procedure verifies the performance of the AI for all NI 445*X*/455*X* devices. Refer to the *Specifications* section for the number of channels, gain settings, and performance specifications for the device.



Note You need the device number for some of the function calls. To find this number, refer to the *Configuring the NI 445X/455X in NI-DAQ* section earlier in this document.

Verifying DC Offset

The following flowchart shows the programming code needed to verify the AI DC offset for the NI 445*X*/455*X*.

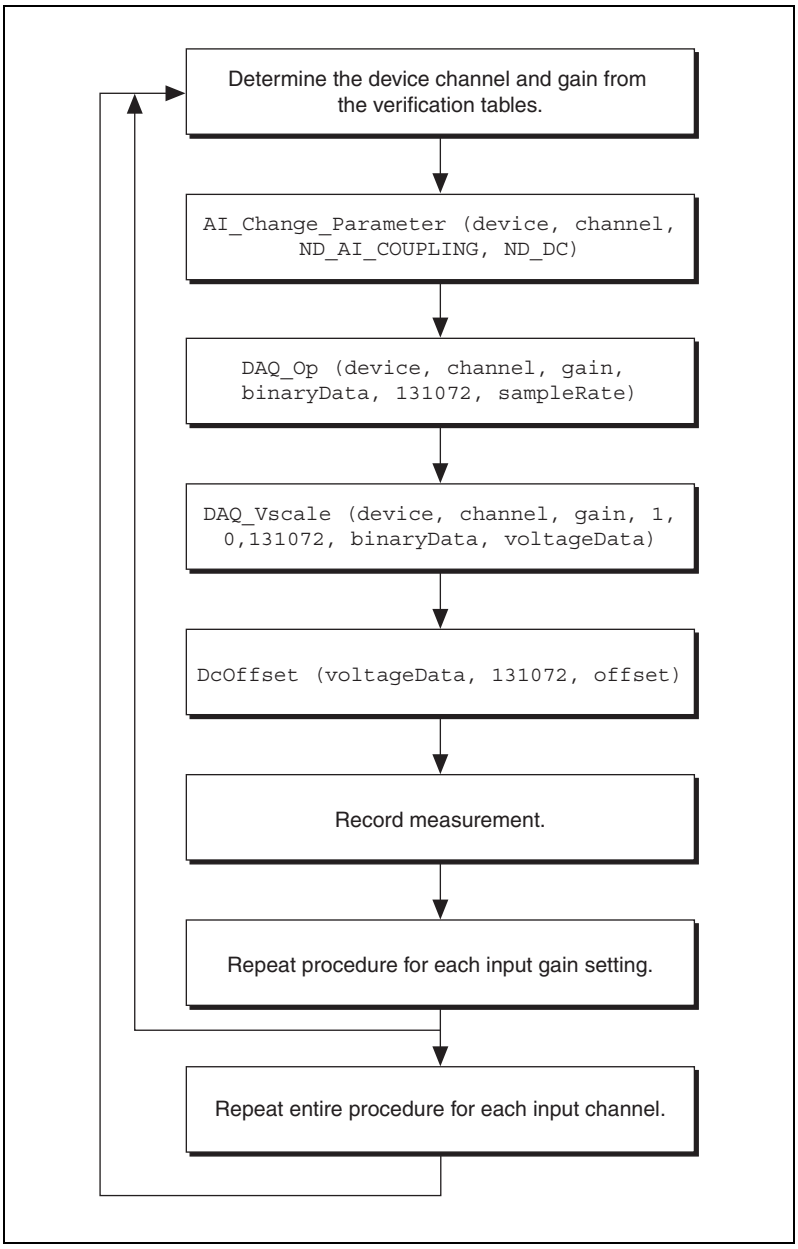


Figure 1. Verifying AI DC Offset

Table 2. AI DC Offset for the NI 4451/4452/4551/4552

Device Gain (dB)	Signal Source	Device Input Amplitude		
		Min (mV)	Actual (mV)	Max (mV)
–20	Terminated	–30		30
–10	Terminated	–10		10
0	Terminated	–3		3
10	Terminated	–1		1
20	Terminated	–0.3		0.3
30	Terminated	–0.1		0.1
40	Terminated	–0.1		0.1
50	Terminated	–0.1		0.1
60	Terminated	–0.1		0.1

Note Use only the Device Gain (dB) 0 column in Table 2 for verifying DC offset on the NI 4454.

To verify DC offset, refer to Table 2 as you complete the following steps:

1. Ground input channel 0 on the BNC-2140 with a 50 Ω BNC terminator.
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. Create an array called **binaryData** containing `131072` elements. Call `DAQ_Op` with the following parameters:
   - **chan**—`0`
   - **gain**—Device gain from Table 2 (Refer to the note following Table 2.)
   - **count**—`131072`
   - **sampleRate**—`204800` (NI 4451/4452/4551/4552) or `51200` (NI 4454)

   This function acquires data from the NI 445*X*/455*X*.

Note Use the first row in Table 2 initially. When you repeat this procedure, use the next row (2, 3, 4, and so on). Use only the Device Gain (dB) 0 row in Table 2 for verifying DC offset on the NI 4454.

4. Call `DAQ_VScale` with the following parameters:
   - **chan**—`0`
   - **gain**—Same value from step 3
   - **gainAdjust**—`1`

   Create an array called **voltageData** containing `131072` elements with **binaryZeroOffset** set to `0`.

   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 row in Table 2 (2, 3, 4, and so on).
8. Repeat steps 1 through 7 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 NI 445*X*/455*X*.

