# NI 5102

## Introduction

This document contains information and step-by-step instructions for calibrating the NI 5102 digitizer. This calibration procedure is intended for metrology labs. It includes specific programming instructions necessary for external calibration of the NI 5102 using Measurement Studio, LabVIEW, C, or Visual Basic programming environments.

### What Is Calibration?

*Calibration* consists of verifying the measurement accuracy of a device and adjusting for any measurement error. *Verification* is measuring the performance of a device and comparing the results to the factory specifications for the device. 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. The NI 5102 has two types of calibration: external calibration and self-calibration.

### External Calibration

External calibration requires using a high-precision digitizer calibrator to verify and adjust calibration constants. This procedure replaces all calibration constants in the EEPROM and is equivalent to a factory calibration. 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.

### Self-Calibration

Self-calibration, or internal calibration, uses a software command and requires no external connections. Self-calibration adjusts a device for use in an environment where external variables, such as temperature, may



  *December 2001*

differ from those in the environment in which the device was externally calibrated.

## 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 National Instruments (NI) standards.

## How Often Should You Calibrate?

The accuracy requirements of your measurement application determine how often you should calibrate the NI 5102 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 the digitizer needs adjustment.

# Equipment and Other Test Requirements

This section describes the equipment, documentation, software, and test conditions required for calibrating the NI 5102.

## Test Equipment

Table 1 lists the equipment required for calibrating the NI 5102. If you do not have the recommended instruments, use these specifications to select a substitute calibration standard.

**Table 1.** Required Equipment for NI 5102 Calibration

| Required Equipment | Recommended Equipment | Parameter Measured | Necessary Specifications |
|---|---|---|---|
| Signal Generator/ Ohmmeter | Fluke 9500B Oscilloscope Calibrator | Vertical Gain | DC ±40 mV to ±4.5 V, ±0.1% into 1 MΩ |
| | | AC Coupling | Sine wave 9–13 Hz ±100 ppm, 1.8 Vpp ±2% into 1 MΩ |
| | | Bandwidth | 1.5% amplitude flatness for leveled sine wave 100 kHz–20 MHz ±50 ppm, 1.5 Vpp ± 2% into 1 MΩ |
| | | Input Impedance | 2-wire resistance accuracy of 0.25% for 1 MΩ measurement |

**Table 1.** Required Equipment for NI 5102 Calibration (Continued)

| Required Equipment | Recommended Equipment | Parameter Measured | Necessary Specifications |
|---|---|---|---|
| BNC Cable | — | — | 50 Ω |
| BNC Shorting Cap | — | Vertical Offset | 0 VDC ±0.1 mV |
| BNC T Connector | — | — | — |

# Test Conditions

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

- Keep connections to the NI 5102 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% non-condensing, or consult the digitizer hardware documentation for the optimum relative humidity.
- Maintain the temperature between 5 and 40 °C, or consult the digitizer hardware documentation for the optimum temperature range.
- Allow a warm-up time of at least 15 minutes to ensure that the measurement circuitry of the NI 5102 is at a stable operating temperature.

# Documentation

This section describes the documentation you need to calibrate the NI 5102. In addition to this calibration document, you may find it helpful to refer to the following documents:

- *NI 5102 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 5102 digitizer. 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. Refer to 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 calibration 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, which 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 |

**Table 2.** Calibration File Location after Installing NI-SCOPE 2.0 or Later (Continued)

| File Name and Location | Description |
|---|---|
| `VXIpnp\winnt(Win95)\Niscope\Niscope.fp` | CVI function panel file that includes function prototypes and help on using NI-SCOPE in the CVI environment |
| `VXIpnp\winnt(Win95)\Niscope\niScopeCal.fp` | CVI function panel file that includes external calibration function prototypes and help on using NI-SCOPE in the CVI environment |
| `VXIpnp\winnt(Win95)\Niscope\Examples\` | Directory of NI-SCOPE examples for CVI, C, Visual C++, and Visual Basic |

# Self-Calibration Procedure

Self-calibrate the digitizer before you begin external calibration. NI-SCOPE includes self-calibration example programs for LabVIEW, Measurement Studio, Visual Basic, and Console C. Table 2 shows the locations of these example programs.

