NI 5112

Introduction

This document contains information and step-by-step instructions for calibrating the National Instruments (NI) 5112 digitizer. This calibration procedure is intended for metrology labs. It includes instructions for self-calibrating the NI 5112 and for verifying its performance using a variety of programming environments.

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 for the device. This document describes self-calibration (or internal calibration) and external verification. External adjustment procedures are not included in this document because the calibration constants of the NI 5112 are not externally adjustable.

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 digitizer 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 verify the performance of the NI 5112 digitizer. NI recommends that 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.

370328B-01 *November 2001*

Equipment and Other Test Requirements

This section describes the equipment, test conditions, documentation, and software required for calibrating the NI 5112.

Test Equipment

Table 1. NI 5112 Calibration Equipment Specifications

Required Equipment	Recommended Equipment	Parameter Measured	Necessary Specifications
Signal Generator/ Ohmmeter	Fluke 9500B Oscilloscope Calibrator	Vertical Gain	DC ±25 mV to ±22.5 V, ±0.25% into 1 MΩ
		AC Coupling	sine wave 9–13 Hz ±100 ppm, 1.8 Vpp, ±2% into 1 MΩ
		Bandwidth	±2% amplitude flatness for leveled sine wave 100 kHz–100 MHz ±50 ppm, 1.5 Vpp, ±2% into 50 Ω
		Input Impedance	2-wire resistance accuracy of 0.25% for 50 Ω and 1 MΩ measurements
		Timing/RIS	sine wave 10 kHz–10 MHz ±15 ppm, 1.8 Vpp, ±2% into 1 MΩ
		Trigger Sensitivity	sine wave 100 kHz–10 MHz ±100 ppm, 300 mVpp, ±2% into 1 MΩ with CH0 and CH1; 750 mVpp with external trigger
5 1/2 Digit Digital Multimeter (DMM)	NI 4060	Internal Reference	DC voltage accuracy of ±0.25% (±12.5 mV) when measuring ±5 V
BNC Cable	—	—	50 Ω
BNC Shorting Cap	—	Vertical Offset	0 VDC, ±0.6 mV

Test Conditions

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

- Keep connections to the NI 5112 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, or consult the digitizer hardware manual for the optimum relative humidity.
- Maintain the temperature between 5 and 40 °C, or consult the digitizer hardware manual for the optimum temperature range.
- Allow a warm-up time of at least 15 minutes to ensure that the measurement circuitry of the NI 5112 is at a stable operating temperature.

Documentation

This section describes the documentation you need for self-calibrating and externally verifying the NI 5112 digitizer. In addition to this calibration document, you may find the following documents helpful:

- *NI 5112 User Manual*
- *Where to Start with Your NI Digitizer*
- *NI-SCOPE Quick Reference Guide*

You can download these documents from the NI Web site at `ni.com/manuals`.

Software

This section describes the software you need to calibrate the NI 5112. Calibration requires the latest version of the NI-SCOPE driver on the calibration system. The calibration functions are C function calls located in the NI-SCOPE instrument driver. These function calls are also valid for any compiler capable of calling a 32-bit DLL. Many of the functions use constants defined in the `niScopeCal.h` file. To use these constants, you must include `niScopeCal.h` in your code when you write your calibration procedure.

NI-SCOPE supports programming for all NI digitizers in the following programming languages: LabVIEW, Microsoft Visual C++, Microsoft Visual Basic, LabWindows/CVI, and Console C. You can download NI-SCOPE from the Instrument Driver Network at `ni.com/idnet`. To install NI-SCOPE, refer to the instructions in *Where to Start with Your NI Digitizer*.

Writing Your Calibration Procedure

NI-SCOPE 2.0 includes all functions necessary for calibrating NI digitizers. Because calibration support is included in `niScope_32.dll`, you can access it through any compiler capable of calling into a 32-bit DLL. If you use a C compiler, include the `niScopeCal.h` header file, which defines all calibration-specific functions and briefly explains the parameters. In Measurement Studio, the NI-SCOPE function panel `niScopeCal.fp` provides further help on these functions. LabVIEW support is installed in `niScopeCal.llb`, and all calibration functions appear in the function palette. See Table 2 for file locations.

Table 2. Calibration File Location After Installing NI-SCOPE 2.0 or Later

File Name and Location	Description
`VXIpnp\winnt(Win95)\Bin\niscope_32.dll`	NI-SCOPE driver containing the entire NI-SCOPE API, including calibration functions
`VXIpnp\winnt(Win95)\lib\msc\niscope_32.lib`	NI-SCOPE library containing the entire NI-SCOPE API, including calibration functions
`LabVIEW\examples\instr\niScopeExamples\`	Directory of LabVIEW NI-SCOPE example VIs, including self-calibration; access the examples from the LabVIEW function palette
`LabVIEW\instr.lib\Niscope\Calibrate\niscopeCal.llb`	LabVIEW VI library containing VIs for calling the NI-SCOPE calibration API; access calibration functions from the **NI-SCOPE Calibration** section of the LabVIEW function palette
`VXIpnp\winnt(Win95)\include\niscopeCal.h`	Calibration header file that you must include in any C program accessing calibration functions; this file automatically includes `niScope.h`, which defines the rest of the NI-SCOPE interface
`VXIpnp\winnt(Win95)\Niscope\Niscope.fp`	LabWindows/CVI function panel file that includes function prototypes and help using NI-SCOPE in the LabWindows/CVI environment
`VXIpnp\winnt(Win95)\Niscope\niScopeCal.fp`	LabWindows/CVI function panel file that includes external verification function prototypes and help on using NI-SCOPE in the LabWindows/CVI environment
`VXIpnp\winnt(Win95)\Niscope\Examples\`	Directory of NI-SCOPE examples for LabWindows/CVI, Console C, Visual C++, and Visual Basic

Self-Calibration Procedures

The NI 5112 includes an internal voltage source that is over 10 times as accurate as the 8-bit digitizer resolution. Self-calibration (or internal calibration) uses this internal reference source to do the following:

- Calibrate vertical range and offset for each input range
- Calibrate AC flatness over the entire bandwidth to within specifications
- Calibrate analog trigger levels
- Calibrate the time-to-digital converter (TDC) used for random interleaved sampling (RIS) measurements

The internal reference source is not user-adjustable, but you can verify the value of the source using a high-precision DMM to provide traceability. Absolute accuracy is ensured by verifying the internal reference voltage using a digital voltmeter.