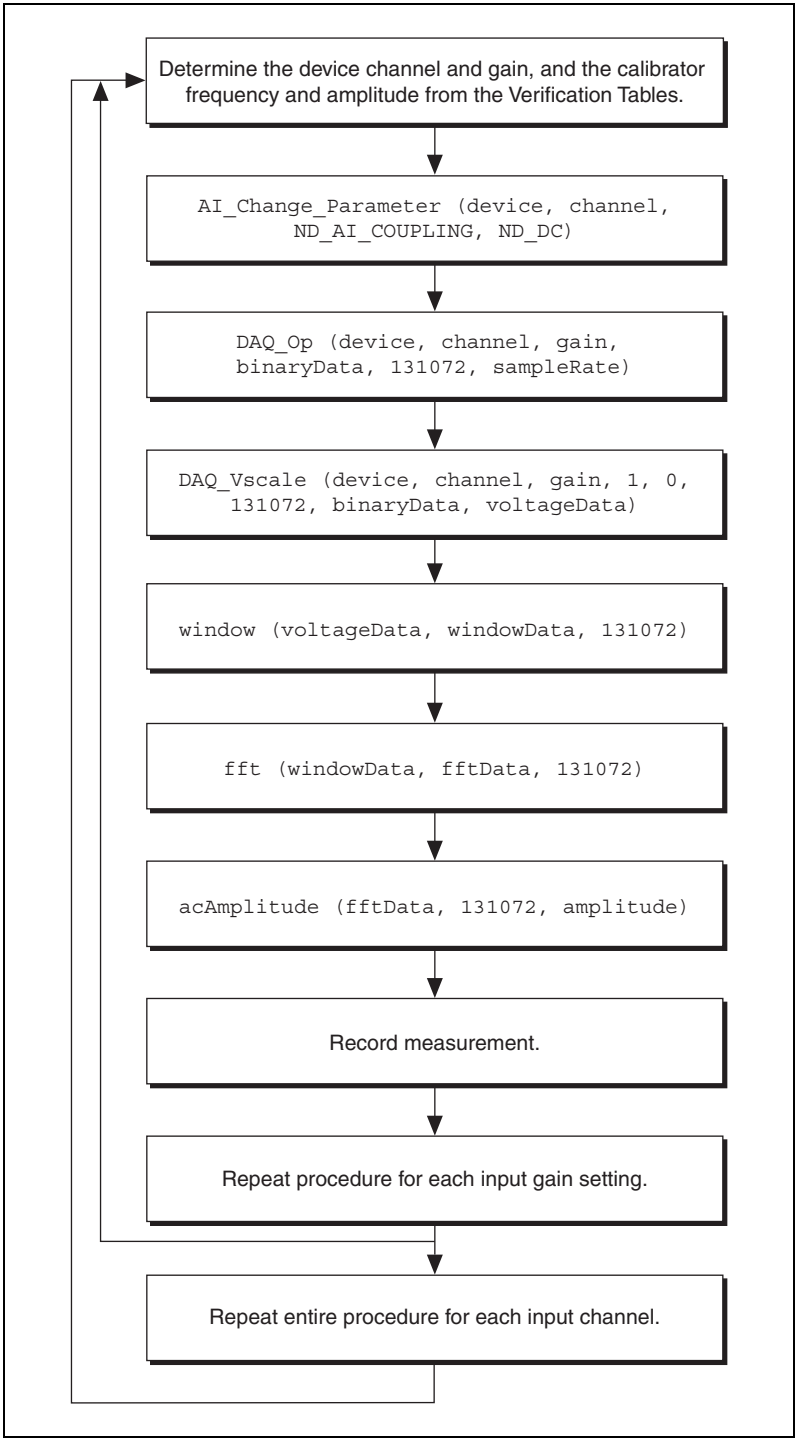


Figure 2. Verifying AI Gain Accuracy and Flatness

Table 3. AI Gain Accuracy for the NI 4451/4452/4454/4551/4552

Device Gain (dB)	Calibrator Output		Device Input Amplitude		
	Amplitude (V_{rms})	Frequency (Hz)	Min (V_{rms})	Actual (V_{rms})	Max (V_{rms})
–20	15	1,000	14.828	A1:	15.174
–10	15	1,000	14.828	A2:	15.174
0	5	1,000	4.943	A3:	5.058
10	1.5	1,000	1.4828	A4:	1.5174
20	0.5	1,000	0.4943	A5:	0.5058
30	0.15	1,000	0.14828	A6:	0.15174
40	0.05	1,000	0.04943	A7:	0.05058
50	0.015	1,000	0.014828	A8:	0.015174
60	0.005	1,000	0.004943	A9:	0.005058

To verify AI gain accuracy, refer to Table 3 as you complete the following steps:

1. Connect the calibrator to input channel 0 on the BNC-2140. Program the calibrator amplitude and frequency listed in Table 3.

Note Use the first row in Table 3 initially. When you repeat this procedure, use the next row (2, 3, 4, and so on). Use only the Device Gain (dB) 0 row in Table 3 to verify the AI gain accuracy on the NI 4454.

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. Create an array called **binaryData** containing `131072` elements. Call the `DAQ_Op` with the following parameters:
   - **chan**—`0`
   - **gain**—Device gain from Table 2 (Refer to the note.)
   - **count**—`131072`
   - **sampleRate**—`204800` (NI 4451/4452/4551/4552) or `51200` (NI 4454)

   This function acquires data from the NI 445*X*/455*X*.

4. Call `DAQ_VScale` with the following parameters:
   - **chan**—`0`
   - **gain**—Same value as in step 3
   - **gainAdjust**—`1`

   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`. 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 to verify each row in Table 3 (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 gain accuracy for the device.

Verifying Flatness

The flowchart in Figure 2 shows the programming code needed to verify the AI flatness for the NI 445*X*/455*X*.

Table 4. AI Flatness for the NI 4451/4452/4551/4552

Device Gain (dB)	Calibrator Output		Device Input Amplitude		
	Amplitude (V_{rms})	Frequency (Hz)	Min (V_{rms})	Actual (V_{rms})	Max (V_{rms})
–20	15	20	98.9% A1		101.2% A1
		2,000	98.9% A1		101.2% A1
		20,000	98.9% A1		101.2% A1
		50,000	89.1% A1		112.2% A1
		95,000	89.1% A1		112.2% A1
–10	15	95,000	89.1% A2		112.2% A2
		50,000	89.1% A2		112.2% A2
		20,000	98.9% A2		101.2% A2
		2,000	98.9% A2		101.2% A2
		20	98.9% A2		101.2% A2
0	5	20	98.9% A3		101.2% A3
		2,000	98.9% A3		101.2% A3
		20,000	98.9% A3		101.2% A3
		50,000	98.9% A3		101.2% A3
		95,000	98.9% A3		101.2% A3
10	1.5	95,000	98.9% A4		101.2% A4
		50,000	98.9% A4		101.2% A4
		20,000	98.9% A4		101.2% A4
		2,000	98.9% A4		101.2% A4
		20	98.9% A4		101.2% A4
20	0.5	20	98.9% A5		101.2% A5
		2,000	98.9% A5		101.2% A5
		20,000	98.9% A5		101.2% A5
		50,000	98.9% A5		101.2% A5
		95,000	98.9% A5		101.2% A5

Table 4. AI Flatness for the NI 4451/4452/4551/4552 (Continued)

Device Gain (dB)	Calibrator Output		Device Input Amplitude		
	Amplitude (V_{rms})	Frequency (Hz)	Min (V_{rms})	Actual (V_{rms})	Max (V_{rms})
30	0.15	95,000	98.9% A6		101.2% A6
		50,000	98.9% A6		101.2% A6
		20,000	98.9% A6		101.2% A6
		2,000	98.9% A6		101.2% A6
		20	98.9% A6		101.2% A6
40	0.05	20	98.9% A7		101.2% A7
		2,000	98.9% A7		101.2% A7
		20,000	98.9% A7		101.2% A7
		50,000	98.9% A7		101.2% A7
		95,000	98.9% A7		101.2% A7
50	0.015	95,000	89.1% A8		112.2% A8
		50,000	89.1% A8		112.2% A8
		20,000	98.9% A8		101.2% A8
		2,000	98.9% A8		101.2% A8
		20	98.9% A8		101.2% A8
60	0.005	20	98.9% A9		101.2% A9
		2,000	98.9% A9		101.2% A9
		20,000	98.9% A9		101.2% A9
		50,000	89.1% A9		112.2% A9
		95,000	89.1% A9		112.2% A9

Table 5. AI Flatness for the NI 4454

Device Gain (dB)	Calibrator Output		Device Input Amplitude		
	Amplitude (V_{rms})	Frequency (Hz)	Min (V_{rms})	Actual (V_{rms})	Max (V_{rms})
0	15	20	98.9% A3		101.2% A3
		2,000	98.9% A3		101.2% A3
		20,000	98.9% A3		101.2% A3

To verify AI flatness, refer to Table 4 or Table 5 as you complete the following steps:

1. Connect the calibrator to input channel 0 on the BNC-2140 or BNC-2142. Program the calibrator amplitude and frequency listed in Table 4 or Table 5.

