

COMPREHENSIVE SERVICES

We offer competitive repair and calibration services, as well as easily accessible documentation and free downloadable resources.

SELL YOUR SURPLUS

We buy new, used, decommissioned, and surplus parts from every NI series. We work out the best solution to suit your individual needs.

 Sell For Cash  Get Credit  Receive a Trade-In Deal

OBSOLETE NI HARDWARE IN STOCK & READY TO SHIP

We stock **New**, **New Surplus**, **Refurbished**, and **Reconditioned** NI Hardware.



Bridging the gap between the manufacturer and your legacy test system.

 1-800-915-6216

 www.apexwaves.com

 sales@apexwaves.com

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

Request a Quote

 **CLICK HERE**

CB-68LPR

NI 6703/6704 Calibration Procedure for NI-DAQ™ mx

This document contains instructions for calibrating the NI 6703/6704 for PCI/PXI/CompactPCI using NI-DAQmx.

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

The NI 6703/6704 should be calibrated at a regular interval as defined by the measurement accuracy requirements of your application. National Instruments 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 accuracy demands of your application or requirements of your processes.

Contents

| | |
|--|----|
| Conventions | 2 |
| Software | 2 |
| Documentation | 2 |
| Test Equipment | 3 |
| Test Considerations | 4 |
| Calibration Process | 4 |
| Calibration Process Overview | 4 |
| Initial Setup | 4 |
| NI 6703/6704 Verification Procedure | 5 |
| Voltage Output Verification | 5 |
| Current Output Verification (NI 6704 Only) | 7 |
| NI 6703/6704 Adjustment Procedure | 10 |
| Voltage Output Adjustment | 10 |
| Current Output Adjustment (NI 6704 Only) | 13 |

| | |
|-------------------------------|----|
| NI 6703/6704 Test Limits..... | 17 |
| Using the Tables..... | 17 |
| Test Point..... | 17 |
| 24-Hour Limits..... | 17 |
| 1-Year Limits | 18 |

Conventions



The following conventions appear in this manual:

This icon denotes a note, which alerts you to important information.

bold

Bold text denotes items that you must select or click in the software, such as menu items and dialog box options. Bold text also denotes parameter names and hardware labels.

italic

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

monospace

Monospace text denotes text or characters that you should enter from the keyboard, sections of code, programming examples, and syntax examples. This font is also used for the proper names of disk drives, paths, directories, programs, subprograms, subroutines, device names, functions, operations, variables, filenames, and extensions.

monospace italic

Italic text in this font denotes text that is a placeholder for a word or value that you must supply.

Software

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

Documentation

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

- The *NI-DAQmx C Reference Help* includes information about the functions in the driver.
- The *DAQ Getting Started Guide* for NI-DAQ 7.4 or later provides instructions for installing and configuring NI-DAQ devices.

- The *NI-DAQmx Help* includes information about creating applications that use the NI-DAQmx driver.

For more information about the device you are calibrating, refer to the *NI 6703/6704 User Manual*.

Test Equipment

Figure 1 shows the test equipment you need to calibrate your device. The specific DMM connections are described in the [Calibration Process](#) section.

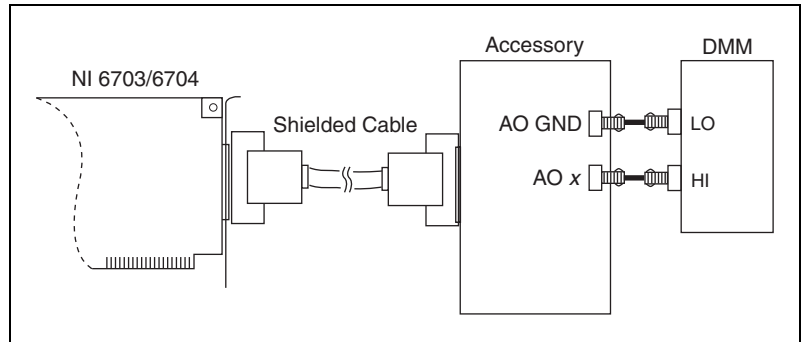


Figure 1. Calibration Connections

When performing calibration, National Instruments recommends that you use the following instruments for calibrating an AO device:

- DMM—NI 4070. If that instrument is unavailable, use a multi-ranging 5 1/2-digit DMM with an accuracy of 40 ppm (0.004%).
- Low thermal copper EMF plug-in cables—Fluke 5440A-7002. Do *not* use standard banana cables.
- DAQ cable—NI recommends using shielded cables, such as the SH68-68-EP.
- One of the following DAQ accessories:
 - SCB-68—The SCB-68 is a shielded I/O connector block with 68 screw terminals for easy signal connection to 68- or 100-pin DAQ devices.
 - CB-68LP/CB-68LPR/TBX-68—The CB-68LP, CB-68LPR, and TBX-68 are low-cost termination accessories with 68 screw terminals for easy connection of field I/O signals to 68-pin DAQ devices.

Test Considerations

Follow these guidelines to optimize connections and test conditions during calibration:

- Keep connections to the NI 6703/6704 short. Long cables and wires act as antennae, picking up extra noise, which can affect measurements.
- Use shielded copper wire for all cable connections to the device.
- Use twisted-pair wire to eliminate noise and thermal offsets.
- Maintain a temperature between 18 and 28 °C. To operate the module at a specific temperature outside this range, calibrate the device at that temperature.
- Keep relative humidity below 80%.
- Allow a warm-up time of at least 15 minutes to ensure that the measurement circuitry is at a stable operating temperature.

Calibration Process

This section provides instructions for verifying and calibrating your device.

Calibration Process Overview

The calibration process has four steps:

1. *Initial Setup*—Configure your device in NI-DAQmx.
2. *NI 6703/6704 Verification Procedure*—Verify the existing operation of the device. This step allows you to confirm that the device was operating within its specified range prior to calibration.
3. *NI 6703/6704 Adjustment Procedure*—Perform an external calibration that adjusts the device calibration constants with respect to a known voltage source.
4. Perform another verification to ensure that the device is operating within its specifications after adjustment.

These steps are described in detail in the following sections. Because a complete verification of all of the device's ranges can take some time, you may wish to verify only the ranges of interest to you.

Initial Setup

NI-DAQmx automatically detects the NI 6703/6704. However, for the driver to communicate with the device, it must be configured in NI-DAQmx.

To configure a device in NI-DAQmx, complete the following steps:

1. Install the NI-DAQmx driver software.
2. Power off the computer that will hold the device and install the device in an available slot.
3. Power on the computer and launch Measurement & Automation Explorer (MAX).
4. Configure the device identifier and select **Self-Test** to ensure that the device is working properly.



Note When a device is configured with MAX, it is assigned a device identifier. Each function call uses this identifier to determine which DAQ device to calibrate.

NI 6703/6704 Verification Procedure

Verification determines how well the DAQ device is meeting its specifications. By performing this procedure, you can see how your device has changed over time. You can use this information to help determine the appropriate calibration interval for your application.

The verification procedure is divided into the major functions of the device. Throughout the verification process, use the tables in the [NI 6703/6704 Test Limits](#) section to determine if your device needs to be adjusted.

Voltage Output Verification

This procedure verifies whether the output of voltage channel 0 is within specifications. Because of the channel-to-channel matching of the NI 6703/6704 voltage channels, if voltage channel 0 is operating within specifications, all the voltage channels should be operating within specifications.

If you want to verify the channels individually, you can expand this procedure to verify all the voltage channels. Perform the procedure once for each value of the channel parameter from 0 to 15. Each time you repeat the procedure, connect the appropriate channel to the DMM.