Self-calibrate the NI 5112 digitizer before you perform an external verification. NI-SCOPE 2.0 includes self-calibration example programs for LabVIEW, Measurement Studio, and Console C. Table 2 shows the locations of these example programs.

Self-Calibrating the NI 5112 Digitizer

To self-calibrate the NI 5112, complete the following steps:

1. Call `niScope_init` to obtain an instrument session handle.
2. Call `niScope_calSelfCalibrate` with **option** set to `VI_NULL`. The new calibration constants are immediately stored in the EEPROM, so you can include this procedure in any application that uses the digitizer.
3. Call `niScope_close` to close the session handle and deallocate system resources.

Verification Procedures

This section describes how to externally verify the performance of the NI 5112 digitizer.

External verification of the NI 5112 tests the following specifications:

- Internal reference
- Vertical offset
- Vertical gain
- Input impedance
- Full bandwidth
- 20 MHz bandwidth
- AC coupling cutoff frequency
- Timing
- Trigger sensitivity
- RIS distribution

All verification procedures start by calling `niScope_init` with **resetDevice** set to `VI_TRUE`, and end by calling `niScope_close`.

The internal reference verification includes function calls to record the measured internal reference value in the EEPROM. External verification automatically stores the date and external verification count to provide traceability. External verification is equivalent to the factory production tests that verify the performance of the NI 5112.

Note Self-calibrate the NI 5112 before beginning these verification procedures. If any of the verification tests fail immediately after you self-calibrate the NI 5112, return it to NI for repair.

Verifying the Internal Reference

Complete the following steps to verify the internal reference:

1. Connect the DMM to the input on the front panel of the NI 5112.
2. Open a calibration session by calling `niScope_CalStart` with the calibration password, which is initially set to `0` or the empty string, `""`.
3. Route the internal reference out of the digitizer by calling `niScope_CalRouteInternalReference` with the following parameters:
    - **whichReference** = `NISCOPE_VAL_CAL_10V_CH0`
    - **option** = `NISCOPE_VAL_CAL_POSITIVE`

4. Measure the internal reference voltage with the DMM and record this value (*y* in this document).
5. Route the internal reference out of the digitizer by calling `niScope_CalRouteInternalReference`. Set the following parameters:
   - **whichReference** = `NISCOPE_VAL_CAL_10V_CH0`
   - **option** = `NISCOPE_VAL_CAL_NEGATIVE`
6. Measure the internal reference voltage with the DMM and record this value (*z* in this document).
7. Calculate the value of the onboard reference *x* using the following formula:

$$x = y - z$$

8. Compare *x* to the Success Condition value in Table 3. If the reference is outside of specification, return the digitizer to NI for repair.
9. If the digitizer passes the test, store the measured value to the driver by calling `niScope_CalStoreInternalReference`. Set the following parameters:
   - **whichReference** = `NISCOPE_VAL_CAL_10V_CH0`
   - **internalReference** = *x* from step 7
10. Call `niScope_CalRouteInternalReference` to write the internal reference value to the EEPROM. Set the following parameters:
    - **whichReference** = `NISCOPE_VAL_CAL_10V_CH0`
    - **option** = `NISCOPE_VAL_CAL_UNROUTE_SIGNAL`
11. Call `niScope_CalEnd` to end the verification session. Set **action** to `NISCOPE_VAL_CAL_ACTION_STORE`. Closing the calibration session stores the date and the incremented external verification count. While this value is not used by NI-SCOPE during operation, storing the value in the EEPROM provides traceability of the verification.

You have completed verifying the internal reference specifications for the NI 5112.

Table 3. NI 5112 Internal Reference Specifications

Digitizer Parameters	Success Condition
whichReference = `NISCOPE_VAL_CAL_10V_CH0` **option** = `NISCOPE_VAL_CAL_POSITIVE`	$9.99 < x < 10.01$ V

Verifying Vertical Offset

Complete the following steps to verify NI 5112 vertical offset specifications:

1. Short-circuit channel 0 on the front panel of the of the NI 5112 with the BNC shorting cap.
2. Call `niScope_ConfigureHorizontalTiming` with the following parameters:
   - **minSampleRate** = `100,000,000`
   - **minNumPts** = `30,000`
   - **refPosition** = `50.0`
   - **numRecords** = `1`
   - **enforceRealtime** = `VI_TRUE`
3. Call `niScope_ConfigureVertical` with the following parameters:
   - **channelList** = `0`
   - **range** = The Digitizer Parameters value in Table 4
   - **offset** = `0.0`
   - **coupling** = `NISCOPE_VAL_DC`
   - **probeAttenuation** = `1.0`
   - **enabled** = `VI_TRUE`
4. Wait 10 ms for the input stage to settle.
5. Call `niScope_InitiateAcquisition`.
6. Call `niScope_FetchMeasurement` with the following parameters:
   - **channelList** = `0`
   - **scalarMeasFunction** = `NISCOPE_VAL_VOLTAGE_AVERAGE`
   - **timeout** = `30`

   Compare the resulting average voltage to the Success Condition listed in Table 4. If the result is outside the Success Condition range, the digitizer has failed this portion of the verification. Return the digitizer to NI for repair.
7. Repeat steps 1 through 6 for each vertical offset entry in Table 4.
8. Move the shorting cap to channel 1.
9. Repeat steps 1 through 7 for channel 1. Change **channelList** to `1` when calling the functions `niScope_ConfigureVertical` and `niScope_fetchMeasurement`.

You have completed verifying the vertical offset specifications for the NI 5112.