Note Use the first row in Table 4 or 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. Create an array called **binaryData** containing `131072` elements. Call the `DAQ_Op` with the following parameters:
   - **chan**—`0`
   - **gain**—Device gain from Table 4 or 5 (Refer to the note.)
   - **count**—`131072`
   - **sampleRate**—`204800` (NI 4451/4452/4551/4552) or `51200` (NI 4454)

   This function acquires data from the NI 445*X*/455*X*.
4. Call `DAQ_VScale` with the following parameters:
   - **chan**—`0`
   - **gain**—Same value used in step 3
   - **gainAdjust**—`1`

   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 or Table 5. 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 or Table 5 for the specified gain.
8. Repeat steps 1 through 7 to verify each row in Table 4 or Table 5 (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 NI 445*X*/455*X*.

Verifying Frequency Accuracy

The flowchart in Figure 3 shows the programming code needed to verify the AI frequency accuracy for the NI 445*X*/455*X*.

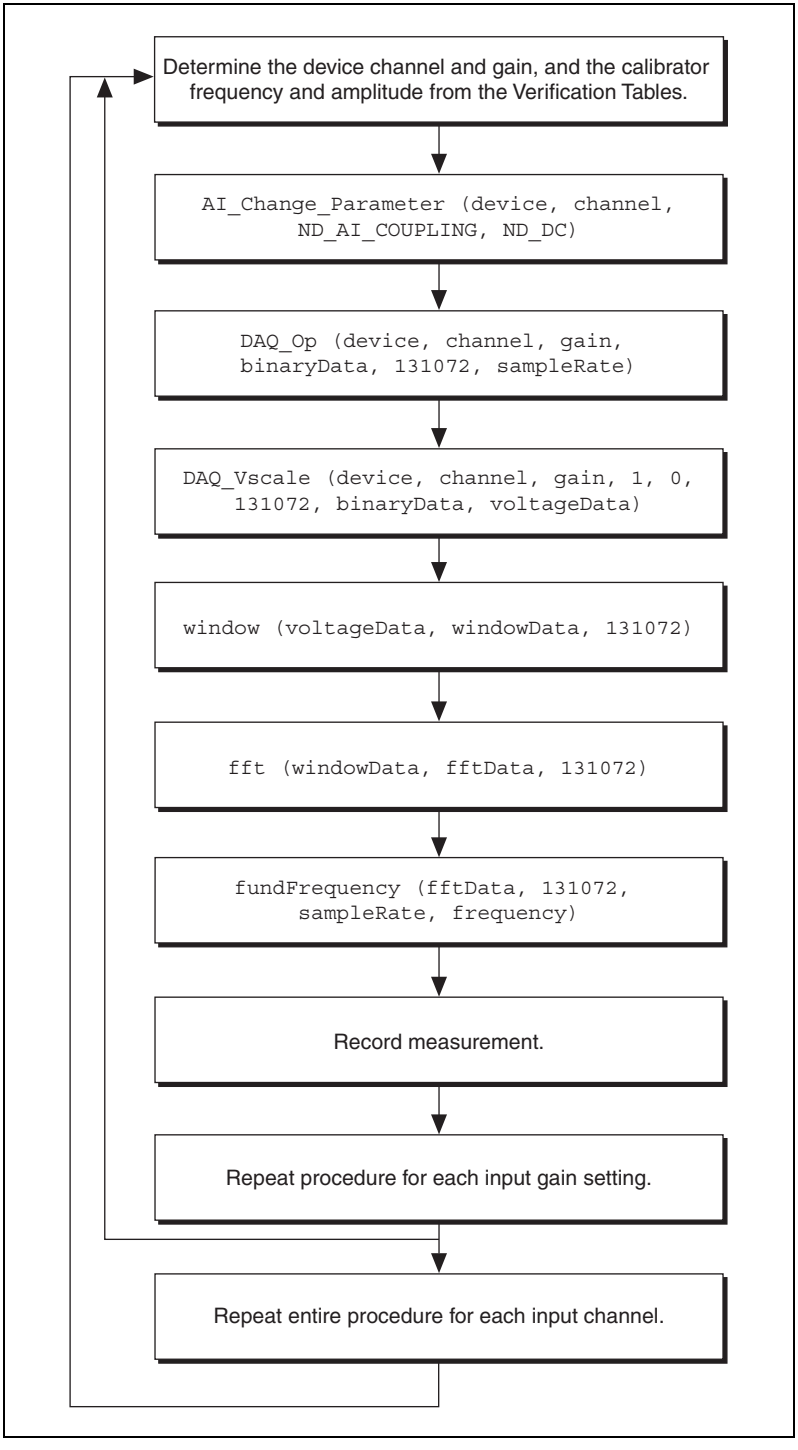


Figure 3. Verifying AI Frequency Accuracy

Table 6. 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
		2,000	1,999.95		2,000.05
		20,000	1,999.95		20,000.5
		50,000	49,999		50,001
		95,000	94,998		95,002

Note Use only the first 3 rows of Table 6 to verify the AI frequency for the NI 4454.

To verify AI frequency accuracy, refer to Table 6 as you complete the following steps:

1. Connect the function generator to input channel 0 on the BNC-2140. Program the function generator amplitude and frequency listed in Table 6.

Note Use the first row in Table 6 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. Create an array called **binaryData** containing `131072` elements. Call the `DAQ_Op` with the following parameters:
   - **channel**—`0`
   - **gain**—Device gain from Table 6
   - **count**—`131072`
   - **sampleRate**—`204800` (NI 4451/4452/4551/4552) or `51200` (NI 4454)

   This function acquires data from the NI 445*X*/455*X*.

4. Call `DAQ_VScale` with the following parameters:
   - **channel**—`0`
   - **gain**—Same value used in step 3
   - **gainAdjust**—`1`
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** will contain 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 `niDsaCal.dll`. Pass in the **fftData** obtained from the `fft` function. Set the following parameters:
   - **scans**—`131072`
   - **sampleRate**—`204800` (NI 4451/4452/4551/4552) or `51200` (NI 4454)

   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 7 to verify each row in Table 6 (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.

Verifying Analog Output

Note This step applies only to the NI 4451 and NI 4551. The NI 4452/4552/4454 do not support AO.

This procedure verifies the performance of the AO for NI 445*X*/455*X* products that support AO. Refer to the *Specifications* section for the number of channels, gain settings, and performance specifications for the device.

Verifying DC Offset

The flowchart in Figure 4 shows the appropriate NI-DAQ function calls for verifying AO DC offset.