To verify the voltage output, complete the following steps:

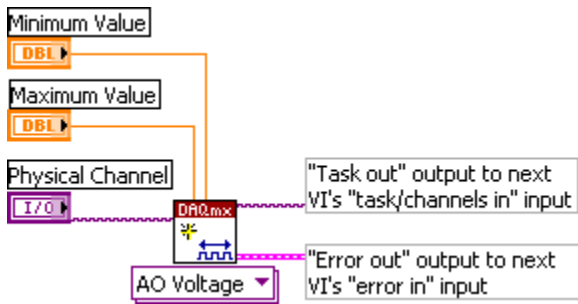
1. Connect the DMM to the NI 6703/6704 as shown in the following table.

| Signal Name | Description | NI 6703/6704 Output | DMM Input |
|-------------|-----------------|---------------------|-----------|
| AO 0 (V) | AO 0 (V) | Pin 34 | High |
| AO GND 0 | Analog Ground 0 | Pin 68 | Low |

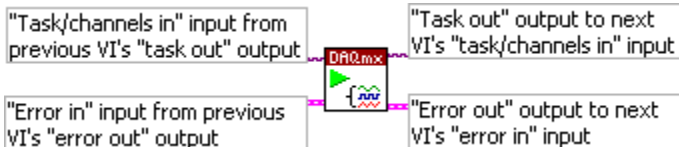
2. Place the DMM in voltage mode.
3. Create a task using DAQmxCreateTask.

| NI-DAQ Function Call | LabVIEW Block Diagram |
|--|-------------------------------------|
| Call DAQmxCreateTask with the following parameters: taskName: <i>MyAOVoltageTask</i> taskHandle: &taskHandle | LabVIEW does not require this step. |

4. Add an AO voltage task using DAQmxCreateAOVoltageChan (DAQmx Create Virtual Channel VI) and configure the channel, AO 0 (V). Use the tables in the [NI 6703/6704 Test Limits](#) section to determine the minimum and maximum values for your device.

| NI-DAQ Function Call | LabVIEW Block Diagram |
|---|--|
| Call DAQmxCreateAOVoltageChan with the following parameters: taskHandle: taskHandle physicalChannel: dev1/ao0 nameToAssignToChannel: AOVoltageChannel minVal: <i>-10.0</i> maxVal: <i>10.0</i> units: DAQmx_Val_Volts customScaleName: NULL |  <p>The diagram shows a DAQmx Create Virtual Channel VI block. It has four inputs: Minimum Value (DBI), Maximum Value (DBI), Physical Channel (I/O), and AO Voltage (dropdown menu). The output is 'Task out' and the error output is 'Error out'.</p> |

5. Start the generation using DAQmxStartTask (DAQmx Start Task VI).

| NI-DAQ Function Call | LabVIEW Block Diagram |
|---|--|
| Call DAQmxStartTask with the following parameters: taskHandle: taskHandle |  <p>The diagram shows a DAQmx Start Task VI block. It has two inputs: 'Task/channels in' and 'Error in'. The output is 'Task out' and the error output is 'Error out'.</p> |

- Generate 9.9 V on AO 0 (V) using DAQmxWriteAnalogF64 (DAQmx Write VI).

| NI-DAQ Function Call | LabVIEW Block Diagram |
|---|-----------------------|
| <p>Call DAQmxWriteAnalogF64 with the following parameters:</p> <p>taskHandle: taskHandle numSampsPerChan: 1 autoStart: 1 timeout: 10.0 dataLayout: DAQmx_Val_GroupByChannel writeArray: &voltage sampsPerChanWritten: &samplesWritten reserved: NULL</p> | |

- Compare the resulting value shown by the DMM to the upper and lower limits listed in Table 1 in the [NI 6703/6704 Test Limits](#) section. If the value is between these limits, the device passes the test.
- Clear the generation using DAQmxClearTask (DAQmx Clear Task VI).

| NI-DAQ Function Call | LabVIEW Block Diagram |
|---|-----------------------|
| <p>Call DAQmxClearTask with the following parameter:</p> <p>taskHandle: taskHandle</p> | |

- Repeat steps 3 through 8 generating 0 V.
- Repeat steps 3 through 8 generating -9.9 V.
- Disconnect the DMM from the NI 6703/6704.