**Note** Self-calibration of the NI 5102 adjusts only the timing for random interleaved sampling (RIS) acquisitions.

## Self-Calibrating the NI 5102

To self-calibrate the 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 and deallocate system resources.

# External Calibration Procedures

External calibration consists of three steps: verifying the performance of the digitizer, adjusting calibration constants, and reverifying after adjustment. All procedures start by calling `niScope_init` with **resetDevice** set to `VI_TRUE` and end by calling `niScope_close`. Table 1 lists the accuracies of input stimuli for specific devices.

## Verifying the Performance of the NI 5102

To determine if the NI 5102 requires adjustment, verify each vertical offset specification listed in Table 3 and each vertical gain specification listed in Table 4. If any of these tests fail, externally adjust the digitizer. If any tests fail immediately after an external adjustment, return the digitizer to NI for repair.

### Verifying Vertical Offset

Complete the following steps to verify NI 5102 vertical offset specifications:

1. Short-circuit channel 0 of the digitizer with the BNC shorting cap.
2. Call `niScope_ConfigureAcquisition` with **acquisitionType** set to `NISCOPE_VAL_NORMAL`.
3. Call `niScope_ConfigureHorizontalTiming` with the following parameters:
   - **minSampleRate** = `20,000,000`
   - **minNumPts** = `30,000`
   - **refPosition** = `50.0`
   - **numRecords** = `1`
   - **enforceRealTime** = `VI_TRUE`
4. Call `niScope_ConfigureVertical` with the following parameters:
   - **channelList** = `0`
   - **range** = The first vertical offset entry in the Digitizer Parameters column in Table 3
   - **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_InitiateAcquisition`.
7. Call `niScope_FetchMeasurement` with the following parameters:
   - **channelList** = `0`
   - **scalarMeasFunction** = `NISCOPE_VAL_VOLTAGE_AVERAGE`
   - **timeout** = `0`
8. Compare the resulting average voltage to the first value in the Success Condition column in Table 3. If it falls outside this range, the NI 5102 has failed this portion of the verification. You must externally adjust

the digitizer to guarantee its accuracy. If the digitizer fails after an external adjustment, return it to NI for repair.

9. Repeat steps 2 through 8 for each vertical offset entry in Table 3.
10. Move the shorting cap to channel 1 of the digitizer.
11. Repeat steps 2 through 9 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 5102.

**Table 3.** NI 5102 Vertical Offset Specifications

| Name | Digitizer Parameters | Stimulus Parameters | Success Condition |
|---|---|---|---|
| Vertical Offset | **range** = 10 V | Short-Circuit Input | $|x| < 0.05800$ V |
| Vertical Offset | **range** = 2 V | Short-Circuit Input | $|x| < 0.01100$ V |
| Vertical Offset | **range** = 0.5 V | Short-Circuit Input | $|x| < 0.00290$ V |
| Vertical Offset | **range** = 0.1 V | Short-Circuit Input | $|x| < 0.00058$ V |

## Verifying Vertical Gain

Complete the following steps to verify NI 5102 vertical gain specifications:

1. Connect the signal generator to channel 0 of the digitizer.
2. Configure the signal generator for a a MΩ load.
3. Call `niScope_ConfigureAcquisition` with **acquisitionType** set to `NISCOPE_VAL_NORMAL`.
4. Call `niScope_ConfigureHorizontalTiming` with the following parameters:
   - **minSampleRate** = `20,000,000`
   - **minNumPts** = `30,000`
   - **refPosition** = `50.0`
   - **numRecords** = `1`
   - **enforceRealTime** = `VI_TRUE`
5. Call `niScope_ConfigureVertical` with the following parameters:
   - **channelList** = `0`
   - **range** = The first vertical offset entry in the Digitizer Parameters column in Table 4
   - **offset** = `0.0`
   - **coupling** = `NISCOPE_VAL_DC`