Table 4. NI 5112 Vertical Offset Specifications

Name	Digitizer Parameters	Stimulus Parameters	Success Condition
Vertical Offset	**range** = 50 V	Short-Circuit Input	$\lvert x \rvert < 1.25$ V
Vertical Offset	**range** = 5 V	Short-Circuit Input	$\lvert x \rvert < 0.125$ V
Vertical Offset	**range** = 0.5 V	Short-Circuit Input	$\lvert x \rvert < 0.0125$ V

Verifying Vertical Gain

Complete the following steps to verify the NI 5112 vertical gain specifications:

1. Connect the signal generator to the channel 0 input of the digitizer.
2. Configure the signal generator for a 1 MΩ load.
3. Call `niScope_ConfigureHorizontalTiming` with the following parameters:
   - **minSampleRate** = `100,000,000`
   - **minNumPts** = `30,000`
   - **refPosition** = `50.0`
   - **numRecords** = `1`
   - **enforceRealtime** = `VI_TRUE`
4. Call `niScope_ConfigureVertical` with the following parameters:
   - **channelList** = `0`
   - **range** = The first Digitizer Parameters value in Table 5
   - **offset** = `0.0`
   - **coupling** = `NISCOPE_VAL_DC`
   - **probeAttenuation** = `1.0`
   - **enabled** = `VI_TRUE`
5. Wait 10 ms for the input stage to settle.
6. Apply the positive DC stimulus voltage listed under Stimulus Parameters in Table 5.
7. Call `niScope_InitiateAcquisition`.
8. Call `niScope_FetchMeasurement` with the following parameters:
   - **channelList** = `0`
   - **scalarMeasFunction =** `NISCOPE_VAL_VOLTAGE_AVERAGE`
   - **timeout** = `30`

9. Apply the negative DC stimulus voltage listed under Stimulus Parameters in Table 5.
10. Call `niScope_InitiateAcquisition`.
11. Call `niScope_FetchMeasurement` with the following parameters:
    - **scalarMeasFunction =** `NISCOPE_VAL_VOLTAGE_AVERAGE`
    - **channelList** = `0`
    - **timeout** = `30`
12. Calculate the error in the vertical gain using the following formula:

$$error = (a - b) - (c - d)$$

    where

    a is the measured positive voltage

    b is the measured negative voltage

    c is the applied positive voltage

    d is the applied negative voltage
13. Compare the value from step 12 to the Success Condition in Table 5. If the error is outside the range of the Success Condition, return the digitizer to NI for repair.
14. Repeat steps 2 through 13 for each vertical gain entry in Table 5.
15. Move the signal generator connection to the channel 1 input of the digitizer.
16. Repeat steps 2 through 14 for channel 1. Change **channelList** to `1` when calling the functions `niScope_ConfigureVertical` and `niScope_FetchMeasurement`.

You have completed verifying the vertical gain specifications for the NI 5112.

Table 5. NI 5112 Vertical Gain Specifications

Name	Digitizer Parameters	Stimulus Parameters	Success Condition
Vertical Gain	**range** = 50 V	±22.5 VDC	$\|x\| < 1.25$ V
Vertical Gain	**range** = 50 V	±5.0 VDC	$\|x\| < 1.25$ V
Vertical Gain	**range** = 5 V	±2.25 VDC	$\|x\| < 0.125$ V
Vertical Gain	**range** = 5 V	±0.25 VDC	$\|x\| < 0.125$ V
Vertical Gain	**range** = 0.5 V	±0.22 VDC	$\|x\| < 0.0125$ V
Vertical Gain	**range** = 0.5 V	±0.022 VDC	$\|x\| < 0.0125$ V

Verifying Full Bandwidth

Complete the following steps to verify the NI 5112 full bandwidth specifications:

1. Connect the signal generator to the channel 0 input of the digitizer.
2. Configure the signal generator for a 50 Ω load.
3. Set the signal generator to the frequency and amplitude listed under Stimulus Parameters in Table 6 for the Reference Full Bandwidth entry.
4. Call `niScope_ConfigureVertical` with the following parameters:
    - **channelList** = `0`
    - **range** = `2.0`
    - **offset** = `0`
    - **coupling** = `NISCOPE_VAL_DC`
    - **probeAttenuation** = `1.0`
    - **enabled** = `VI_TRUE`
5. Wait 300 ms for input stage to settle.
6. Call `niScope_ConfigureChanCharacteristics` with the following parameters:
    - **channelList** = `0`
    - **inputImpedance** = `NISCOPE_VAL_50_OHM`
    - **maxInputFrequency** = `0.0`
7. Call `niScope_ConfigureHorizontalTiming` with the following parameters:
    - **minSampleRate** = The Digitizer Parameters value for Reference Full Bandwidth in Table 6
    - **minNumPts** = `30,000`
    - **refPosition** = `50.0`
    - **numRecords** = `1`
    - **enforceRealtime** = `VI_TRUE`
8. Call `niScope_InitiateAcquisition`.
9. Call `niScope_FetchMeasurement` with the following parameters:
    - **channelList** = `0`
    - **scalarMeasFunction =** `NISCOPE_VAL_AC_ESTIMATE`
    - **timeout** = `30`

    Record this value to use as *reference AC estimate* in step 14.

10. Apply the signal specified in the first full bandwidth entry in Table 6.
11. Call `niScope_ConfigureHorizontalTiming` with the following parameters:
    - **minSampleRate** = The Digitizer Parameters value for full bandwidth in Table 6
    - **minNumPts** = `30,000`
    - **refPosition** = `50.0`
    - **numRecords** = `1`
    - **enforceRealTime** = `VI_TRUE`
12. Call `niScope_InitiateAcquisition`.
13. Call `niScope_FetchMeasurement` with the following parameters:
    - **scalarMeasFunction =** `NISCOPE_VAL_AC_ESTIMATE`
    - **channelList** = `0`
    - **timeout** = `30`

    Record this value to use as *AC estimate* in step 14.
14. Calculate the response in decibels using the following formula:

$$response = 20\log_{10}\left[\frac{AC\ estimate}{reference\ AC\ estimate}\right]$$

15. Compare the response to the Success Condition in Table 6. If the response is outside the range of the Success Condition, return the digitizer to NI for repair.
16. Repeat steps 10 through 15 for the remaining Full Bandwidth entries in Table 6.
17. Repeat steps 2 through 16 with **coupling** set to `NISCOPE_VAL_AC` when calling `niScope_ConfigureVertical`.
18. Move the signal generator connection to the channel 1 input of the digitizer.
19. Repeat steps 1 through 17 for channel 1. Change **channelList** to `1` for `niScope_ConfigureVertical`, `niScope_FetchMeasurement`, and `niScope_ConfigureChanCharacteristics`.

You have completed verifying the full bandwidth specifications for the NI 5112.