You have now verified the voltage output of the NI 6703/6704.

Current Output Verification (NI 6704 Only)

This procedure verifies that the current output of channel 16 is within specifications. Because of the channel-to-channel matching of the NI 6704 current channels, if current channel 16 is operating within specifications, all the current channels should be operating within specifications.

You can also verify the channels individually for each current channel. Perform the procedure once for each value of the channel parameter from 16 to 31. Each time you repeat the procedure, connect the appropriate channel to the DMM.

To verify current output, complete the following steps:

1. Connect the DMM to the NI 6704 as shown in the following table.

| Signal Name | Description | NI 6704 Output | DMM Input |
|-------------|------------------|----------------|-----------|
| AO 16 (I) | AO 16 (I) | Pin 67 | High |
| AO GND 16 | Analog Ground 16 | Pin 68 | Low |

2. Place the DMM in current mode.
3. Create a task using `DAQmxCreateTask`.

| NI-DAQ Function Call | LabVIEW Block Diagram |
|--|-------------------------------------|
| Call <code>DAQmxCreateTask</code> with the following parameters: taskName: <code>MyAOVoltageTask</code> taskHandle: <code>&taskHandle</code> | LabVIEW does not require this step. |

4. Add an AO voltage task using `DAQmxCreateAOCurrentChan` (**DAQmx Create Virtual Channel VI**) and configure the channel, AO 16 (I). Use the tables in the [NI 6703/6704 Test Limits](#) section to determine the minimum and maximum values for your device.

| NI-DAQ Function Call | LabVIEW Block Diagram |
|---|-----------------------|
| Call <code>DAQmxCreateAOCurrentChan</code> with the following parameters: taskHandle: <code>taskHandle</code> physicalChannel: <code>dev1/ao16</code> nameToAssignToChannel: <code>AOCurrentChannel</code> minVal: 0 maxVal: 0.02 units: <code>DAQmx_Val_Amps</code> customScaleName: <code>NULL</code> | |

- Start the generation using DAQmxStartTask (DAQmx Start Task VI).

| NI-DAQ Function Call | LabVIEW Block Diagram |
|---|-----------------------|
| Call DAQmxStartTask with the following parameters: taskHandle: taskHandle | |

- Generate 0.1 mA on AO 16 (I) using DAQmxWriteAnalogF64 (DAQmx Write VI).

| NI-DAQ Function Call | LabVIEW Block Diagram |
|--|-----------------------|
| Call DAQmxWriteAnalogF64 with the following parameters: taskHandle: taskHandle numSampsPerChan: 1 autoStart: 1 timeout: 10.0 dataLayout: DAQmx_Val_GroupByChannel writeArray: ¤t sampsPerChanWritten: &samplesWritten reserved: NULL | |

- Compare the resulting value shown by the DMM to the upper and lower limits listed in Table 2 in the [NI 6703/6704 Test Limits](#) section. If the value is between these limits, the device passes the test.
- Clear the generation using DAQmxClearTask (DAQmx Clear Task VI).

| NI-DAQ Function Call | LabVIEW Block Diagram |
|--|-----------------------|
| Call DAQmxClearTask with the following parameter: taskHandle: taskHandle | |

9. Repeat steps 3 through 8 generating 20 mA.
10. Disconnect the DMM from the NI 6704.

You have now verified the current output of the NI 6704.

NI 6703/6704 Adjustment Procedure

At the end of each calibration procedure, new constants are stored in the user calibration area of the NI 6703/6704, ensuring that the device stores a permanent record of the calibration constants. Storing the calibration constants here also ensures that the constants automatically load when the NI 6703/6704 is powered on.

