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 specific programming instructions for externally calibrating the NI 5112 using Measurement Studio, LabVIEW, C, or Visual Basic programming environments.

What Is Calibration?

Calibration consists of verifying the measurement accuracy of a device and correcting for any measurement error. *Verification* is measuring the performance of a device and comparing the results to the factory specifications of the device. NI calibrates every NI 5112 digitizer at the factory. During the factory calibration process, the calibration constants are stored on the EEPROM. These values are loaded from memory and used as needed by the digitizer.

NI digitizers have three types of calibration: external, internal, and external restore. These three types of calibration are described in the next three sections.

External Calibration

External calibration is generally performed with a high-precision oscilloscope calibrator at either NI or a metrology lab. This procedure replaces all calibration constants in the EEPROM and is equivalent to a factory calibration at NI. Because the external calibration procedure changes all EEPROM constants, it invalidates the original National Institute of Standards and Technology (NIST)-traceability certificate. If an external calibration is done with a NIST-certified voltage source, a new NIST-traceability certificate can be issued.

370328A-01 *June 2001*

Self-Calibration

Self-calibration, or internal calibration, uses a software command and requires no external connections.

External Restore Calibration

External restore calibration restores the previous external calibration settings for any calibration constants computed during external calibration or self-calibration. You should use this option only during a self-calibration failure. For example, if you lose power during a self-calibration, the digitizer may be rendered unusable. External restore allows the digitizer to function for a self-calibration.

Why Should You Calibrate?

The accuracy of electronic components drifts with time and temperature, which can affect measurement accuracy as a device ages. Calibration restores your 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 externally calibrate your 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.

You can also use the verification procedure at a regular interval to determine if your digitizer needs adjustment.

Equipment and Other Test Requirements

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

Test Equipment

Table 1 lists the equipment required for calibrating your NI 5112. If you do not have the recommended instruments, use these specifications to select a substitute calibration standard.

Table 1. Required Equipment Specifications for NI 5112 Verification and Calibration

Required Equipment	Recommended Equipment	Parameter Measured	Necessary Specifications
Signal Generator/ Ohmmeter	Wavetek 9500 Scope 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 Short-Circuiting Cap	—	Vertical Offset	0 VDC ±0.6 mV

Test Conditions

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

- 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 your digitizer hardware manual for the optimum relative humidity.
- Maintain the temperature between 5 and 40 °C, or consult your 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 to calibrate your NI 5112 digitizer. In addition to this calibration procedure, you may need to refer to the following documents:

- *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 your NI 5112 digitizer. Unless otherwise specified, calibration functions are C function calls in the NI-SCOPE driver, which you can download from the NI Web site at `ni.com`. These function calls are also valid for any compiler capable of calling a 32-bit DLL. While LabVIEW virtual instruments (VIs) are not discussed in this procedure because many LabVIEW VIs have the same names as the listed NI-SCOPE function calls. 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.

Calibration requires the latest version of the NI-SCOPE driver on the calibration system. You can download NI-SCOPE from the Instrument Driver Network at the NI web site at `ni.com/idnet`. NI-SCOPE supports programming for all NI digitizers using a number of languages, including LabVIEW, Microsoft Visual C++, and Microsoft Visual Basic.

To install the calibration software, complete the following steps:

1. Launch the NI-SCOPE installer from the NI-SCOPE CD.
2. Select **Programmatic and Interactive Support** as the installation type.
3. Click on the **Components** button in lower left corner of the Application Development Environments screen.
4. Check the box labelled **External Calibration Support**.

Complete the installation of NI-SCOPE by following the instructions in *Where to Start with Your NI 5112 Oscilloscope*.

Writing Your Calibration Procedure

The latest version of NI-SCOPE includes all the 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 DLL. If you use a C compiler, include the `niScopeCal.h` header file, which defines all calibration-specific functions and briefly explains the parameters. With 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 1.6 or Later

File Name and Location	Description
`vxipnp\winXX\bin\niScope_32.dll`	NI-SCOPE driver containing the entire NI-SCOPE API, including calibration functions
`vxipnp\winXX\lib\msc\niScope_32.lib`	NI-SCOPE library containing the entire NI-SCOPE API, including calibration functions
`LabView\examples\instr\niScopeExamples`	LabVIEW VI library containing NI-SCOPE examples, including self calibration; you can access calibration examples from the function palette in LabVIEW
`LabView\instr.lib\niscope\calibrateniScopeCal.llb`	LabVIEW VI library containing VIs for calling the NI-SCOPE calibration API; you can access calibration functions from the function palette in LabVIEW
`vxipnp\winXX\include\niScopeCal.h`	Calibration header file, which must be included in any C program accessing calibration functions; this file automatically includes `niScope.h`, which defines the rest of the NI-SCOPE interface
`vxipnp\winXX\niScope\niScope.fp`	CVI function panel file, which includes function prototypes and help using NI-SCOPE in the CVI environment
`vxipnp\winXX\niScope\niScopeCal.fp`	CVI function panel file, which includes function prototypes and help using NI-SCOPE in the CVI environment for external calibration
`vxipnp\winXX\niScope\examples`	Directory of NI-SCOPE examples for CVI, MSVC, Visual C++, and Visual Basic, including the self-calibration examples in CVI and MSVC