- **probeAttenuation** = `1.0`
- **enabled** = `VI_TRUE`

6. Wait 10 ms for the input stage to settle.
7. Apply the positive DC voltage listed in the Stimulus Parameters column for the first vertical gain entry in Table 4.
8. Call `niScope_InitiateAcquisition`.
9. Call `niScope_FetchMeasurement` with the following parameters:
   - **channelList** = `0`
   - **scalarMeasFunction** = `NISCOPE_VAL_VOLTAGE_AVERAGE`
   - **timeout** = `30`
10. Apply the negative DC voltage listed in the Stimulus Parameters column for the first vertical gain entry in Table 4.
11. Call `niScope_InitiateAcquisition`.
12. Call `niScope_FetchMeasurement` with the following parameters:
    - **channelList** = `0`
    - **scalarMeasFunction** = `NISCOPE_VAL_VOLTAGE_AVERAGE`
    - **timeout** = `30`
13. Calculate the error in the vertical gain using the formula:

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

    where

    $a$ = the measured positive voltage

    $b$ = the measured negative voltage

    $c$ = the applied positive voltage

    $d$ = the applied negative voltage
14. Compare the error to the first entry in the Success Condition column in Table 4. If the error is less than this value, the NI 5102 has passed this portion of the vertical gain test. Otherwise, externally adjust the digitizer. If the digitizer fails after external adjustment, return it to NI for repair.
15. Repeat steps 3 through 14 for each vertical gain entry in Table 4.
16. Move the signal generator connection to channel 1 of the digitizer.
17. Repeat steps 3 through 15 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 5102.

**Table 4.** NI 5102 Vertical Gain Specifications

| Name | Digitizer Parameters | Stimulus Parameters | Success Condition |
|---|---|---|---|
| Vertical Gain | **range** = 10 V | ±4.5 VDC | \|x\| < 0.1 V |
| Vertical Gain | **range** = 10 V | ±1.5 VDC | \|x\| < 0.1 V |
| Vertical Gain | **range** = 2 V | ±0.9 VDC | \|x\| < 0.02 V |
| Vertical Gain | **range** = 2 V | ±0.5 VDC | \|x\| < 0.02 V |
| Vertical Gain | **range** = 0.5 V | ±0.22 VDC | \|x\| < 0.005 V |
| Vertical Gain | **range** = 0.5 V | ±0.1 VDC | \|x\| < 0.005 V |
| Vertical Gain | **range** = 0.1 V | ±0.045 VDC | \|x\| < 0.001 V |
| Vertical Gain | **range** = 0.1 V | ±0.007 VDC | \|x\| < 0.001 V |

## Verifying Large Signal Bandwidth

To verify NI 5102 large signal bandwidth specifications, complete the following steps:

1. Connect the BNC T connector to channel 0 of the digitizer, and connect the signal generator and 50 Ω terminator to the two inputs on the T connector.
2. Configure the signal generator for a 50 Ω load.
3. Set the signal generator to the frequency and amplitude listed in Table 5 for the 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_ConfigureHorizontalTiming` with the following parameters:
   - **minNumPts** = `30,000`
   - **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 *reference AC estimate* in step 13.
9. Apply the signal listed in the Stimulus Parameters column for the second bandwidth entry in Table 5.
10. Call `niScope_ConfigureHorizontalTiming` with the following parameters:
    - **minNumPts** = `30,000`
    - **refPosition** = `50.0`
    - **numRecords** = `1`
    - **enforceRealTime** = `VI_TRUE`
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* in step 13.
13. Calculate the response in decibels using the formula:

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

14. Compare *response* to the second range listed in the Success Condition column in Table 5. If *response* is outside the given range, the NI 5102 has failed this test. Return the digitizer to NI for repair.
15. Repeat steps 9 through 14 for the last bandwidth entry in Table 5.
16. Move the signal generator connection to channel 1 of the digitizer.
17. Repeat steps 2 through 15 for channel 1. Change **channelList** to `1` for the functions `niScope_ConfigureVertical` and `niScope_FetchMeasurement`,

You have completed verifying the large signal bandwidth specifications for the NI 5102.