Table 6. NI 5112 Full Bandwidth Specifications

Name	Digitizer Parameters	Stimulus Parameters	Success Condition
Reference Full Bandwidth	**minSampleRate** = 20,000,000 S/s	100 kHz, 1.5 Vpp	—
Full Bandwidth	**minSampleRate** = 100,000,000 S/s	1 MHz, 1.5 Vpp	$\|x\| < 3$ dB
Full Bandwidth	**minSampleRate** = 50,000,000 S/s	49 MHz, 1.5 Vpp (intentionally aliased)	$\|x\| < 3$ dB
Full Bandwidth	**minSampleRate** = 100,000,000 S/s	99 MHz, 1.5 Vpp (intentionally aliased)	$\|x\| < 3$ dB

Verifying 20 MHz Bandwidth

Complete the following steps to verify the NI 5112 20 MHz bandwidth specifications:

1. Connect the signal generator to the channel 0 input of the digitizer.
2. Configure the signal generator for a 50 Ω load.
3. Set the signal generator to the frequency and amplitude listed in Table 7 for the 20 MHz reference bandwidth entry.
4. Call `niScope_ConfigureVertical` with the following parameters:
   - **channelList** = `0`
   - **range** = `2.0`
   - **offset** = `0`
   - **coupling** = `NISCOPE_VAL_DC`
   - **probeAttenuation** = `1.0`
   - **enabled** = `VI_TRUE`
5. Wait 300 ms for the input stage to settle.
6. Call `niScope_ConfigureChanCharacteristics` with the following parameters:
   - **channelList** = `0`
   - **inputImpedance** = `NISCOPE_VAL_50_OHM`
   - **maxInputFrequency** = `20,000,000`

7. Call `niScope_ConfigureHorizontalTiming` with the following parameters:
   - **minSampleRate** = The Digitizer Parameters value for Reference 20 MHz Bandwidth in Table 7
   - **minNumPts** = `30,000`
   - **refPosition** = `50.0`
   - **numRecords** = `1`
   - **enforceRealTime** = `VI_TRUE`
8. Call `niScope_InitiateAcquisition`.
9. Call `niScope_FetchMeasurement` with the following parameters:
   - **channelList** = `0`
   - **scalarMeasFunction** = `NISCOPE_VAL_AC_ESTIMATE`
   - **timeout** = `30`

   Record this value to use as *reference AC estimate* in step 14.
10. Apply the signal specified in the 20 MHz Bandwidth entry in Table 7.
11. Call `niScope_ConfigureHorizontalTiming` with the following parameters:
    - **minSampleRate** = The Digitizer Parameters value for the 20 MHz Bandwidth entry in Table 7
    - **minNumPts** = `30,000`
    - **refPosition** = `50.0`
    - **numRecords** = `1`
    - **enforceRealtime** = `VI_TRUE`
12. Call `niScope_InitiateAcquisition`.
13. Call `niScope_FetchMeasurement` with the following parameters:
    - **channelList** = `0`
    - **scalarMeasFunction** = `NISCOPE_VAL_AC_ESTIMATE`
    - **timeout** = `30`

    Record this value to use as *AC estimate* in step 14.
14. Calculate the response in decibels using the following formula:

$$response = 20\log_{10}\left[\frac{AC\ estimate}{reference\ AC\ estimate}\right]$$

15. Compare the response to the Success Condition in Table 7. If the response is outside the range of the Success Condition, return the digitizer to NI for repair.

16. Repeat steps 10 through 15 for the remaining bandwidth entries in Table 7.
17. Repeat steps 2 through 16 with **coupling** set to `NISCOPE_VAL_AC` for the function `niScope_ConfigureVertical`.
18. Move the signal generator connection to the channel 1 input on the front panel of the digitizer.
19. Repeat steps 2 through 17 for channel 1. Change **channelList** to `1` for `niScope_ConfigureVertical`, `niScope_FetchMeasurement`, and `niScope_ConfigureChanCharacteristics`.

You have completed verifying the 20 MHz bandwidth specifications for the NI 5112.

Table 7. NI 5112 20 MHz Bandwidth Specifications

Name	Digitizer Parameters	Stimulus Parameters	Success Condition
Reference 20 MHz Bandwidth	**minSampleRate** = 20,000,000 S/s	100 kHz, 1.5 Vpp	—
20 MHz Bandwidth	**minSampleRate** = 100,000,000 S/s	1 MHz, 1.5 Vpp	$\|x\| < 3$ dB
20 MHz Bandwidth	**minSampleRate** = 20,000,000 S/s	19.9 MHz, 1.5 Vpp (intentionally aliased)	$\|x\| < 3$ dB
20 MHz Bandwidth	**minSampleRate** = 20,000,000 S/s	21 MHz, 1.5 Vpp (intentionally aliased)	$\|x\| > 3$ dB

Verifying Input Impedance

Complete the following steps to verify NI 5112 input impedance specifications:

1. Connect the ohmmeter to the channel 0 input of the digitizer.
2. Call `niScope_ConfigureVertical` with the following parameters:
   - **channelList** = `0`
   - **range** = The Digitizer Parameters value in Table 8
   - **offset** = `0.0`
   - **coupling** = `NISCOPE_VAL_DC`
   - **probeAttenuation** = `1.0`
   - **enabled** = `VI_TRUE`
3. Wait 10 ms for the input stage to settle.

4. Call `niScope_ConfigureChanCharacteristics` with the following parameters:
   - **channelList** = `0`
   - **inputImpedance** = The Digitizer Parameters value in Table 8
   - **maxInputFrequency** = `0.0`
5. Call `niScope_Read` to ensure the hardware is programmed. Set the following parameters:
   - **channelList** = `0`
   - **timeout** = `30`
   - **numSamples** = `128`
6. Measure the impedance with the ohmmeter and compare it to the Success Condition in Table 8. If the impedance is outside the range of the Success Condition, return the digitizer to NI for repair.
7. Repeat steps 2 through 6 for each input impedance entry in Table 8.
8. Move the ohmmeter connection to the channel 1 of the digitizer.
9. Repeat steps 2 through 7 for channel 1. Change **channelList** to `1` when calling the functions `niScope_configureVertical` `niScope_ConfigureChanCharacteristics`, and `niScope_Read`.

You have completed verifying the input impedance specifications for the NI 5112.