Self-Calibration Procedure

The NI 5112 includes an internal voltage source that is over 10 times the accuracy of an 8-bit digitizer resolution. Self-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 specified tolerances.
- Calibrate analog trigger levels.
- Calibrate the time-to-digital converter (TDC) used for random interleaved sampling (RIS) measurements.

There is no method for adjusting the internal reference source, but verifying the value of the source using a high-precision DMM provides traceability for the verification procedure. Absolute accuracy is ensured by verifying the internal reference voltage using a digital voltmeter. Therefore, the verification procedure for the internal reference includes calls to `niScope_CalStart` and `niScope_CalEnd`.

Self-calibrate your digitizer before you externally calibrate. NI-SCOPE for LabVIEW, CVI, and MSVC includes self-calibration example programs. Table 2 shows the filenames of these programs and their locations.

Self-Calibrating Your NI 5112 Digitizer

To self-calibrate your NI 5112 digitizer, 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.

External Calibration Procedures

The external calibration procedure consists of verifying the performance of your digitizer, adjusting the calibration constants, and verifying again after the adjustments. The *Equipment and Other Test Requirements* section of this document gives required accuracies of input stimuli for specific devices.

Verifying NI 5112 Specifications

All procedures start by calling `niScope_init` with **resetDevice** set to `NISCOPE_VAL_TRUE`, and end with `niScope_close`. These procedures describe the program necessary for verifying digitizer specifications and are not intended as interactive steps. Required accuracies of input stimuli for specific devices are given in Table 1.

Perform an internal, or self-calibration, before verifying your NI 5112. Notice that the internal reference verification procedure includes function calls to record the measured internal reference value in the EEPROM. This procedure automatically stores the date and external calibration count to allow traceability. An external verification is equivalent to the factory production tests that verify the operation of the NI 5112. If any of these tests fail immediately after you perform an external calibration, return your digitizer to NI for repair.

Verifying Internal Reference

Complete the following steps to verify each internal reference entry in Table 3:

1. Connect the DMM to the digitizer input referenced by **whichReference** in Table 3.
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`.
     This constant encodes the channel name where the reference will be routed, such as `"0"` for channel 0.
   - **options**—`NISCOPE_VAL_CAL_POSITIVE`
4. Measure the internal reference voltage with the voltmeter and record this value (called *y* in calculations).
5. Route the internal reference out of the digitizer by calling `niScope_CalRouteInternalReference` with the following parameters:
   - **whichReference**—`NISCOPE_VAL_CAL_10V_CH0`
   - **options**—`NISCOPE_VAL_CAL_NEGATIVE`
6. Measure the internal reference voltage with the voltmeter and record this value (called *z* in calculations).
7. Calculate the value of the onboard reference with the formula $x = y - z$.
8. Compare this value (x) to the Success Condition value in Table 3.
   If the source is outside of specification, return the digitizer to NI for repair.
9. If the digitizer passes the test, record the measured value by calling `niScope_CalStoreInternalReference` with the following parameters:
   - **whichReference**—`NISCOPE_VAL_CAL_10V_CH0`
   - **internalReference**—The onboard reference value you computed in step 7

   This function stores the value in the driver, then it is written to the EEPROM during the next step.
10. Call `niScope_CalRouteInternalReference` with the following parameters:
    - **whichReference**—`NISCOPE_VAL_CAL_10V_CH0`
    - **option**—`NISCOPE_VAL_CAL_UNROUTE_SIGNAL`

11. Call `niScope_CalEnd` with **action** set to `NISCOPE_VAL_CAL_ACTION_STORE`.

    Closing the calibration session writes the internal reference value to the EEPROM. It also stores the date and the incremented external calibration count. While this value is not used by NI-SCOPE during operation, you can store the value in the EEPROM to provide traceability of the verification procedure.

You have now finished verifying the internal reference specifications.