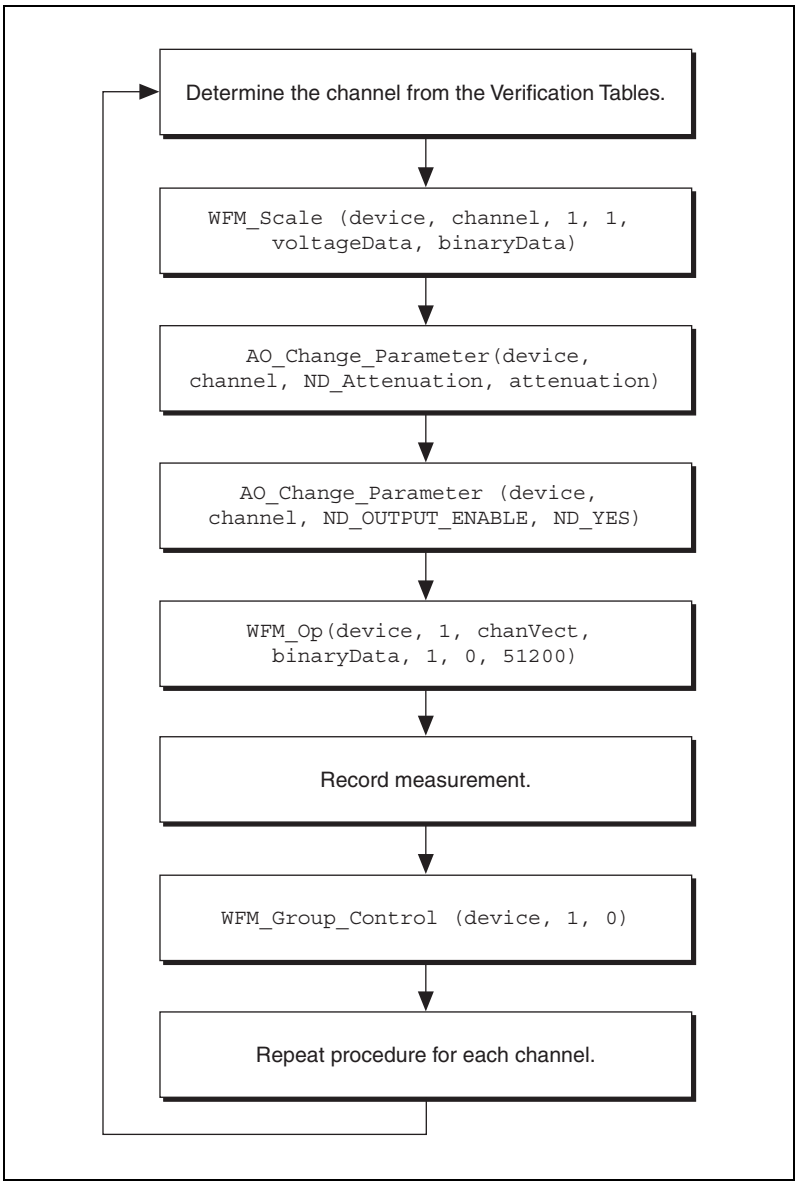


Figure 4. Verifying AO DC Offset

Table 7. AO DC Offset

Device Attenuation (dB)	Device Output Amplitude (V)	Digital Voltmeter Amplitude		
		Min (mV)	Actual (mV)	Max (mV)
0	0	–5		5
20	0	–5		5
40	0	–5		5

To verify AO DC Offset, refer to Table 7 as you complete the following steps:

1. Connect the digital voltmeter to output channel 0 of the BNC-2140.
2. Refer to Table 7 for the amplitude to be generated by the NI 445*X*/455*X*.
3. Call `WFM_Scale` with the following parameters:
   - **chan**—`0`
   - **bufferSize**—`1`
   - **gain**—`1`

   Pass in an array of **voltageData** with two elements of value equal to `0`. Calling this function converts **voltageData** to an array called **binaryData** containing two elements.
4. Call `AO_Change_Parameter` with the following parameters:
   - **channel**—0
   - **paramID**—`ND_Attenuation`
   - **paramValue**—Attenuation in Table 7

Note Use the first row in Table 7 initially. When you repeat this procedure, use the next row (2, 3, 4, and so on). Convert decibels (dB) in the table to millidecibels (mdB) for this function.

5. Call `AO_Change_Parameter` with the following parameters:
   - **channel**—`0`
   - **paramID**—`ND_OUTPUT_ENABLE`
   - **paramValue**— `ND_YES`

   This function enables output channel 0.

6. Call `WFM_Op` with the following parameters:
   - **numChans**—`1`
   - **chanVect**—`0`

   Pass in the array of **binaryData** generated by `WFM_Scale` with the following parameters:
   - **bufferSize**—`1`
   - **iterations**—`0`
   - **updateRate**—`51200`

   This function transfers the output voltage data to the device and begins output generation.
7. Record the measurement made by the digital voltmeter in Table 7. If the measurement falls within the minimum and maximum values, the device has passed this portion of the DC offset test.
8. Call `WFM_Group_Control` with the following parameters:
   - **group**—`1`
   - **operation**—`0`

   This step discontinues output generation.
9. Repeat steps 1 through 8 to verifies each row in Table 7 (2, 3).
10. Repeat steps 1 through 9 for each output channel. Verify the output channels sequentially (0 then 1).

You have now verified the AO DC offset for the device.

Verifying Gain Accuracy

The flowchart in Figure 5 shows the appropriate NI-DAQ function calls for verifying AO frequency accuracy.