**Table 5.** NI 5102 Large Signal Bandwidth Specifications

| Name | Digitizer Parameters | Stimulus Parameters | Success Condition |
|---|---|---|---|
| Reference Bandwidth (Large Signal) | **minSampleRate** = 20,000,000 S/s | 100 kHz, 1.9 Vpp | — |
| Bandwidth (Large Signal) | **minSampleRate** = 1,000,000 S/s | 999,500 Hz, 1.9 Vpp (intentionally aliased) | $\|x\| < 3$ dB |
| Bandwidth (Large Signal) | **minSampleRate** = 10,000,000 S/s | 9.9 MHz, 1.9 Vpp (intentionally aliased) | $\|x\| < 3$ dB |

## Verifying Small Signal Bandwidth

To verify the NI 5102 small signal bandwidth specifications, complete the following steps:

1. Connect the BNC T connector to channel 0 of the digitizer, and connect the signal generator and 50 Ω terminator to the two inputs on the T connector.
2. Configure the signal generator for a 50 Ω load.
3. Set the signal generator to the frequency and amplitude listed in the Stimulus Parameters column for the Reference Bandwidth entry in Table 6.
4. Call `niScope_ConfigureVertical` with the following parameters:
   - **channelList** = `0`
   - **range** = `2 V`
   - **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_ConfigureHorizontalTiming` with the following parameters:
   - **minNumPts** = `30,000`
   - **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 *reference AC estimate* in step 13.
9. Apply the signal specified in the Stimulus Parameters column in Table 6.
10. Call niScope_ConfigureHorizontalTiming with the following parameters:
    - **minNumPts** = 30,000
    - **refPosition** = 50.0
    - **numRecords** = 1
    - **enforceRealTime** = VI_TRUE
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* in step 13.
13. Calculate the response in decibels using the formula:

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

14. Compare *response* to the range listed in the Success Condition column in Table 6. If *response* is outside the given range, the NI 5102 has failed this test. Return the digitizer to NI for repair.
15. Repeat steps 9 through 14 for the remaining bandwidth entries in Table 6.
16. Repeat steps 2 through 15 with **coupling** set to NISCOPE_VAL_AC for the function niScope_ConfigureVertical.
17. Move the signal generator connection to channel 1 of the digitizer.
18. Repeat steps 3 through 16 for channel 1. Change **channelList** to 1 for the functions niScope_ConfigureVertical and niScope_FetchMeasurement,

You have completed verifying the small signal bandwidth specifications for the NI 5102.

**Table 6.** NI 5102 Small Signal Bandwidth Specifications

| Name | Digitizer Parameters | Stimulus Parameters | Success Condition |
|---|---|---|---|
| Reference Bandwidth (Small Signal) | **minSampleRate** = 20,000,000 S/s **bandwidth** = `NISCOPE_VAL_FULL_BANDWIDTH` (`0.0` in LabVIEW) | 100 kHz, 0.2 Vpp | — |
| Bandwidth (Small Signal) | **minSampleRate** = 1,000,000 S/s | 999,500 Hz, 0.2 Vpp (intentionally aliased) | $\|x\| < 3$ dB |
| Bandwidth (Small Signal) | **minSampleRate** = 5,000,000 S/s | 14.9 MHz, 0.2 Vpp (intentionally aliased) | $\|x\| < 3$ dB |

## Verifying Input Impedance

To verify the NI 5102 input impedance specifications, complete the following steps:

1. Connect the ohmmeter to channel 0 of the digitizer.
2. Call `niScope_ConfigureVertical` with the following parameters:
   - **channelList** = `0`
   - **range** = The range values in the Digitizer Parameters column of Table 7
   - **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_Read` to ensure that the hardware is programmed. Set the following parameters:
   - **channelList** = `0`
   - **timeout** = `30`
   - **numSamples** = `128`
5. Measure the impedance ($x$) on the ohmmeter and compare it to the Success Condition range, $990{,}000 < x < 1{,}010{,}000\ \Omega$. If $x$ is outside the given range, the NI 5102 has failed this test. Return the digitizer to NI for repair.
6. Repeat steps 2 through 5 for each range listed in the Digitizer Parameters column of Table 7.