Table 8. NI 5112 Input Impedance Specifications

Name	Digitizer Parameters	SuccessCondition
Input Impedance	**range** = 40.0 V **inputImpedance** = `NISCOPE_VAL_1_MEG_OHM`	990,000 < x < 1,010,000 Ω
Input Impedance	**range** = 4.0 V **inputImpedance** = `NISCOPE_VAL_1_MEG_OHM`	990,000 < x < 1,010,000 Ω
Input Impedance	**range** = 0.4 V **inputImpedance** = `NISCOPE_VAL_1_MEG_OHM`	990,000 < x < 1,010,000 Ω
Input Impedance	**range** = 4.0 V **inputImpedance** = `NISCOPE_VAL_50_OHM`	49.5 < x < 50.5 Ω
Input Impedance	**range** = 0.4 V **inputImpedance** = `NISCOPE_VAL_50_OHM`	49.5 < x < 50.5 Ω

Verifying AC Coupling Cutoff Frequency

Complete the following steps to verify NI 5112 AC coupling specifications:

1. Connect a the signal generator to channel 0 of the digitizer.
2. Configure the signal generator for a 1 MΩ load.
3. Set the signal generator to the frequency and amplitude listed in the Stimulus Parameters column in Table 9.
4. Call `niScope_ConfigureVertical` with the following parameters:
   - **channelList** = `0`
   - **range** = `2.0`
   - **offset** = `0.0`
   - **coupling** = `NISCOPE_VAL_DC`
   - **probeAttenuation** = `1.0`
   - **enabled** = `VI_TRUE`
5. Wait 10 ms for the input stage to settle.
6. Call `niScope_ConfigureHorizontalTiming` with the following parameters:
   - **minSampleRate** = `10,000`
   - **minNumPts** = `10,000`
   - **refPosition** = `50.0`
   - **numRecords** = `1`
   - **enforceRealtime** = `VI_TRUE`
7. Call `niScope_InitiateAcquisition`.
8. Call `niScope_FetchMeasurement` with the following parameters:
   - **channelList** = `0`
   - **scalarMeasFunction** = `NISCOPE_VAL_AC_ESTIMATE`
   - **timeout** = `30`

   Record this value to use as *AC estimate with DC coupling* in step 13.
9. Call `niScope_ConfigureVertical` with the following parameters:
   - **channelList** = `0`
   - **range** = `2.0`
   - **offset** = `0.0`
   - **coupling** = `NISCOPE_VAL_AC`
   - **probeAttenuation** = `1.0`
   - **enabled** = `VI_TRUE`
10. Wait 300 ms for the input stage to settle.

11. Call `niScope_InitiateAcquisition`.
12. Call `niScope_FetchMeasurement` with the following parameters:
    - **channelList** = `0`
    - **scalarMeasFunction** = `NISCOPE_VAL_AC_ESTIMATE`
    - **timeout** = `30`

    Record this value to use as *AC estimate with AC coupling* in step 13.
13. Calculate the response in decibels using the following formula:

$$response = 20\log_{10}\left[\frac{AC\ estimate\ with\ AC\ coupling}{AC\ estimate\ with\ DC\ coupling}\right]$$

14. Compare the response to the Success Condition in Table 9. If the response is outside the listed range, return the digitizer to NI for repair.
15. Repeat steps 2 through 14 for each AC coupling entry in Table 9.
16. Move the signal generator connection to the channel 1 input of the digitizer.
17. Repeat steps 2 through 15 for channel 1. Change **channelList** to `1` when calling the functions `niScope_ConfigureVertical` and `niScope_FetchMeasurement`.

You have completed verifying the AC coupling cutoff frequency specifications for the NI 5112.

Table 9. NI 5112 AC Coupling Specifications

Name	Stimulus Parameters	Success Condition
AC Coupling	12.1 Hz, 1.8 Vpp	$\lvert x \rvert < 3$ dB
AC Coupling	9.9 Hz, 1.8 Vpp	$\lvert x \rvert > 3$ dB

Verifying Timing

Complete the following steps to verify the NI 5112 timing specifications:

1. Connect the signal generator to the channel 0 input of the digitizer.
2. Configure the signal generator for a 1 MΩ load.
3. Generate a 10 kHz, 1.8 Vpp sine wave.

4. Call `niScope_ConfigureVertical` with the following parameters:
   - **channelList =** `0`
   - **range** = `2.0`
   - **offset** = `0.0`
   - **coupling** = `NISCOPE_VAL_DC`
   - **probeAttenuation** = `1.0`
   - **enabled** = `VI_TRUE`
5. Wait 10 ms for the input stage to settle.
6. Call `niScope_ConfigureHorizontalTiming` with the following parameters:
   - **minSampleRate** = `1,000,000`
   - **minNumPts** = `100,000`
   - **refPosition** = `50.0`
   - **numRecords** = `1`
   - **enforceRealtime** = `VI_TRUE`
7. Call `niScope_InitiateAcquisition`.
8. Call `niScope_FetchMeasurement` with the following parameters:
   - **channelList** = `0`
   - **scalarMeasFunction** = `NISCOPE_VAL_AVERAGE_FREQUENCY`
   - **timeout** = `30`
9. If the returned frequency value does not fall between 9999 and 10,001 Hz, a hardware error exists. If the digitizer fails this step, terminate the verification procedure and return it to NI for repair.
10. Generate a 1.8 Vpp, 10 MHz sine wave. This wave is intentionally undersampled, where the sampling rate is an even multiple of the sine wave frequency.
11. Call `niScope_InitiateAcquisition`.
12. Call `niScope_FetchMeasurement` with the following parameters:
    - **channelList** = `0`
    - **scalarMeasFunction** = `NISCOPE_VAL_AVERAGE_PERIOD`
    - **timeout** = `30`

    Record the *period* measurement to use in step 14.
13. If the returned status is `NISCOPE_ERROR_UNABLE_TO_PERFORM_MEASUREMENT`, call `niScope_errorHandler` with **errorCode** set to the returned error value. If the timing is perfectly aliased, the waveform is a DC level and the period measurement fails. Therefore, if the error description

indicates the measurement failed due to insufficient crosspoints, the digitizer passed the test.

14. If the returned status is anything other than `NISCOPE_ERROR_UNABLE_TO_PERFORM_MEASUREMENT`, calculate the actual sample rate (x), assuming a perfect source, with the following formula:

$$x = \frac{specified\ sample\ rate \times source\ frequency \times period}{source\ frequency \times period - 1}$$

which is:

$$x = \frac{10^{13} \times period}{10^{7} \times period - 1}$$

15. Compare the actual sample rate (x) to the Success Condition, $999{,}950 < x < 1{,}000{,}050$ Hz. If x is outside the Success Condition range, return the digitizer to NI for repair.
16. Move the signal generator connection to the channel 1 input of the digitizer.
17. Repeat steps 2 through 15 for channel 1. Change **channelList** to `1` when calling `niScope_ConfigureVertical` and `niScope_FetchMeasurement`.