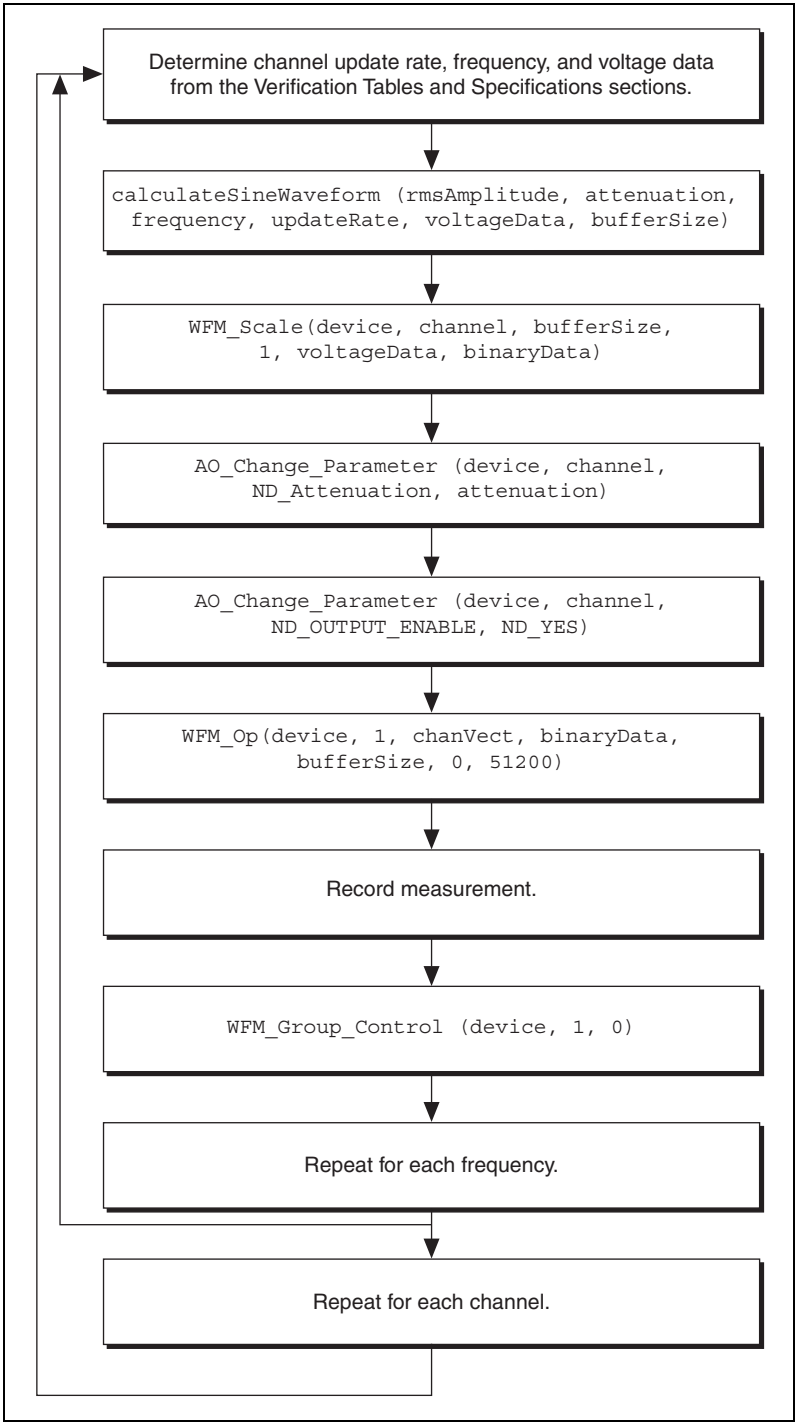


Figure 5. Verifying AO Gain Accuracy, Flatness, and Frequency Accuracy

Table 8. AO Gain Accuracy

Device Attenuation (dB)	Device Output		Digital Voltmeter Amplitude		
	Amplitude (V_{rms})	Frequency (Hz)	Min (V_{rms})	Actual (V_{rms})	Max (V_{rms})
0	5	1,000	4.943	B1:	5.058
20	0.5	1,000	0.4943	B2:	0.5058
40	0.05	1,000	0.04943	B3:	0.05058

This procedure verifies the AO gain amplitude accuracy. To complete each verification, refer to Table 8.

1. Connect the digital voltmeter to output channel 0 of the BNC-2140.
2. Refer to Table 8, and note the entries under *Amplitude* and *Frequency.*
3. Call `calculateSineWaveform` from `niDsaCal.dll`. Pass in the device output amplitude and frequency listed in Table 8. Set the following parameters:
   - **updateRate**—`51200`
   - **attenuation**—Value in Table 8 (Refer to the note.)

   Create an array called **voltageData** containing `65536` elements. This step calculates the sine wave.

Note Use the first row in Table 8 initially. When you repeat this procedure, use the next row (2, 3). Convert decibels (dB) to millidecibels (mdB) for this function.

4. Call `WFM_Scale` with the following parameters:
   - **chan**—`0`
   - **bufferSize**—Buffer size returned by the `calculateSineWaveform` function
   - **gain**—`1`

   Pass in the array of **voltageData**. This function converts the **voltageData** obtained from `calculateSineWaveform` to **binaryData** containing `131072` elements.
5. Call `AO_Change_Parameter` with the following parameters:
   - **channel**—0
   - **paramID**—`ND_Attenuation`
   - **paramValue**—Attenuation (in mdB) used in step 3

6. Call `AO_Change_Parameter` with the following parameters:
   - **channel**—`0`
   - **paramID**—`ND_OUTPUT_ENABLE`
   - **paramValue**—`ND_YES`

   This function enables output channel 0.
7. Call `WFM_Op` with the following parameters:
   - **numChans**—`1`
   - **chanVect**—`0`
8. Pass in **binaryData** obtained from `WFM_Scale` with the following parameters:
   - **bufferSize**—Buffer size returned by `calculateSineWaveform`
   - **iterations**—`0`
   - **updateRate**—`51200`

   This function transfers the output voltage data to the device and begins output generation.
9. Record the measurement made by the digital voltmeter in the *Actual* column in Table 8.
10. Call `WFM_Group_Control` with the following parameters:
    - **group**—`1`
    - **operation**—`0`

    This function discontinues output generation.
11. Repeat steps 1 through 10 for each row in Table 8 (2, 3).
12. Repeat steps 1 through 11 for each output channel. Verify the output channels sequentially (0 then 1).

You have now verified the AO amplitude accuracy of the device.

Verifying Flatness

The flowchart in Figure 5 shows the appropriate NI-DAQ function calls for verifying AO frequency accuracy.

The following procedure verifies the AO flatness. To complete each verification, refer to Table 9.

Table 9. AO Flatness

Device Attenuation (dB)	Device Output		Digital Voltmeter Amplitude		
	Amplitude (V_{rms})	Frequency (Hz)	Min (V_{rms})	Actual (V_{rms})	Max (V_{rms})
0	5	20	97.7% B1		102.3% B1
		200	97.7% B1		102.3% B1
		2,000	97.7% B1		102.3% B1
		20,000	97.7% B1		102.3% B1
		23,000	97.7% B1		102.3% B1
20	0.5	23,000	97.7% B1		102.3% B1
		20,000	97.7% B2		102.3% B2
		2,000	97.7% B2		102.3% B2
		200	97.7% B2		102.3% B2
		20	97.7% B2		102.3% B2
40	0.05	20	97.7% B3		102.3% B3
		200	97.7% B3		102.3% B3
		2,000	97.7% B3		102.3% B3
		20,000	97.7% B3		102.3% B3
		23,000	97.7% B3		102.3% B3

1. Connect the digital voltmeter to output channel 0 of the BNC-2140.
2. Refer to Table 9, and note the first entry under the *Amplitude* and *Frequency* columns.
3. Call `calculateSineWaveform` from `niDsaCal.dll`. Pass in the device output amplitude and frequency listed in Table 9. Set the following parameters:
   - **updateRate**—`51200`
   - **attenuation** (in mdB)—Value in Table 9 (Refer to the note.)

   Create an array called **voltageData** containing `65536` elements.

   This step calculates the sine wave.

Note Use the first row in Table 9 initially. When you repeat this procedure, use the next row (2, 3, 4, and so on).

4. Call `WFM_Scale` with the following parameters:
   - **chan**—`0`
   - **bufferSize**—Buffer size returned by the `calculateSineWaveform` function
   - **gain**—`1`

   Pass in the array of **voltageData**. This function converts the voltage data obtained from the `calculateSineWaveform` to binary data with `131072` elements.
5. Call `AO_Change_Parameter` with the following parameters:
   - **channel**—0
   - **paramID**—`ND_Attenuation`
   - **paramValue**—Attenuation (in mdB) used in step 3
6. Call `AO_Change_Parameter` with the following parameters:
   - **channel**—`0`
   - **paramID**—`ND_OUTPUT_ENABLE`
   - **paramValue**—`ND_YES`

   This function enables output channel 0.
7. Call `WFM_Op` with the following parameters:
   - **numChans**—`1`
   - **chanVect**—`0`
8. Pass in **binaryData** obtained from `WFM_Scale`. Set the following parameters:
   - **bufferSize**—Buffer size returned by the `calculateSineWaveform` function
   - **iterations**—`0`
   - **rate**—`51200`

   This function transfers the output voltage data to the device and begins output generation.
9. Record the measurement made by the digital voltmeter in the *Actual* column in Table 9. If the measurement falls within 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 9 for the specified gain.
10. Call `WFM_Group_Control` with the following parameters:
    - **group**—`1`
    - **operation**—`0`

    This function discontinues output generation.