Voltage Output Adjustment

For the voltage outputs, you need to make offset and gain adjustments. Adjust offsets first, followed by gains. Adjust offset and gain again to ensure accuracy. This second set of offset and gain adjustments is necessary because a large initial change in one of the calibration channels can affect the accuracy of the other calibration channel.

The following table lists the offset and gain internal channels.

| Calibration Channel | Internal Channel |
|---------------------|------------------------|
| Voltage Offset | _cal_ao_voltage_offset |
| Voltage Gain | _cal_ao_voltage_gain |

To measure the voltage outputs, complete the following steps:

1. Open a calibration session on your device using DAQmxInitExtCal (**DAQmx Initialize External Calibration VI**). The default password is NI.

| NI-DAQ Function Call | LabVIEW Block Diagram |
|---|--|
| <p>Call DAQmxInitExtCal with the following parameters:</p> <p>deviceName: dev1 password: NI calHandle: &calHandle infoType: ND_CALIBRATION_ENABLE infoValue: ND_YES</p> | <p>The diagram illustrates the LabVIEW implementation of the DAQmxInitExtCal function call. It features a 'device in' control with 'I/O' selected, a 'password' control with 'abc' entered, and a 'DAQmx' block. The 'DAQmx' block is connected to a 'DAQmx Initialize External Calibration.vi' block. The output of this block is 'Calhandle out', which is connected to the 'calhandle in' input of the next VI. The error output is 'Error out', which is connected to the 'error in' input of the next VI.</p> |

- Connect the DMM to the NI 6703/6704 as shown in the following table.

| Signal Name | Description | NI 6703/6704 Output | DMM Input |
|-------------|-----------------|---------------------|-----------|
| AO 0 (V) | AO 0 (V) | Pin 34 | High |
| AO GND 0 | Analog Ground 0 | Pin 68 | Low |

- Place the DMM in voltage mode.
- Create a task using DAQmxCreateTask.

| NI-DAQ Function Call | LabVIEW Block Diagram |
|--|-------------------------------------|
| Call DAQmxCreateTask with the following parameters: taskName: <i>MyAOVoltageTask</i> taskHandle: &taskHandle | LabVIEW does not require this step. |

- Add an AO voltage task using DAQmxCreateAOVoltageChan (DAQmx Create Virtual Channel VI) and configure the channel, AO 0 (V).

| NI-DAQ Function Call | LabVIEW Block Diagram |
|---|-----------------------|
| Call DAQmxCreateAOVoltageChan with the following parameters: taskHandle: taskHandle physicalChannel: dev1/ao0 nameToAssignToChannel: AOVoltageChannel minVal: -10.24 maxVal: 10.24 units: DAQmx_Val_Volts customScaleName: NULL | |

- Start the generation using DAQmxStartTask (DAQmx Start Task VI).

| NI-DAQ Function Call | LabVIEW Block Diagram |
|---|-----------------------|
| Call DAQmxStartTask with the following parameters: taskHandle: taskHandle | |

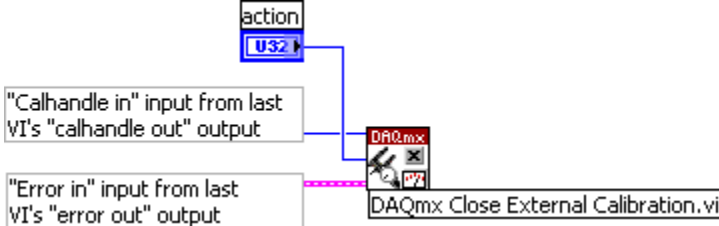
- Generate 0.0 V on AO 0 (V) using DAQmxWriteAnalogF64 (DAQmx Write VI).

| NI-DAQ Function Call | LabVIEW Block Diagram |
|--|-----------------------|
| Call DAQmxWriteAnalogF64 with the following parameters: taskHandle: taskHandle numSampsPerChan: 1 autoStart: 1 timeout: 10.0 dataLayout: DAQmx_Val_GroupByChannel writeArray: &voltage sampsPerChanWritten: &samplesWritten reserved: NULL | |