Verifying Vertical Offset

Complete the following steps to verify each vertical offset entry in Table 3 and each channel:

1. Short-circuit the input of the digitizer.
2. Call `niScope_ConfigureAcquisition` with **acquisitionType** set to `NISCOPE_VAL_NORMAL`.
3. Call `niScope_ConfigureHorizontalRate` with the following parameters:
    - **minSampleRate**—`100,000,000` S/s
    - **minRecordLength**—`30,000`
    - **referencePosition**—`50.0`
4. Call `niScope_ConfigureVertical` with the following parameters:
    - **range**—The digitizer parameter value in Table 3
    - **offset**—`0.0`
    - **coupling**—`NISCOPE_VAL_DC`
    - **probeAttenuation**—`1.0`
    - **enabled**—`NISCOPE_VAL_TRUE`
5. Wait 10 ms for the input stage to settle.
6. Call `niScope_ReadWaveformMeasurement` with **measFunction** set to `NISCOPE_VAL_VOLTAGE_AVERAGE`.

    Compare the resulting average voltage to the success condition listed in Table 3. If the result is outside the success condition range, this portion of the verification has failed. Return your digitizer to NI for repair.
7. Repeat steps 1 through 5 for each vertical offset entry in Table 3.

You have now finished verifying the vertical offset specifications.

Verifying Vertical Gain

Complete the following to verify each vertical gain entry in Table 3 for each channel:

1. Connect the source to the input of the digitizer channel.
2. Call `niScope_ConfigureAcquisition` with **acquisitionType** set to `NISCOPE_VAL_NORMAL`.
3. Call `niScope_ConfigureHorizontalRate` with the following parameters:
   - **minSampleRate**—`100,000,000` S/s
   - **minRecordLength**—`30,000`
   - **referencePosition**—`50.0`
4. Call `niScope_ConfigureVertical` with the following parameters:
   - **range**—The digitizer parameter value in Table 3
   - **offset**—`0.0`
   - **coupling**—`NISCOPE_VAL_DC`
   - **probeAttenuation**—`1.0`
   - **enabled**—`NISCOPE_VAL_TRUE`
5. Wait 10 ms for the input stage to settle.
6. Call `niScope_ConfigureChanCharacteristics` with the following parameters:
   - **inputImpedance**—`NISCOPE_VAL_1_MEG_OHM`
   - **bandwidth**—`NISCOPE_VAL_FULL_BANDWIDTH` (or zero in LabVIEW)
7. Apply the positive DC stimulus voltage listed under Stimulus Parameters in Table 3.
8. Call `niScope_ReadWaveformMeasurement` with **measFunction** set to `NISCOPE_VAL_VOLTAGE_AVERAGE`.
9. Apply the negative DC stimulus voltage listed under Stimulus Parameters in Table 3.
10. Call `niScope_ReadWaveformMeasurement` with **measFunction** set to `NISCOPE_VAL_VOLTAGE_AVERAGE`.
11. Compute the error in the vertical gain using the 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, and *d* is the applied negative voltage.

12. Compare the value to the Success Condition in Table 3. If the error is outside the range of the success condition, return your digitizer to NI for repair.

You have now finished verifying the vertical gain specifications.

Table 3. NI 5112 Internal Reference, Vertical Offset, and Vertical Gain Specifications

Name	Digitizer Parameters	Stimulus Parameters	Success Condition
Internal Reference	**which Reference=** `NISCOPE_VAL_CAL_10V_CH0`	—	$9.99 < x < 10.01$ V
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
Vertical Gain	**range** = 50 V	±22.5 VDC	$\lvert x\rvert < 1.25$ V
Vertical Gain	**range** = 50 V	±5.0 VDC	$\lvert x\rvert < 1.25$ V
Vertical Gain	**range** = 5 V	±2.25 VDC	$\lvert x\rvert < 0.125$ V
Vertical Gain	**range** = 5 V	±0.25 VDC	$\lvert x\rvert < 0.125$ V
Vertical Gain	**range** = 0.5 V	±0.22 VDC	$\lvert x\rvert < 0.0125$ V
Vertical Gain	**range** = 0.5 V	±0.022 VDC	$\lvert x\rvert < 0.0125$ V

Verifying Full Bandwidth

Complete the following steps to verify each reference bandwidth entry in Table 4 for each channel and coupling mode:

1. Connect the signal generator to the digitizer input with a BNC cable.
2. Set the signal generator to the frequency and amplitude stimuli listed under stimulus parameters in Table 4 for the reference bandwidth entry.
3. Call `niScope_ConfigureVertical` with the following parameters:
    - **range**—`2.0`
    - **offset**—`0`
    - **coupling**—`NISCOPE_VAL_DC` or `NISCOPE_VAL_AC` (both should be tested)
    - **probeAttenuation**—`1.0`
    - **enabled**—`NISCOPE_VAL_TRUE`
4. Wait 300 ms for input stage to settle.

5. Call `niScope_ConfigureChanCharacteristics` with the following parameters:
   - **inputImpedance**—`NISCOPE_VAL_50_OHM`
   - **bandwidth**—Reference bandwidth value in Table 4
6. Call `niScope_ConfigureHorizontalRate` with the following parameters:
   - **minSampleRate**—The digitizer parameter value in Table 4
   - **minRecordLength**—`30,000`
   - **referencePosition**—`50.0`
7. Call `niScope_ReadWaveformMeasurement` with **measFunction** set to `NISCOPE_VAL_AC_ESTIMATE`.