11. Repeat steps 1 through 10 for each row in Table 9 (2, 3, 4, and so on).
12. Repeat steps 1 through 11 for each output channel. Verify the output channels sequentially (0, then 1).

You have now verified the AO flatness of the device.

Verifying Frequency Accuracy

The flowchart in Figure 5 shows the appropriate NI-DAQ function calls for verifying AO frequency accuracy.

Table 10. AO Frequency Accuracy

Device Attenuation (dB)	Device Output		Frequency Counter Reading		
	Amplitude (V_{rms})	Frequency (Hz)	Min (Hz)	Actual (Hz)	Max (Hz)
0	2.5	20	19.9995		20.0005
		200	199.995		200.005
		2,000	1,999.95		2,000.05
		20,000	19,999.5		20,000.5

This procedure verifies the AO frequency accuracy. To complete this procedure, refer to Table 10.

1. Connect the frequency counter to output channel 0 of the BNC-2140.
2. Refer to Table 10, and note the first entry under the *Amplitude* and *Frequency* columns.
3. Call `calculateSineWaveform` from `niDsaCal.dll`. Pass in the device output amplitude and frequency listed in Table 10. Set the following parameters:
   - **updateRate**—`51200`
   - **attenuation** (in mdB)—Value in Table 10 (Refer to the note.)

   Create an array called **voltageData** containing `65536` elements. This step calculates the sine wave.

Note Use the first row in Table 10 initially. When you repeat this procedure, use the next row (2, 3, 4).

4. Call `WFM_Scale` with the following parameters:
   - **chan**—`0`
   - **bufferSize**—Value returned by the `calculateSineWaveform` function
   - **gain**—`1`

   Pass in an array of **voltageData** with two elements of value equal to `0`. This function converts the voltage data obtained from the `calculateSineWaveform` to binary data.
5. Call `AO_Change_Parameter` with the following parameters:
   - **channel**—0
   - **paramID**—`ND_Attenuation`
   - **paramValue**—Attenuation (in mdB) used in step 3
6. Call `AO_Change_Parameter` with the following parameters:
   - **channel**—`0`
   - **paramID**—`ND_OUTPUT_ENABLE`
   - **paramValue**—`ND_YES`

   This function enables output channel 0.
7. Call `WFM_Op` with the following parameters:
   - **numChans**—`1`
   - **chanVect**—`0`
8. Pass in **binaryData** obtained from `WFM_Scale`. Set the following parameters:
   - **bufferSize**—Buffer size returned by the `calculateSineWaveform` function
   - **iterations**—`0`
   - **updateRate**—`51200`

   This function transfers the output voltage data to the device and begins output generation.
9. Record the measurement made by the frequency counter in the *Actual* column in Table 10. If the measurement falls within the minimum and maximum values, the device has passed this portion of the flatness test.
10. Call `WFM_Group_Control` with the following parameters:
    - **group**—`1`
    - **operation**—`0`

    This function discontinues output generation.

11. Repeat steps 1 through 10 for each row in Table 10 (2, 3, 4).
12. Repeat steps 1 through 11 for each output channel. Verify output channels sequentially (0 then 1).

You have now verified the AO frequency accuracy for the device.

Verifying the Counter

Note The following section applies only to the NI 4451/4452/4551/4552 devices. The NI 4454 does not support counter operations. Use the information for the NI-TIO for the NI 4551 and NI 4552. Use the DAQ-STC information for the NI 4451 or NI 4452.

Refer to Figure 6 and Figure 7, which show the appropriate NI-DAQ function calls for verifying the TIO and DAQ-STC counters.

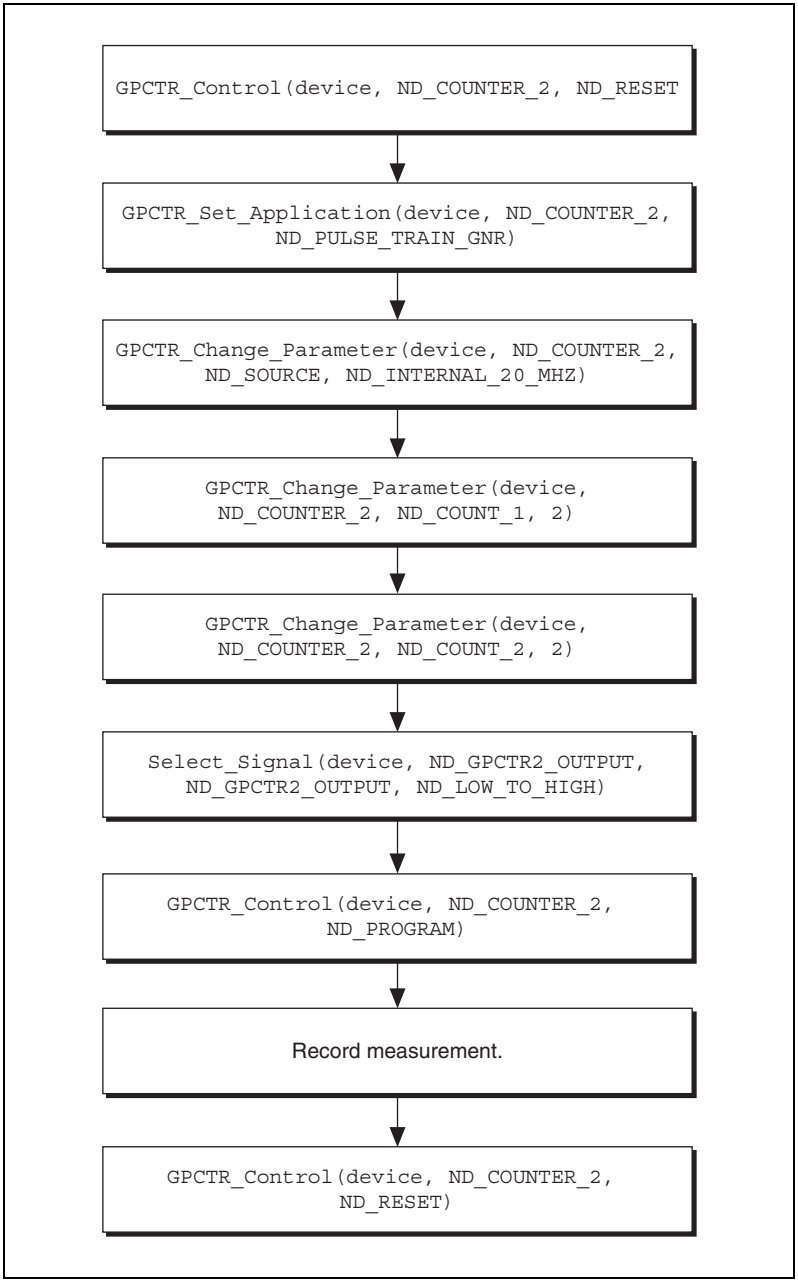


Figure 6. Verifying the NI-TIO Counter (NI 4551 and NI 4552)