7. Move the ohmmeter connection to channel 1 of the digitizer.
8. Repeat steps 2 through 6 for channel 1. Change **channelList** to `1` when calling the function `niScope_configureVertical` and `niScope_Read`.

You have completed verifying the input impedance specifications for the NI 5102.

**Table 7.** NI 5102 Input Impedance Specifications

| Name | Digitizer Parameters | Success Condition |
|---|---|---|
| Input Impedance | **range** = 10, 2, 0.5, and 0.1 | 990,000 < $x$ < 1,010,000 Ω |

## Verifying AC Coupling Cutoff Frequency

To verify the NI 5102 AC coupling cutoff frequency specifications, complete the following steps:

1. Connect a the signal generator to channel 0 of the digitizer.
2. Configure the signal generator for a 1 MΩ load.
3. Set signal generator to the frequency and amplitude listed in Table 8.
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** = `30,000`
   - **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 formula:

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

14. Compare *response* to the value in the Success Condition column in Table 8. If *response* is outside the listed range, the NI 5102 has failed this test. Return the digitizer to NI for repair.
15. Repeat steps 3 through 14 for the last entry in Table 8.
16. Move the signal generator connection to channel 1 of the digitizer.
17. Repeat steps 3 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 5102.

**Table 8.** NI 5102 AC Coupling Cutoff Specifications

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

## Verifying Timing

Complete the following steps to verify the NI 5102 timing specifications:

1. Connect 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 output 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 value falls outside the specified range, terminate the verification procedure and return the digitizer to NI for repair.

10. Set the signal generator to output a 1.8 Vpp, 10 MHz sine wave. This wave is intentionally undersampled. 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. If the error description indicates the measurement failed due to insufficient crosspoints, the device passed this portion of the test.
14. If the return 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:

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

    which is:

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

15. Compare the actual sample rate (*x*) to the success condition, $999{,}950 < x < 1{,}000{,}050$ Hz. If the sample rate is outside this range, return the digitizer to NI for repair.
16. Move the signal generator connection to channel 1 of the digitizer.
17. Repeat steps 3 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 5102.

## 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 9.
4. Call `niScope_ConfigureVertical` with the following parameters:
   - **channelList** = `0`
   - **range** = `10.0`
   - **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** = `20,000,000`
   - **numSumples** = `128`
   - **refPosition** = `50.0`
   - **numRecords** = `1`
   - **enforceRealTime** = `VI_TRUE`
7. Call `niScope_ConfigureTriggerEdge` with the following parameters:
   - **triggerSource** = `NISCOPE_VAL_EXTERNAL`
   - **holdoff** = `0`
   - **delay** = `0`
   - **level** = The trigger level as discussed in step 10
   - **slope** = `NISCOPE_VAL_POSITIVE`
   - **triggerCoupling** = `NI_SCOPE_VAL_AC`
8. Call `niScope_Read` to read a waveform. Set the following parameters:
   - **channelList** = `0`
   - **timeout** = `0.1`
   - **numSamples** = `128`

9. Call `niScope_Abort` to stop the test.
10. If the digitizer did not time out, the channel has passed the test. If the digitizer timed out, increment the level setting by the trigger level delta specified in Table 9, then repeat steps 7 through 10 until the level setting exceeds the high trigger level entry in Table 9. If incrementing the trigger level by the trigger level delta results in a trigger level greater than the high trigger level in Table 9, this channel fails the test.
11. Connect the signal generator to channel 0 of the digitizer.
12. 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`
13. Repeat steps 1 through 12 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 sensitivity test. If any channel fails the test, return the digitizer to NI for repair.

You have completed verifying the trigger sensitivity for the NI 5102.

**Table 9.** NI 5102 Trigger Sensitivity Specifications

| Name | Digitizer Parameters | Stimulus Parameters | Success Condition |
|---|---|---|---|
| Trigger Sensitivity | **low trigger level** = –0.5<br>**high trigger level** = 0.5<br>**trigger level delta** = 0.01 | 300 mVpp | digitizer triggers with any valid trigger level |

## Adjusting the NI 5102

If the verification procedure determines that the digitizer is out of specification, you need to adjust the calibration constants.