- Repeat steps 4 through 7 replacing AO 0 (V) with `_cal_ao_voltage_offset`. Continue to repeat step 7 with various values in `_cal_ao_voltage_offset` until you have generated approximately 0 V on AO 0 (V).
- Clear the generation using DAQmxClearTask (DAQmx Clear Task VI).

| NI-DAQ Function Call | LabVIEW Block Diagram |
|--|-----------------------|
| Call DAQmxClearTask with the following parameter: taskHandle: taskHandle | |

10. Repeat steps 4 through 9 generating 9.9 V on AO 0 (V), and replace `_cal_ao_voltage_offset` with `_cal_ao_voltage_gain`. Continue to repeat step 7 with various values in `_cal_ao_voltage_gain` until you have generated approximately 9.9 V on AO 0 (V).
11. Repeat steps 4 through 10 to perform another set of offset and gain adjustments until they are within the values listed in the *24-Hour Limits* column of Table 1 in the *NI 6703/6704 Test Limits* section.
12. Save the adjustment to the EEPROM, or onboard memory, using `DAQmxCloseExtCal` (**DAQmx Close External Calibration VI**). This function also saves the date, time, and temperature of the adjustment to the onboard memory.

| NI-DAQ Function Call | LabVIEW Block Diagram |
|---|--|
| <p>Call <code>DAQmxCloseExtCal</code> with the following parameters:</p> <p>calHandle: <code>calHandle</code></p> <p>action: <code>DAQmx_Val_Action_Commit</code></p> |  |

13. Disconnect the DMM from the device.

You have now adjusted the voltage output.

Current Output Adjustment (NI 6704 Only)

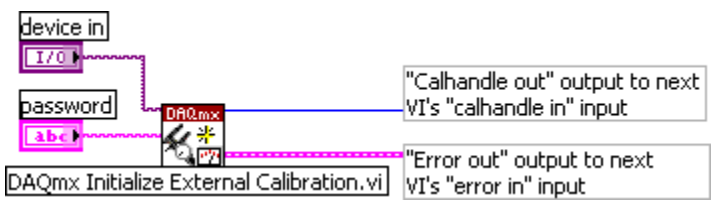
For the current outputs, you need to adjust offset and gain. Adjust offset first, followed by gain. Adjust both offset and gain again to ensure accuracy. This second set of offset and gain adjustments is necessary because a large initial change in one of the calibration channels can affect the accuracy of the other calibration channel.

The following table lists the offset and gain internal channels.

| Calibration Channel | NI 6704 Internal Channel |
|---------------------|-------------------------------------|
| Current Offset | <code>_cal_ao_current_offset</code> |
| Current Gain | <code>_cal_ao_current_gain</code> |

To adjust the current outputs, complete the following steps:

1. Open a calibration session on your device using `DAQmxInitExtCal` (**DAQmx Initialize External Calibration VI**). The default password is NI.

| NI-DAQ Function Call | LabVIEW Block Diagram |
|--|--|
| <p>Call <code>DAQmxInitExtCal</code> with the following parameters:</p> <p>deviceName: dev1 password: NI calHandle: &calHandle infoType: ND_CALIBRATION_ENABLE infoValue: ND_YES</p> |  |

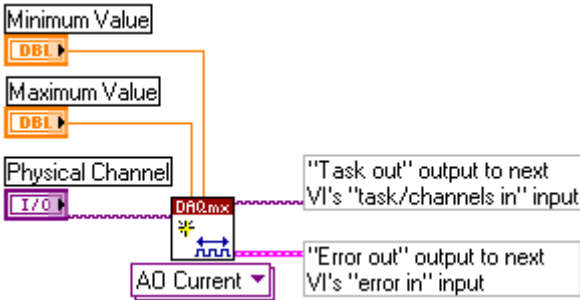
2. Connect the DMM to the NI 6704 as shown in the following table.