   Record this value to use as the reference AC estimate in step 11.
8. Apply the signal specified in the bandwidth entry in Table 4.
9. Call `niScope_ConfigureHorizontalRate` with the following parameters:
   - **minSampleRate**—The value from Table 4
   - **minRecordLength**—`30,000`
   - **referencePosition**—`50.0`
10. Call `niScope_ReadWaveformMeasurement` with **measFunction** set to `NISCOPE_VAL_AC_ESTIMATE`.

    Record this value to use as the AC estimate in step 11.
11. Compute the response in decibels using the formula:

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

12. Compare the response to the success condition in Table 4. If the response is outside the range of the success condition, return your digitizer to NI for repair.
13. Repeat steps 7 through 11 for all the remaining bandwidth entries in Table 4 that correspond to the reference bandwidth you are using for this iteration of this procedure.

You have now finished verifying the full bandwidth specifications.

Table 4. NI 5112 Full Bandwidth Specifications

Name	Digitizer Parameters	Stimulus Parameters	Success Condition
Reference Full Bandwidth	**Bandwidth** = 0.0 = `NISCOPE_VAL_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	$\lvert x \rvert < 3$ dB
Full Bandwidth	**minSampleRate** = 50,000,000 S/s	49 MHz, 1.5 Vpp (measurement is intentionally aliased)	$\lvert x \rvert < 3$ dB
Full Bandwidth	**minSampleRate** = 100,000,000 S/s	99 MHz, 1.5 Vpp (measurement is intentionally aliased)	$\lvert x \rvert < 3$ dB

Verifying 20 MHz Bandwidth

Complete the following steps to verify each reference bandwidth entry in Table 5 for each channel and coupling mode:

1. Connect the signal generator to the digitizer input with a BNC cable.
2. Set the signal generator to the frequency and amplitude listed in Table 5 for the reference bandwidth entry.
3. Call `niScope_ConfigureVertical` with the following parameters:
   - **range**—`2.0`
   - **offset**—`0`
   - **coupling**—`NISCOPE_VAL_DC` or `NISCOPE_VAL_AC` (both should be tested)
   - **probeAttenuation**—`1.0`
   - **enabled**—`NISCOPE_VAL_TRUE`
4. Wait 300 ms for input stage to settle.
5. Call `niScope_ConfigureChanCharacteristics` with the following parameters:
   - **inputImpedance**—`NISCOPE_VAL_50_OHM`
   - **bandwidth**—The bandwidth value in Table 5

6. Call `niScope_ConfigureHorizontalRate` with the following parameters:
   - **minSampleRate**—The digitizer parameter value in Table 5
   - **minRecordLength**—`30,000`
   - **referencePosition**—`50.0`
7. Call `niScope_ReadWaveformMeasurement` with **measFunction** set to `NISCOPE_VAL_AC_ESTIMATE`.

   Record this value to use as the reference AC estimate in step 11.
8. Apply the signal specified in the digitizer parameter bandwidth entry in Table 5.
9. Call `niScope_ConfigureHorizontalRate` with the following parameters:
   - **minSampleRate**—The digitizer parameter value in Table 5
   - **minRecordLength**—`30,000`
   - **referencePosition**—`50.0`
10. Call `niScope_ReadWaveformMeasurement` with **measFunction** set to `NISCOPE_VAL_AC_ESTIMATE`.

    Record this value to use as the AC estimate in step 11.
11. Compute the response in decibels using the formula:

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

12. Compare the response to the success condition in Table 5. If the response is outside the range of the success condition, return your digitizer to NI for repair.
13. Repeat steps 7 through 11 for all the remaining bandwidth entries in Table 5 that correspond to the reference bandwidth you are using for this iteration of this procedure.

You have now finished verifying the 20 MHz bandwidth specifications.

Table 5. NI 5112 20 MHz Bandwidth Specifications

Name	Digitizer Parameters	Stimulus Parameters	Success Condition
Reference 20 MHz Bandwidth	**bandwidth = 20,000,000 =** `NISCOPE_VAL_20MHZ_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	$\lvert x \rvert < 3$ dB
20 MHz Bandwidth	**minSampleRate** = 20,000,000 S/s	19.9 MHz, 1.5 Vpp (measurement is intentionally aliased)	$\lvert x \rvert < 3$ dB
20 MHz Bandwidth	**minSampleRate** = 20,000,000 S/s	21 MHz, 1.5 Vpp (measurement is intentionally aliased)	$\lvert x \rvert > 3$ dB