You have completed verifying the timing specifications for the NI 5112.

Verifying Trigger Sensitivity

To verify trigger sensitivity, you must determine the smallest signal on which the digitizer can trigger by trying all possible trigger levels. Complete the following steps:

1. Connect the signal generator to the trigger channel input of the digitizer.
2. Configure the signal generator for a 1 MΩ load.
3. Apply a 1 MHz sine wave with zero vertical offset, and peak-to-peak voltage as listed in Table 10.
4. Call `niScope_ConfigureVertical` with the following parameters:
   - **channelList** = `0`
   - **range** = `20`
   - **offset** = `0.0`
   - **coupling** = `NISCOPE_VAL_DC`
   - **probeAttenuation** = `1.0`
   - **enabled** = `VI_TRUE`

5. Wait 10 ms for the input stage to settle.
6. Call `niScope_ConfigureHorizontalTiming` with the following parameters:
   - **minSampleRate** = `20,000,000`
   - **minNumPts** = `128`
   - **refPosition** = `50.0`
   - **numRecords** = `1`
   - **enforceRealtime** = `VI_TRUE`
7. Call `niScope_ConfigureTriggerEdge` with the following parameters:
   - **triggerSource** = `NISCOPE_VAL_EXTERNAL`
   - **level** = The low trigger level from Table 10
   - **slope** = `NISCOPE_VAL_POSITIVE`
   - **triggerCoupling** = `NISCOPE_VAL_AC`
   - **holdoff** = `0`
   - **delay** = `0`
   - **channelList** = `0`
   - **timeout** = `1`
8. Call `niScope_Read` to read a waveform with the following parameters:
   - **channelList =** `0`
   - **timeout** = `0.1`
   - **numSamples** = `128`

   If this function returns a maximum time exceeded error, proceed to step 9. If the digitizer did not time out, skip to step 11.
9. Call `niScope_Abort` to stop the test.
10. If the digitizer timed out, increment the **level** setting by the trigger level delta specified in Table 10, then repeat step 7. The digitizer fails this test if incrementing the trigger level causes the trigger sensitivity to be higher than the high trigger level entry in Table 10. If the digitizer fails the test, return it to NI for repair. Otherwise, repeat steps 7 and 8 for the remaining trigger levels.
11. Repeat steps 2 through 10 to test the trigger sensitivity on channel 0. Make the following changes:
    - Change **channelList** to `0` when calling `niScope_Read` and `niScope_ConfigureVertical`
    - Change **triggerSource** to `0` when calling `niScope_ConfigureTriggerEdge`

12. Repeat steps 2 through 11 to test the trigger sensitivity on channel 1. Make the following changes:
    - Change **channelList** to `1` when calling `niScope_Read` and `niScope_ConfigureVertical`
    - Change **triggerSource** to `1` when calling `niScope_ConfigureTriggerEdge`

The digitizer passes the trigger sensitivity test if all channels pass the trigger susceptivity test. If any channel fails the test, return the digitizer to NI for repair.

You have completed verifying the trigger sensitivity for the NI 5112.

Table 10. NI 5112 Trigger Sensitivity Specifications

Name	Digitizer Parameters	Stimulus Parameters	Success Condition
Trigger Sensitivity	**low trigger level** = –5.0 **high trigger level** = 5.0 **trigger level delta** = 0.02	750 mVpp	digitizer triggers with any valid trigger level

Verifying Random Interleaved Sampling Distribution (RIS)

The TDC provides an extremely accurate trigger time resolution between two samples. This trigger should happen with a uniform distribution between two digitizer samples to accurately reconstruct the periodic signal. This method of trigger distribution is called RIS. Complete the following steps to measure RIS:

1. Connect the signal generator to the channel 0 input of the digitizer. This test requires a signal generator that is completely independent of the digitizer. The source cannot be a signal derived from the digitizer, and it cannot be the output of a function generator that is synchronized with the digitizer.
2. Configure the signal generator for a 1 MΩ load.
3. Apply the signal listed in the Stimulus Parameters column of Table 11.
4. Call `niScope_ConfigureVertical` with the following parameters:
    - **channelList** = `0`
    - **range** = `2`
    - **offset** = `0`
    - **coupling** = `NISCOPE_VAL_DC`
    - **probeAttenuation** = `1.0`
    - **enabled** = `VI_TRUE`

5. Call `niScope_ConfigureHorizontalTiming` with the following parameters:
   - **minSampleRate** = `100,000,000`
   - **minNumPts** = `128`
   - **refPosition** = `50.0`
   - **numRecords** = `1`
   - **enforceRealtime** = `VI_TRUE`
6. Call `niScope_ConfigureTriggerEdge` with the following parameters:
   - **triggerSource** = `0`
   - **level** = `0`
   - **slope** = `NISCOPE_VAL_POSITIVE`
   - **triggerCoupling** = `NISCOPE_VAL_DC`
   - **holdoff** = `0`
   - **delay** = `0`
7. Call `niScope_CalMeasureRISDistribution` with the following parameters:
   - **channelName** = `0`
   - **distributionSize** = The Digitizer Parameters value in Table 11
   - **maxTime** = `10,000`
   - **distribution** = A pointer to an array of **distributionSize** number of elements

   If you do not want **distribution** returned, set **distribution** to `VI_NULL`. The function `niScope_CalMeasureRISDistribution` performs 2,000 acquisitions and creates a probability distribution based on the initial *x* value, which includes the TDC value.
8. Compare the returned **minimumBinPercent** (*x*) to the Success Condition in Table 11. If the returned value is outside the range of the Success Condition, return the digitizer to NI for repair.
9. Move the signal generator connection to the channel 1 input of the digitizer.
10. Repeat steps 2 through 8 for channel 1 with the following changes:
    - Change **channelList** to `1` when calling `niScope_ConfigureVertical`
    - Change **triggerSou**rce to `1` when calling `niScope_ConfigureTriggerEdge`
    - Change **channelName** to `1` when calling `niScope_CalMeasureRISDistribution`

You have completed verifying the RIS distribution for the NI 5112.

Table 11. NI 5112 RIS Distribution Specifications

Digitizer Parameters	StimulusParameters	Success Condition
distributionSize = 25	1 MHz, ±100 kHz, 1.8 Vpp	$x > 0.8$

Note If the NI 5112 digitizer fails any of the verification procedures immediately after a self-calibration, return it to NI for repair or replacement.