The NI 5102 has no internal reference voltage, so external connections are required for vertical range and offset adjustments. The external adjustment is completely independent for each channel and vertical range setting combination. You can perform the verification and adjustment for all the ranges or for only the vertical ranges in question. The adjustment procedure adjusts the calibration constants stored in the external calibration area of the EEPROM and is equivalent to a factory calibration. The procedure is password-protected so that users cannot accidentally modify calibration constants. You can change the password using the function `niScope_CalChangePassword`.

To adjust the calibration constants of the NI 5102, complete the following steps:

1. Call `niScope_CalStart` to obtain a calibration-specific session handle. You must provide the external calibration password, which defaults to `0` or the empty string, `""`. Set the following parameters:
   - **resourceName** = A string assigned by Measurement & Automation Explorer (MAX), such as `DAQ::1`
   - **password** = Default password is `0` or `""`
   - **newSessionHandle** = The returned session handle
2. Short-circuit channel 0 of the digitizer and call `niScope_CalAdjustOffset` to perform a preliminary offset calibration for the 0.1 V range on channel 0. Set the following parameters:
   - **sessionHandle** = The value returned by `niScope_calStart`
   - **channelName** = The string name of the channel, such as `0` or `1`
   - **range** = The voltage range you want to calibrate
3. Connect the calibrator to channel 0.
4. Apply the positive stimulus as specified in the Stimulus Parameters column of Table 10.
5. Call `niScope_CalAdjustRange` to adjust the vertical range of the specified channel. Set the following parameters:
   - **sessionHandle** = The value returned by `niScope_CalStart`
   - **channelName** = The string name of the channel, such as `0` or `1`
   - **range** = The voltage range you want to calibrate
   - **stimulus** = The positive DC voltage you apply; refer to Table 10 for the proper stimulus for each vertical range entry
6. Apply a negative stimulus as specified in the Stimulus Parameters column of Table 10.
7. Call `niScope_CalAdjustRange` to adjust the vertical range of a specified channel. Set the following parameters:
   - **sessionHandle** = The value returned by `niScope_CalStart`
   - **channelName** = The string name of the channel, such as `0` or `1`

- **range** = The voltage range you want to calibrate
- **stimulus** = The negative DC voltage you apply; refer to Table 10 for the proper stimulus for each vertical range entry

8. Short-circuit the input channel and call `niScope_CalAdjustOffset` to perform an offset calibration using the calibrated vertical range from the previous step. Set the following parameters:
   - **sessionHandle** = The value returned by `niScope_CalStart`
   - **channelName** = The string name of the channel, such as `0` or `1`
   - **range** = The voltage range you want to calibrate; refer to Table 10 to obtain the value for each vertical range
9. Repeat steps 2 through 8 for each channel and vertical range you want to adjust.
10. Call `niScope_CalSelfCalibrate` to self-calibrate the digitizer. Set the following parameters:
    - **sessionHandle** = The session handle returned by `niScope_CalStart` or `niScope_init`
    - **channelName** = Ignore this parameter; use `VI_Null`
    - **option** = Use `VI_NULL` for a normal self-calibration operation; only `NISCOPE_VAL_CAL_RESTORE_EXTERNAL_CALIBRATION` is supported
11. Call `niScope_CalEnd` to release the session handle. Set the following parameters:
    - **sessionHandle** = The session handle returned by `niScope_CalStart`
    - **action** = `NISCOPE_VAL_CAL_ACTION_STORE` to store the calibration constants in the EEPROM; this option stores the external calibration date and the incremented external calibration count in the EEPROM

The vertical range and offset constants of the digitizer are now adjusted with respect to the external source.

**Table 10.** NI 5102 Calibration Stimuli

| Calibration Step | Digitizer Parameters | Stimulus Parameters |
|---|---|---|
| Vertical Range | **range** = 10.0 V | ±4.25 VDC |
| Vertical Range | **range** = 2.0 V | ±0.85 VDC |
| Vertical Range | **range** = 0.5 V | ±0.21 VDC |
| Vertical Range | **range** = 0.1 V | ±0.04 VDC |

## Verifying the New Calibration Constants

NI recommends that you reverify the NI 5102 operating specifications after adjustment by repeating the verification procedures in the *Verifying the Performance of the NI 5102* section. If the digitizer fails any test immediately after you perform an external calibration, return the digitizer to NI for repair.