Verifying Input Impedance

Complete the following steps to verify each input impedance entry in Table 6 for each channel:

1. Connect the digitizer input to the ohmmeter with a coaxial cable.
2. Call `niScope_ConfigureVertical` with the following parameters:
   - **range**—The digitizer parameter value in Table 6
   - **offset**—`0.0`
   - **coupling**—`NISCOPE_VAL_DC`
   - **probeAttenuation**—`1.0`
   - **enabled**—`NISCOPE_VAL_TRUE`
3. Wait 10 ms for the input stage to settle.
4. Call `niScope_ConfigureChanCharacteristics` with the following parameters:
   - **inputImpedance**—The digitizer parameter value in Table 6
   - **bandwidth**—`NISCOPE_VAL_FULL_BANDWIDTH` (or zero in LabVIEW)
5. Call `niScope_ReadWaveform` to ensure the hardware is programmed.
6. Measure the impedance (*x*) on the ohmmeter and compare it to the impedance value in Table 6.

   If the impedance is outside the range of the success condition, return your digitizer to NI for repair.

You have now finished verifying the input impedance specifications.

Table 6. NI 5112 Input Impedance Specifications

Name	Digitizer Parameters	Stimulus Parameters	Success Condition

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
	inputImpedance = `NISCOPE_VAL_1_MEG_OHM`
		—	990,000 < *x* < 1,010,000 Ω

Input Impedance	range = 4.0
	inputImpedance = `NISCOPE_VAL_50_OHM`
		—	49.5 < *x* < 50.5 Ω

Input Impedance	range = 0.4
	inputImpedance = `NISCOPE_VAL_50_OHM`
		—	49.5 < *x* < 50.5 Ω

Verifying AC Coupling Cutoff Frequency

Complete the following to verify each channel and AC coupling entry in Table 7:

1. Connect a BNC cable from the signal generator to the high-impedance digitizer input.
2. Set signal generator to the frequency and amplitude listed in Table 7.
3. Call `niScope_ConfigureVertical` with the following parameters:
   - **range**—`2.0`
   - **offset**—`0.0`
   - **coupling**—`NISCOPE_VAL_DC`
   - **probeAttenuation**—`1.0`
   - **enabled**—`NISCOPE_VAL_TRUE`
4. Wait 10 ms for the input stage to settle.

5. Call `niScope_ConfigureChanCharacteristics` with the following parameters:
   - **inputImpedance**—`NISCOPE_VAL_1_MEG_OHM`
   - **bandwidth**—`NISCOPE_VAL_FULL_BANDWIDTH` (or `0.0` in LabVIEW)
6. Call `niScope_ConfigureHorizontalRate` with the following parameters:
   - **minSampleRate**—`10,000` S/s
   - **minRecordLength**—`10,000`
   - **referencePosition**—`50.0`
7. Call `niScope_ReadWaveformMeasurement` with **measFunction** set to `NISCOPE_VAL_AC_ESTIMATE`.

   Record this value to use as the AC estimate with DC coupling in step 11.
8. Call `niScope_ConfigureVertical` with the following parameters:
   - **range**—`2.0`
   - **offset**—`0.0`
   - **coupling**—`NISCOPE_VAL_AC`
   - **probeAttenuation**—`1.0`
   - **enabled**—`NISCOPE_VAL_TRUE`
9. Wait 300 ms for the input stage to settle.
10. Call `niScope_ReadWaveformMeasurement` with **measFunction** set to `NISCOPE_VAL_AC_ESTIMATE`.

    Record this value to use as the AC estimate with AC coupling in step 11.
11. Compute the response in decibels using the formula:

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

12. Compare the response to the success condition in Table 7. If the response is outside the listed range, return your digitizer to NI for repair.

You have now finished verifying the AC coupling cutoff frequency specifications.

Verifying Timing

Complete the following steps to verify each timing entry in Table 7:

1. Connect the BNC cable from the signal generator to the high-impedance digitizer input.
2. Set the signal generator to 10 kHz, 1.8 Vpp sine wave.
3. Call `niScope_ConfigureVertical` with the following parameters:
   - **range**—`2.0`
   - **offset**—`0.0`
   - **coupling**—`NISCOPE_VAL_DC`
   - **probeAttenuation**—`1.0`
   - **enabled**—`NISCOPE_VAL_TRUE`
4. Wait 10 ms for the input stage to settle.
5. Call `niScope_ConfigureChanCharacteristics` with the following parameters:
   - **inputImpedance**—`NISCOPE_VAL_1_MEG_OHM`
   - **bandwidth**—`NISCOPE_VAL_FULL_BANDWIDTH` (or `0.0` in LabVIEW)
6. Call `niScope_ConfigureHorizontalRate` with the following parameters:
   - **minSampleRate**—`1,000,000`
   - **minRecordLength**—`100,000`
7. Call `niScope_ReadWaveformMeasurement` with **measFunction** set to `NISCOPE_VAL_AVERAGE_FREQUENCY`.
8. The returned frequency value must be between 9999 and 10,001 Hz, or a hardware error exists. If this step fails, terminate the verification procedure and return your digitizer to NI for repair.
9. Set the signal generator to 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.
10. Call `niScope_ReadWaveformMeasurement` with **measFunction** set to `NISCOPE_VAL_AVERAGE_PERIOD`.