Calibration Function Reference

This section lists functions specific to NI-SCOPE calibration. Refer to *Where to Start with Your NI Digitizer* for instructions on how to install NI-SCOPE.

niScope_CalStart

Function Prototype

```
ViStatus _VI_FUNC niScope_CalStart
(
    ViRsrc resourceName,
    ViConstString password,
    ViSession *newSessionHandle
);
```

Purpose

niScope_CalStart opens an external verification session.

Using This Function

For additional security, this function compares **password** to the password stored in the EEPROM. By default, the password is set to NULL, or the empty string, "". The password is stored in the EEPROM as an array of 4 bytes. Non-printable characters are allowed, but the array is padded with NULLs after the first NULL is found. This padding allows strings of less than four characters to be legal passwords. If you forget your password, call NI.

All calibration functions require a session handle, such as **newSessionHandle**, that is returned by this function. Only the external verification functions require a calibration session handle for password protection. All other functions work with both a calibration session and a session handle obtained from niScope_init. Acceptable session handles are documented for each function in this section.

Only one session handle can be obtained at a time, and every session must be closed by calling niScope_CalEnd. If you fail to close the session, you must unload the niScope_32.dll by closing your application or development environment before you can open another session.

If an error occurs during calibration, call niScope_errorHandler to get the error message text and niScope_CalEnd with **action** set to NISCOPE_VAL_CAL_ACTION_ABORT to close the session.

Parameters

Name	Description
resourceName	assigned by Measurement and Automation Explorer (MAX), this is a string such as "DAQ::1"
password	compared to password in EEPROM
newSessionHandle	returned session handle

niScope_CalEnd

Function Prototype

```
ViStatus _VI_FUNC niScope_CalEnd
(
   ViSession sessionHandle,
   ViInt32 action
);
```

Purpose

`niScope_CalEnd` closes an external verification session.

Using This Function

If **action** is `NISCOPE_VAL_CAL_ACTION_ABORT`, the session closes and the calibration constants are lost. The abort close is necessary when an error occurs during calibration. Some devices write to the EEPROM during calibration, in which case the **abort** parameter restores the EEPROM to its original state. It is, therefore, very important to call `niScope_CalEnd` each time `niScope_CalStart` is called, even if an error occurs during calibration.

If **action** is `NISCOPE_VAL_CAL_ACTION_STORE`, the calibration constants are stored in the EEPROM. If you call `niScope_CalStoreMiscInfo` during the calibration session, the miscellaneous information is stored. Otherwise, the miscellaneous information is set to `0` or the empty string, `""`, in the EEPROM. The current system date and an incremented external verification count are automatically stored in the EEPROM.

Parameters

Name	Description
sessionHandle	session handle returned by `niScope_CalStart`
action	`NISCOPE_VAL_CAL_ACTION_STORE`, `NISCOPE_VAL_CAL_ACTION_ABORT`

niScope_CalChangePassword

Function Prototype

```
ViStatus _VI_FUNC niScope_CalChangePassword
(
    ViSession sessionHandle,
    ViConstString oldPassword,
    ViConstString newPassword
```

Purpose

To use `niScope_CalChangePassword`, you must enter an old password and a new password. The function verifies your old password against the one stored in the EEPROM. If they match, the new password is stored in the EEPROM.

Using This Function

The password can be four characters long, but shorter strings are acceptable. Non-printable values are acceptable, but zero is treated as an end-of-string character. If a zero (or end-of-string marker) is detected, zeros are added to the end to make the string four characters long.

By default, the password in the EEPROM is an array of nulls, or the empty string, `""`. If you forget your password, call NI.

Parameters

Name	Description
sessionHandle	session handle returned by `niScope_CalStart` or `niScope_init`
oldPassword	value currently stored in EEPROM (factory default is the empty string, `""`)
newPassword	new value to store in EEPROM

niScope_CalFetchCount

Function Prototype

```
ViStatus _VI_FUNC niScope_CalFetchCount
(
   ViSession sessionHandle
   ViInt32 whichOne,
   ViInt32 *calibrationCount
);
```

Purpose

`niScope_CalFetchCount` returns the calibration count, which is the number of times the digitizer has been calibrated.

Using This Function

whichOne determines whether to return the self-calibration or the external verification count. Possible values are defined in `niScopeCal.h`.

Parameters

Name	Description
sessionHandle	session handle returned by `niScope_CalStart` or `niScope_init`
whichOne	`NISCOPE_VAL_CAL_SELF`, `NISCOPE_VAL_CAL_EXTERNAL`
calibrationCount	number of times digitizer has been calibrated

niScope_CalFetchDate

Function Prototype

```
ViStatus _VI_FUNC niScope_CalFetchDate
(
   ViSession sessionHandle
   ViInt32 whichOne,
   ViInt32 *year,
   ViInt32 *month,
   ViInt32 *day
);
```

Purpose

`niScope_CalFetchDate` returns the self-calibration or external verification date.

Using This Function

If you are upgrading to NI-SCOPE 2.0 from an earlier version, the initial calibration dates are incorrect because older versions of NI-SCOPE do not support the date feature.

Parameters

Name	Description
sessionHandle	session handle returned by `niScope_CalStart` or `niScope_init`
whichOne	`NISCOPE_VAL_CAL_SELF`, `NISCOPE_VAL_CAL_EXTERNAL`, or `NISCOPE_VAL_CAL_MANUFACTURE`
year	returned year of last calibration (for example, 2000)
month	returned month of last calibration (1–12)
day	returned day of last calibration (1–31)

niScope_CalFetchMiscInfo

Function Prototype

```
ViStatus _VI_FUNC niScope_CalFetchMiscInfo
(
   ViSession sessionHandle,
   ViChar *info
);
```

Purpose

`niScope_CalFetchMiscInfo` returns the miscellaneous information you can store during an external verification using `niScope_StoreMiscInfo`.

Using This Function

info must be a character array of length five bytes. The fifth byte is always set to `NULL` to terminate the string.