# 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 calibration session.

## Using This Function

By default, the password is set to `NULL` or the empty string, `""`. The password is stored in the EEPROM as an array of four 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. **password** is verified against the password stored in the EEPROM for additional security.

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 to allow password protection. All other functions, such as verification and read 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 application development environment (ADE) before you can open another session.

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

## Parameters

| Name | Description |
|---|---|
| **resourceName** | assigned by MAX; this is a string such as `DAQ::1` |
| **password** | verified against the password in the EEPROM |
| **newSessionHandle** | returned session handle |

# niScope_CalEnd

## Function Prototype

```
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 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 very important to call `niScope_CalEnd` each time you call `niScope_CalStart`, 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 calibration count are automatically stored in the EEPROM.

## Parameters

| Name | Description |
|---|---|
| **sessionHandle** | session handle returned by `niScope_CalStart` |
| **action** | `NISCOPE_VAL_CAL_ACTION_STORE`,<br>`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 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 `""`) |
| **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 the returned count is the self-calibration count or external calibration 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** | returned number of times device 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 date of the last self-calibration or external calibration, or the manufacture date.

## Using This Function

If you are upgrading from NI-SCOPE version 1.5 or earlier, the initial calibration dates will be incorrect because older versions of NI-SCOPE calibration 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 calibration using `niScope_StoreMiscInfo`.

## Using This Function

**info** must be a character array of length five. 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 calibration |

# niScope_CalStoreMiscInfo

## Function Prototype

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

## Purpose

`niScope_CalStoreMiscInfo` allows you 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 `0` 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. 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 of info stored in EEPROM during last external calibration |

# 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

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** | ignore this parameter; use `VI_NULL` |
| **option** | `NISCOPE_VAL_SELF_CALIBRATE` |

# niScope_CalAdjustRange

## Function Prototype

```
ViStatus _VI_FUNC niScope_CalAdjustRange
(
    ViSession sessionHandle,
    ViConstString channelName,
    ViReal64 range,
    ViReal64 stimulus
);
```

## Purpose

`niScope_CalAdjustRange` performs an external calibration of the vertical range for a specified channel and vertical range setting of the NI 5102. Call `niScope_CalAdjustOffset` prior to calling his function, using the same **channelName** and **range** parameters, to perform a preliminary offset calibration using a default vertical range. See the *External Calibration Procedures* section for details on writing a calibration procedure for the NI 5102.

## Parameters

| Name | Description |
|---|---|
| **sessionHandle** | session handle returned by `niScope_CalStart` |
| **channelName** | string name of channel; for example, `0` or `1` |
| **range** | the voltage range to calibrate; see the *External Calibration Procedures* section for the legal values |
| **stimulus** | the DC voltage of the applied signal; see the *External Calibration Procedures* section for the proper stimulus to apply |

# niScope_CalAdjustOffset

## Function Prototype

```
ViStatus _VI_FUNC niScope_CalAdjustOffset
(
   ViSession sessionHandle,
   ViConstString channelName,
   ViReal64 range
);
```

## Purpose

niScope_CalAdjustOffset externally calibrates a specified channel and vertical range of the NI 5102. Call this function before niScope_CalAdjustRange to perform a preliminary offset calibration. The first time you call niScope_CalAdjustOffset, a default vertical range parameter is used in NI-SCOPE to calibrate the vertical offset. When you call the function again after niScope_CalAdjustRange, it performs a precise offset calibration using the calibrated vertical range. See the *External Calibration Procedures* section for details on writing a calibration procedure for the NI 5102.

## Parameters

| Name | Description |
|---|---|
| **sessionHandle** | session handle returned by niScope_CalStart |
| **channelName** | string name of channel; for example, 0 or 1 |
| **range** | the voltage range to calibrate; see the *External Calibration Procedures* section for legal values |