    Record the period measurement to use in step 11.
11. 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 not enough crosspoints, the device passed the test.

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

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

which is:

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

13. Compare the actual sample rate (x) to the success condition in Table 7. If x is outside the range of the success condition, return your digitizer to NI for repair.

You have now finished verifying the timing specifications.

Verifying Trigger Sensitivity

Complete the following steps for all analog trigger channels to test the smallest signal on which the digitizer should trigger by trying all possible trigger levels:

1. Connect the BNC cable from the signal generator to the high-impedance digitizer input.
2. Apply a 1 MHz sine wave with zero vertical offset, and peak-to-peak voltage as listed in Table 7.
3. Call `niScope_ConfigureVertical` with the following parameters:
   - **range**—`2.0`
   - **offset**—`0.0`
   - **coupling**—`NISCOPE_VAL_DC`
   - **probeAttenuation**—`1.0`
   - **enabled**—`NISCOPE_VAL_TRUE`
4. Wait 10 ms for the input stage to settle.
5. Call `niScope_ConfigureChanCharacteristics` with the following parameters:
   - **inputImpedance**—`NISCOPE_VAL_1_MEG_OHM`
   - **bandwidth**—`NISCOPE_VAL_FULL_BANDWIDTH` (or `0.0` in LabVIEW)

6. Call `niScope_ConfigureHorizontalRate` with the following parameters:
   - **minSampleRate**—`20,000,000`
   - **minRecordLength**—`128`
7. Call `niScope_ConfigureTriggerSource` with the following parameters:
   - **triggerSource**—The channel you are verifying, such as 0 or `NISCOPE_VAL_EXTERNAL`
   - **triggerType**—`NISCOPE_VAL_EDGE`
   - **triggerDelay**—`0`
   - **holdoff**—`0`
8. Call `niScope_ConfigureEdgeTrigger` with the following parameters:
   - **level**—The trigger level as discussed in step 10
   - **triggerCoupling**—`NISCOPE_VAL_AC`
   - **slope**—`NISCOPE_VAL_POSITIVE`
9. Call `niScope_ReadWaveform` to read a waveform with **maxTime** set to 1,000 ms or greater and **waveform size** set to 128. If this function returns a "maximum time exceeded" error, the digitizer did not trigger. If the `niScope_ReadWaveform` succeeds, the digitizer passed this test.
10. Call `niScope_Abort` to stop the test.
11. Repeat steps 7 through 10, starting with the low-trigger level listed in Table 7. Increment the **level** by the trigger level delta value in Table 7 until **level** is greater than the high trigger level listed in the table, or until the digitizer triggers. If all trigger levels have been tested, a hardware problem exists with the trigger sensitivity. Return your digitizer to NI for repair.

You have now finished verifying the trigger sensitivity specifications.

Verifying Random Interleaved Sampling Timing

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 period signal. This distribution is called Random Interleaved Sampling (RIS). Complete the following steps to measure RIS:

1. Connect the signal generator to 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. Apply the signal in Table 7.
3. Call `niScope_ConfigureVertical` with the following parameters:
   - **range**—`2`
   - **offset**—`0`
   - **coupling**—`NISCOPE_VAL_DC`
   - **probeAttenuation**—`1.0`
   - **enabled**—`NISCOPE_VAL_TRUE`
4. Call `niScope_ConfigureHorizontalRate` with the following parameters:
   - **minSampleRate**—`100,000,000`
   - **minRecordLength**—`128`
   - **referencePosition**—`50.0`
5. Call `niScope_ConfigureTriggerSource` with the following parameters:
   - **triggerSource**—The channel you are verifying
   - **triggerType**—`NISCOPE_VAL_EDGE`
   - **triggerDelay**—`0`
   - **holdoff**—`0`
6. Call `niScope_ConfigureEdgeTrigger` with the following parameters:
   - **level**—`0`
   - **triggerCoupling**—`NISCOPE_VAL_DC`
   - **slope**—`NISCOPE_VAL_POSITIVE`
7. Call `niScope_CalMeasureRISDistribution` with the following parameters:
   - **distributionSize**—The digitizer parameter value in Table 7
   - **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`.

   `niScope_CalMeasureRISDistribution` acquires 2000 data points 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 7. If the returned value is outside the range of the success condition, return your digitizer to NI for repair.