Verifying the DAQ-STC Counter (NI 4451 and NI 4452)

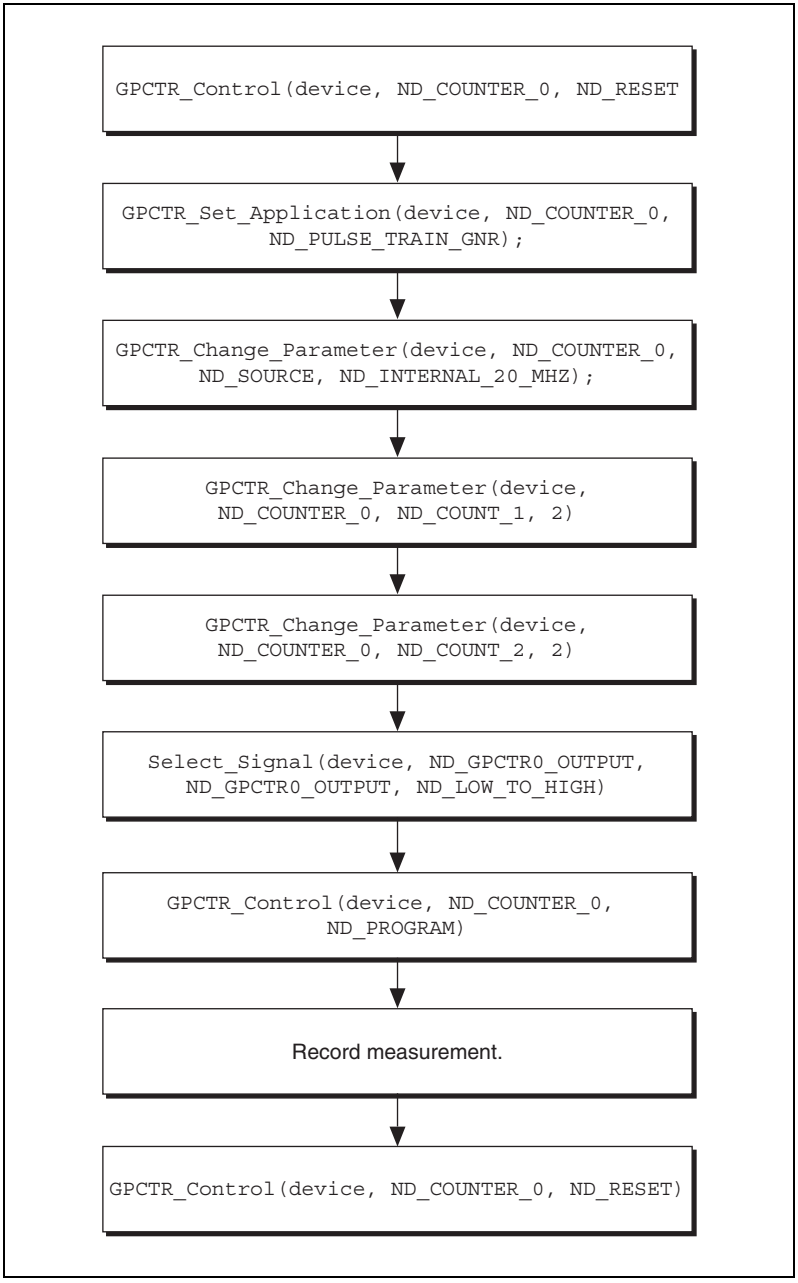


Figure 7. Verifying the DAC-STC Counter (NI 4451 and NI 4452)

Table 11. NI-TIO/DAQ-STC Frequency Accuracy

Device Output	Counter Frequency		
Frequency (MHz)	**Min (MHz)**	**Actual (MHz)**	**Max (MHz)**
5	4.9995		5.0005

To verify the performance of the NI-TIO or DAQ-STC counter, complete the following steps:

1. Connect the frequency counter to positive input.
   - For the NI-TIO counter, connect the positive input to GPCTR2_OUT (SCB-68 pin 30) and the negative input to DGND (SCB-68 pin 63).
   - For the DAQ-STC counter, connect the positive input to GPCTR0_OUT (pin 3 on the SCB-68, LC-68, or CB-68) and the negative input to DGND (pin 35 on the SCB-68, LC-68, or CB-68).
2. To reset the counter, call `GPCTR_Control` with the following parameters:
   - **gpctrNum**—`ND_COUNTER_2` for NI-TIO (`ND_COUNTER_0` for STC)
   - **action**—`ND_RESET`
3. To configure the counter for pulse train generation, call `GPCTR_Set_Application` with the following parameters:
   - **gpctrNum**—`ND_COUNTER_2` for the TIO counter (`ND_COUNTER_0` for the DAQ-STC counter)
   - **application**—`ND_PULSE_TRAIN_GNR`
4. Call `GPCTR_Change_Parameter` with the following parameters:
   - **gpctrNum**—`ND_COUNTER_2` for the NI-TIO counter (`ND_COUNTER_0` for the STC counter)
   - **paramID**—`ND_SOURCE`
   - **paramValue**—`ND_INTERNAL_20_MHZ`

   This step configures the counter to use the internal 20 MHz timebase.
5. Call `GPCTR_Change_Parameter` with the following parameters:
   - **gpctrNum**—`ND_COUNTER_2` for the NI-TIO counter (`ND_COUNTER_0` for the DAQ-STC counter)
   - **paramID**—`ND_COUNT1`
   - **paramValue**—`2`

   This step configures the counter to output a pulse with a low time of 100 ns.

6. Call `GPCTR_Change_Parameter` with the following parameters:
   - **gpctrNum**—`ND_COUNTER_2` for the NI-TIO counter (`ND_COUNTER_0` for the DAQ-STC counter)
   - **paramID**—`ND_COUNT2`
   - **paramValue**—`2`

   This step configures the counter to output a pulse with a high time of 100 ns.
7. Call `Select_Signal` with the following parameters:
   - **signal**—`ND_GPCTR2_OUTPUT` for the NI-TIO counter (`ND_COUNTER_0_OUTPUT` for the DAQ-STC counter)
   - **source**—`ND_GPCTR2_OUTPUT` for the NI-TIO counter (`ND_COUNTER_0_OUTPUT` for the DAQ-STC counter)
   - **sourceSpec**—`ND_LOW_TO_HIGH`

   This step routes the counter signal to the GPCTR2_OUT pin.
8. Call `GPCTR_Control` with the following parameters:
   - **gpctrNum**—`ND_COUNTER_2` for the NI-TIO counter (`ND_COUNTER_0` for the DAQ-STC counter)
   - **action**—`ND_PROGRAM`
9. Measure the frequency of the pulse train using the frequency counter. Record the measurement in the *Actual* column in Table 10. If the value falls between the minimum and maximum values, the device has passed this portion of the test.
10. Call `GPCTR_Control` to reset the counter with the following parameters:
    - **gpctrNum**—`ND_COUNTER_2` for the NI-TIO counter (`ND_COUNTER_0` for the DAQ-STC counter)
    - **action**—`ND_RESET`

The counter is now verified.