Parameters

Name	Description
sessionHandle	session handle returned by `niScope_CalStart` or `niScope_init`
info	array of 5 bytes (4 bytes of information plus 1 `NULL` byte) stored in EEPROM during last external verification

niScope_CalFetchInternalReference

Function Prototype

```
ViStatus _VI_FUNC niScope_CalFetchInternalReference
(
   ViSession sessionHandle,
   ViInt32 whichReference,
   ViReal64 *internalReference
);
```

Purpose

`niScope_CalFetchInternalReference` returns the internal reference value that is stored by the `niScope_CalStoreInternalReference` function during external verification. This function enables tracking of the internal reference value, although the value is not used during digitizer operation or self-calibration.

Using This Function

The internal reference is stored as a 32-bit floating point number in the EEPROM.

Parameters

Name	Description
sessionHandle	session handle returned by `niScope_CalStart` or `niScope_init`
whichReference	`NISCOPE_VAL_CAL_10V_CH0`
internalReference	returned value of internal reference stored during the last external verification

niScope_CalStoreMiscInfo

Function Prototype

```
ViStatus _VI_FUNC niScope_CalStoreMiscInfo
(
   ViSession sessionHandle,
   ViConstString info
);
```

Purpose

`niScope_CalStoreMiscInfo` stores miscellaneous information in the EEPROM during external verification, such as an operator ID for the person or company calibrating the digitizer.

Using This Function

If this function is not called during an external verification, the miscellaneous information is set to `NULL` in the EEPROM. This setting ensures a consistent calibration date, count, and miscellaneous information values in the EEPROM.

Four bytes are stored in the EEPROM, and non-printable characters are valid. However, `NULL` is treated as an end of string marker, and all bytes following the first `NULL` are set to `NULL`.

Parameters

Name	Description
sessionHandle	session handle returned by `niScope_CalStart`
info	array of 4 bytes stored in EEPROM during external verification

niScope_CalStoreInternalReference

Function Prototype

```
ViStatus _VI_FUNC niScope_CalStoreInternalReference
(
   ViSession sessionHandle,
   ViInt32 whichReference,
   ViReal64 internalRefValue
);
```

Purpose

`niScope_CalStoreInternalReference` stores the measured internal reference voltage value.

Using This Function

The NI 5112 supports routing the internal reference to the front BNC connectors with the function `niScope_CalRouteInternalReference`. You can measure with a DMM to verify that the voltage reference value is within specification. This function stores the measured value in the EEPROM so you can track any drift over time. The value is stored for tracking purposes only and is not used during digitizer operation or calibration. The reference value is stored in the EEPROM as a 32-bit floating point number.

`niScope_CalStoreInternalReference` requires an external verification session, which you open with `niScope_CalStart`. The value is not written to the EEPROM until `niScope_CalEnd` is called with **action** set to `NISCOPE_VAL_CAL_ACTION_STORE`. If you do not call `niScope_CalStoreInternalReference` during an external verification session, the internal reference is set to zero in the EEPROM when you call `niScope_CalEnd` with **action** set to `NISCOPE_VAL_CAL_ACTION_STORE`. This ensures consistent calibration count, date, and internal reference values in the EEPROM.

Parameters

Name	Description
sessionHandle	session handle returned by `niScope_CalStart`
whichReference	`NISCOPE_VAL_CAL_10V_CH0`
internalRefValue	reference value to store

niScope_CalSelfCalibrate

Function Prototype

```
ViStatus _VI_FUNC niScope_CalSelfCalibrate
(
   ViSession sessionHandle,
   ViConstString channelName,
   ViInt32 option
);
```

Purpose

`niScope_CalSelfCalibrate` performs a self-calibration.

Using This Function

This function self-calibrates the digitizer. If the self-calibration is successful, the calibration constants are immediately stored in the self-calibration area of the EEPROM, along with the self-calibration date and incremented count.

Parameters

Name	Description
sessionHandle	session handle returned by `niScope_CalStart` or `niScope_init`
channelName	this parameter is currently ignored; use `VI_NULL`
option	`NISCOPE_VAL_SELF_CALIBRATION`

niScope_CalRouteInternalReference

Function Prototype

```
ViStatus _VI_FUNC niScope_CalRouteInternalReference
(
   ViSession sessionHandle,
   ViInt32 option,
   ViInt32 whichReference
);
```

Purpose

For use in measuring the internal reference, `niScope_CalRouteInternalReference` routes the NI 5112 internal reference to the front panel BNC connector of the specified channel.

Using This Function

Refer to the *Verifying the Internal Reference* section for details on using the function `niScope_CalRouteInternalReference`. Also refer to the *Verification Procedures* section for details on how to verify the performance of the NI 5112.

Note Be sure to unroute the signal after you are finished by calling `niScope_CalRouteInternalReference` again with **option** set to `NISCOPE_VAL_CAL_UNROUTE_SIGNAL`.

Parameters

Name	Description
sessionHandle	session handle returned by `niScope_CalStart`
option	`NISCOPE_VAL_CAL_UNROUTE_SIGNAL`, `NISCOPE_VAL_CAL_POSITIVE`, `NISCOPE_VAL_CAL_NEGATIVE`
whichReference	`NISCOPE_VAL_CAL_10V_CH0`

niScope_CalMeasureRISDistribution

Function Prototype

```
ViStatus _VI_FUNC niScope_CalMeasureRISDistribution
(
    ViSession sessionHandle,
    ViConstString channelName,
    ViInt32 maxTime,
    ViReal64 *minimumBinPercent,
    ViInt32 distributionSize,
    ViInt32 *distribution
);
```

Purpose

`niScope_CalMeasureRISDistribution` calls `niScope_Read` 2,000 times to acquire data from the specified channel and retrieve the initial *x* value, which includes the TDC.

Using This Function

The TDC is a uniform distribution between two sample points because triggers occur randomly. To test this uniformity, the distribution of initial *x* values is created. Compare the percentage of triggers in the smallest bin of this distribution to a specification to determine if RIS is operating correctly. The distribution parameter must be declared as an array of **distributionSize.** Optionally, you can set **distribution** to `VI_NULL` to specify that the distribution should not be returned.

Parameters

Name	Description
sessionHandle	session handle returned by `niScope_CalStart` or `niScope_init`
channelName	string name of channel whose internal reference value you are storing (for example, `0` or `1`)
maxTime	maximum number of milliseconds each acquisition can take
minimumBinPercent	percent of triggers (0.0–100.0) that fall in the smallest bin
distributionSize	number of bins to use for the distribution of initial *x* values; see the *Verification Procedures* section for details about using this parameter
distribution	array of **distributionSize** for the returned distribution, or `VI_NULL` if the distribution should not be returned