You have now finished verifying the random interleaved sampling timing specifications.

Table 7. NI 5112 AC Coupling, Timing, Trigger Sensitivity, and RIS Distribution Specifications

Name	Digitizer Parameters	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
Timing	—	—	$999{,}950 < x < 1{,}000{,}050$ Hz
Trigger Sensitivity	**low trigger level** = –0.75 **high trigger level** = 0.75 **trigger level delta** = 0.01	300 mVpp with CH0 and CH1; 750 mVpp with external trigger	scope triggers with any valid trigger level
RIS Distribution	**distributionSize** = 25	1 MHz, ±100 kHz, 1.8 Vpp	$x > 0.8$

Calibration Function Reference

This section lists functions specific to NI-SCOPE calibration. See the *Software* section of this document for instructions on installing these functions.

niScope_CalStart

Format

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

Purpose

`niScope_CalStart` opens an external calibration session.

Using This Function

password is compared to the password stored in the EEPROM for additional security. By default, the password is set to `NULL`, or an 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 `NULL`s after the first `NULL` is found. This padding allows strings of less than four characters to be legal passwords.

All calibration functions require a session handle, such as **newSessionHandle**, that is returned by this function. Only the external calibration functions require a calibration session handle for password protection. All other functions, such as verification and fetch 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 may 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

Table 8. niScope_CalStart Function

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

niScope_CalEnd

Format

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

Purpose

`niScope_CalEnd` closes an external calibration session.

Using This Function

If **action** is `NISCOPE_VAL_CAL_ACTION_ABORT`, the session is closed, 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 zero or the empty string in the EEPROM. The current system date and an incremented external calibration count are automatically stored in the EEPROM.

Parameters

Table 9. `niScope_CalEnd` Function

Name	Description
sessionHandle	session handle returned by `niScope_CalStart`
action	values defined in `niScopeCal.h`: `NISCOPE_VAL_CAL_ACTION_STORE`, `NISCOPE_VAL_CAL_ACTION_ABORT`

niScope_CalChangePassword

Format

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

Purpose

`niScope_CalChangePassword` checks your old password with the one stored in the EEPROM. If they match, the new password is stored in the EEPROM.

Using This Function

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

By default, the password in the EEPROM is an array of `NULL`s, or the empty string.

If you forget your password, call NI.

Parameters

Table 10. `niScope_CalChangePassword` Function

Name	Description
sessionHandle	session handle returned by `niScope_CalStart` or `niScope_init`
oldPassword	value currently stored in EEPROM (factory default is “”)
newPassword	new value to store in EEPROM