Externally Adjusting the NI 445*X*/455*X*

The NI 445*X*/455*X* adjustment procedure adjusts the AI and AO calibration constants. At the end of each calibration procedure, these new constants are stored in the factory area of the device EEPROM, which provides a level of security that ensures you do not accidentally access or modify any calibration constants adjusted by the metrology laboratory.

To adjust the device, follow these steps:

1. Connect the output of the calibrator to all AI channels of the BNC-2140 or BNC-2142.
2. Set the calibrator to output a voltage of 5.000 V.
3. To determine the date of the last calibration, run `getCalDate` 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 NI 445X/455X* section.

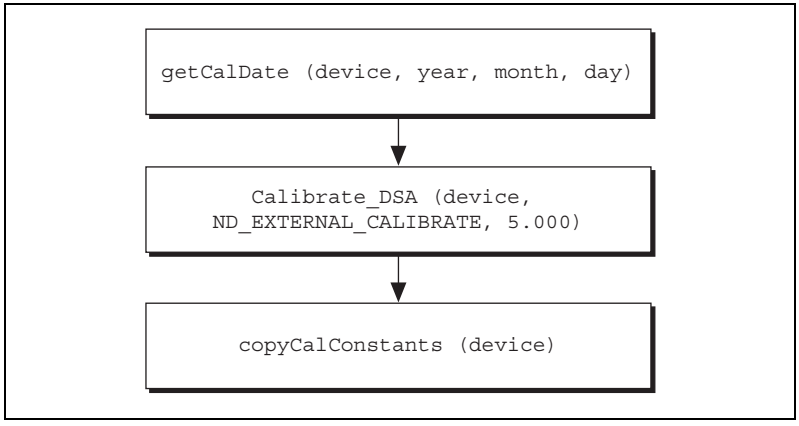


Figure 8. Externally Calibrating the Device

Specifications

This section lists the specifications for the NI 445*X*/455*X* devices.

NI 4451

Input

Channels 0, 1

Gain ranges .....–20, –10, 0, +10, +20, +30, +40, +50, +60 dB

Maximum sample rate .....204.8 kS/s

Offset (residual DC)

Offset (Residual DC)	Voltage
–20 dB	±30 mV
–10 dB	±10 mV
0 dB	±3 mV
+10 dB	±1 mV
+20 dB	±300 µV
+30, +40, +50, +60 dB	±100 µV

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

Flatness (referenced to 1 kHz).....204.8 kS/s, DC coupling

0, +10, +20, +30, +40 dB .....±0.1 dB, 0 to 95 kHz

–20, –10, +50, +60 dB .....±1 dB, 0 to 95 kHz
±0.1 dB, 0 to 20 kHz

Frequency accuracy .....±25 ppm

Output

Channels 0, 1

Gain ranges .....–40, –20, 0 dB

Maximum update rate.....51.2 kS/s

Offset (residual DC) .....±5 mV, any gain

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

Flatness..... 51.2 kS/s, any gain,
±0.2 dB, 0 to 23 kHz

Frequency accuracy..... ±25 ppm

Counter

Type ..... DAQ-STC

Frequency accuracy..... ±100 ppm

NI 4452

Input

Channels..... 0, 1, 2, 3

Gain ranges ..... –20, –10, 0, +10, +20,
+30, +40, +50, +60 dB

Maximum sample rate..... 204.8 kS/s

Offset (residual DC)

Offset (Residual DC)	Voltage
–20 dB	±30 mV
–10 dB	±10 mV
0 dB	±3 mV
+10 dB	±1 mV
+20 dB	±300 µV
+30, +40, +50, +60 dB	±100 µV

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

Flatness (referenced to 1 kHz) ..... 204.8 kS/s, DC coupling

0, +10, +20, +30, +40 dB..... ±0.1 dB, 0 to 95 kHz

–20, –10, +50, +60 dB ..... ±1 dB, 0 to 95 kHz,
±0.1 dB, 0 to 20 kHz

Frequency accuracy..... ±25 ppm

Output

None

Counter

Type .....DAQ-STC

Frequency accuracy .....± 100 ppm

NI 4454

Input

Channels 0, 1, 2, 3

Gain ranges .....0 dB

Maximum sample rate .....51.2 kS/s

Offset (residual DC) .....±3 mV

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

Flatness (relative to 1 kHz).....±0.1 dB, 0 to 23 kHz

Frequency accuracy .....±25 ppm

Output

None

Counter

None

NI 4551

Input

Channels 0, 1

Gain ranges .....–20, –10, 0, +10, +20, +30, +40, +50, +60 dB

Maximum sample rate .....204.8 kS/s

Offset (residual DC)

Offset (Residual DC)	Voltage
–20 dB	±30 mV
–10 dB	±10 mV
0 dB	± 3 mV
+10 dB	±1 mV
+20 dB	±300 mV
+30, +40, +50, +60 dB	±100 mV

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

Flatness (referenced to 1 kHz) ..... 204.8 kS/s, DC coupling

0, +10, +20, +30, +40 dB ..... ±0.1 dB, 0 to 95 kHz

–20, –10, +50, +60 dB ..... ±1 dB, 0 to 95 kHz,
±0.1 dB, 0 to 20 kHz

Frequency accuracy..... ±25 ppm

Output

Channels..... 0, 1

Gain ranges ..... –40, –20, 0 dB

Maximum update rate ..... 51.2 kS/s

Offset (residual DC)..... ±5 mV, any gain

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

Flatness..... 51.2 kS/s, any gain
±0.2 dB, 0 to 23 kHz

Frequency accuracy..... ±25 ppm

Counter

Type ..... TIO

Frequency accuracy..... ±100 ppm

Input

Channels 0, 1, 2, 3

Gain ranges .....–20, –10, 0, +10, +20, +30, +40, +50, +60 dB

Maximum sample rate .....204.8 kS/s

Offset (residual DC)

Offset (Residual DC)	Voltage
–20 dB	±30 mV
–10 dB	±10 mV
0 dB	±3 mV
+10 dB	±1 mV
+20 dB	±300 mV
+30, +40, +50, +60 dB	±100 mV

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

Flatness (referenced to 1 kHz).....204.8 kS/s, DC coupling

0, +10, +20, +30, +40 dB .....±0.1 dB, 0 to 95 kHz

–20, –10, +50, +60 dB .....±1 dB, 0 to 95 kHz, ±0.1 dB, 0 to 20 kHz

Frequency accuracy .....±25 ppm

Output

None

Counter

Type .....TIO

Frequency accuracy .....±100 ppm

Artisan Technology Group is an independent supplier of quality pre-owned equipment

Gold-standard solutions

Extend the life of your critical industrial, commercial, and military systems with our superior service and support.

We buy equipment

Planning to upgrade your current equipment? Have surplus equipment taking up shelf space? We'll give it a new home.

Learn more!

Visit us at **artisantg.com** for more info on price quotes, drivers, technical specifications, manuals, and documentation.

Artisan Scientific Corporation dba Artisan Technology Group is not an affiliate, representative, or authorized distributor for any manufacturer listed herein.

We're here to make your life easier. How can we help you today?

(217) 352-9330 | sales@artisantg.com | artisantg.com

ARTISAN® TECHNOLOGY GROUP