niScope_CalFetchCount

Format

```
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 it is the self or external calibration count. Possible values are defined in `niScopeCal.h`.

Parameters

Table 11. `niScope_CalFetchCount` Function

Name	Description
sessionHandle	session handle returned by `niScope_CalStart` or `niScope_init`
whichOne	values defined in niScopeCal.h: `NISCOPE_VAL_CAL_SELF`, `NISCOPE_VAL_CAL_EXTERNAL`
calibrationCount	number of times device has been calibrated

niScope_CalFetchDate

Format

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

Purpose

`niScope_CalFetchDate` returns the self-calibration, external calibration, or manufacture date.

Using This Function

If you are upgrading to NI-SCOPE 1.6 from version 1.5 or earlier, the initial calibration dates will be incorrect because the older versions of NI-SCOPE calibration do not support the date feature.

Parameters

Table 12. `niScope_CalFetchDate` Function

Name	Description
sessionHandle	session handle returned by `niScope_CalStart` or `niScope_init`
whichOne	values defined in `niScopeCal.h`: `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

Format

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

Purpose

`niScope_CalFetchMiscInfo` returns the miscellaneous information you may store during an external calibration using `niScope_StoreMiscInfo`.

Using This Function

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

Parameters

Table 13. `niScope_CalFetchMiscInfo` Function

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 calibration

niScope_CalFetchInternalReference

Format

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

Purpose

`niScope_CalFetchInternalReference` returns the value of the internal reference that is stored with the `niScope_CalStoreInternalReference` function during external calibration. This ability allows tracking of the internal reference value, though 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

Table 14. `niScope_CalFetchInternalReference` Function

Name	Description
sessionHandle	session handle returned by `niScope_CalStart` or `niScope_init`
whichReference	value defined in `niScopeCal.h`: `NISCOPE_VAL_CAL_10V_CH0`
internalReference	returned value of internal reference that the user stored during external calibration

niScope_CalStoreMiscInfo

Format

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

Purpose

`niScope_CalStoreMiscInfo` allows the user to store miscellaneous information in the EEPROM during external calibration. For example, you can store an operator ID for the person or company performing the calibration.

Using This Function

If this function is not called during an external calibration, 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

Table 15. `niScope_CalStoreMiscInfo` Function

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

niScope_CalStoreInternalReference

Format

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

Purpose

`niScope_CalStoreInternalReference` allows you to store the measured internal reference voltage value.

Using This Function

The NI 5112 supports routing the internal reference to the front BNC connectors, using `niScope_CalRouteInternalReference`. You can measure with a DMM to verify that it is within specification. `niScope_CalStoreInternalReference` can store the measured value in the EEPROM to allow the user to track the drift of the internal reference over time. The value is stored for tracking purposes only and is not used during digitizer operation. The reference value is stored in the EEPROM as a 32-bit floating point number.

`niScope_CalStoreInternalReference` requires an external calibration 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 `niScope_CalStoreInternalReference` is not called during an external calibration session, the internal reference is set to zero in the EEPROM when `niScope_CalEnd` is called with **action** set to `NISCOPE_VAL_CAL_ACTION_STORE`. This ensures consistent calibration count, date, and internal reference values in the EEPROM.

Parameters

Table 16. `niScope_CalStoreInternalReference` Function

Name	Description
sessionHandle	session handle returned by `niScope_CalStart`
whichReference	values defined in `niScopeCal.h`: `NISCOPE_VAL_CAL_10V_CH0`
internalRefValue	reference value to store

niScope_CalSelfCalibrate

Format

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

Purpose

`niScope_CalSelfCalibrate` performs a self-calibration.

Using This Function

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. The outdated `niScope_Calibrate` with **calibrationOperation** set to `NISCOPE_VAL_SELF_CALIBRATION` calls this function. The only valid value for **option** is `NISCOPE_VAL_CAL_RESTORE_EXTERNAL_CALIBRATION`. This is equivalent to calling the outdated `niScope_Calibrate` with **calibrationOperation** set to `NISCOPE_VAL_RESTORE_FACTORY_CALIBRATION`. You should use this operation only if the self-calibration routine fails, and you must use the digitizer rather than return it for repair. This operation restores the previous external calibration constants. Notice that using the external calibration constants does not correct for environmental conditions, so the digitizer will not be in an optimal calibration state.

Parameters

Table 17. `niScope_CalSelfCalibrate` Function

Name	Description
sessionHandle	session handle returned by `niScope_CalStart` or `niScope_init`
channelName	this parameter is currently ignored; use `VI_NULL`
option	only `NISCOPE_VAL_CAL_RESTORE_EXTERNAL_CALIBRATION` is supported; use `VI_NULL` for a normal self-calibration operation

niScope_CalRouteInternalReference

Format

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

Purpose

`niScope_CalRouteInternalReference` routes the internal reference to the front BNC connector of the specified channel for use in measuring the internal reference and storing the value in the EEPROM with `niScope_CalStoreInternalReference`. This function allows you to track the verification procedure and the drift of the source over time.

Using This Function

See the *Verifying Internal Reference* subsection in the *Verifying NI 5112 Specifications* section for details on using `niScope_CalRouteInternalReference`. Also see the *Verifying NI 5112 Specifications* section for details on verifying the operation of your NI 5112.

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

Table 18. `niScope_CalRouteInternalReference` Function

Name	Description
sessionHandle	session handle returned by `niScope_CalStart`
option	values defined in `niScopeCal.h`: `NISCOPE_VAL_CAL_UNROUTE_SIGNAL`, `NISCOPE_VAL_CAL_POSITIVE`, `NISCOPE_VAL_CAL_NEGATIVE`
whichReference	values defined in `niScopeCal.h`: `NISCOPE_VAL_CAL_10V_CH0`

niScope_CalMeasureRISDistribution

Format

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

Purpose

`niScope_CalMeasureRISDistribution` calls `niScope_ReadWaveform` 2000 times to take an acquisition from the specified channel and retrieve the initial *x* value, which includes the TDC.

Using This Function

The TDC should be a uniform distribution between two sample points since triggers should occur randomly. To test this uniformity, the distribution of initial *x* values is created. The percentage of triggers in the smallest bin of this distribution is returned for comparison to a specification to determine if RIS is operating correctly.

The distribution parameter must be declared as an array of **distributionSize** in order to return the distribution. Optionally, setting **distribution** to `VI_NULL` specifies that the distribution should not be returned.

For step-by-step instructions on verifying the RIS distribution, see the *Verifying NI 5112 Specifications* section of this document.

Parameters

Table 19. `niScope_CalMeasureRISDistributions` Function

Name	Description
sessionHandle	session handle returned by `niScope_CalStart` or `niScope_init`
channelName	string name of channel for which to store the internal reference value, e.g. "`0`" or "`1`"
maxTime	the **maxTime** parameter to `niScope_ReadWaveform`, this is the 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 *Verifying NI 5112 Specifications* 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