| Signal Name | Description | NI 6704 Output | DMM Input |
|-------------|------------------|----------------|-----------|
| AO 16 (I) | AO 16 (I) | Pin 67 | High |
| AO GND 16 | Analog Ground 16 | Pin 68 | Low |

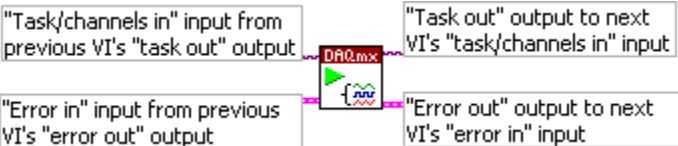
3. Place the DMM in current mode.
4. Create a task using `DAQmxCreateTask`.

| NI-DAQ Function Call | LabVIEW Block Diagram |
|--|--|
| <p>Call <code>DAQmxCreateTask</code> with the following parameters:</p> <p>taskName: <i>MyAOVoltageTask</i> taskHandle: &taskHandle</p> | <p>LabVIEW does not require this step.</p> |

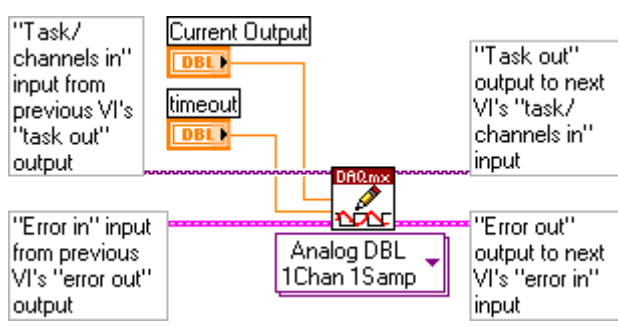
- Add an AO voltage task using DAQmxCreateAOCurrentChan (DAQmx Create Virtual Channel VI) and configure the channel, AO 16.

| NI-DAQ Function Call | LabVIEW Block Diagram |
|--|--|
| <p>Call DAQmxCreateAOCurrentChan with the following parameters:</p> <p>taskHandle: taskHandle physicalChannel: dev1/ao16 nameToAssignToChannel: AOCurrentChannel minVal: 0 maxVal: 0.0204 units: DAQmx_Val_Amps customScaleName: NULL</p> |  |

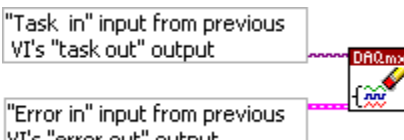
- Start the generation using DAQmxStartTask (DAQmx Start Task VI).

| NI-DAQ Function Call | LabVIEW Block Diagram |
|--|--|
| <p>Call DAQmxStartTask with the following parameters:</p> <p>taskHandle: taskHandle</p> |  |

7. Generate 0.1 mA on AO 16 (I) using DAQmxWriteAnalogF64 (DAQmx Write VI).

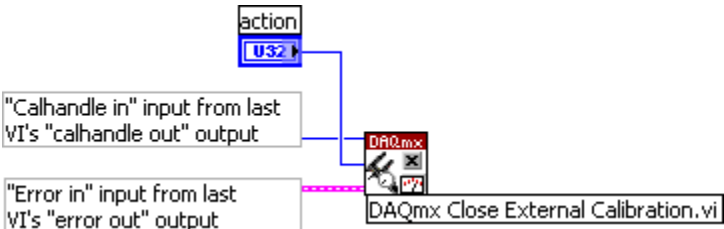
| NI-DAQ Function Call | LabVIEW Block Diagram |
|---|--|
| <p>Call DAQmxWriteAnalogF64 with the following parameters:</p> <p>taskHandle: taskHandle numSampsPerChan: 1 autoStart: 1 timeout: 10.0 dataLayout: DAQmx_Val_GroupByChannel writeArray: &current sampsPerChanWritten: &samplesWritten reserved: NULL</p> |  |

8. Repeat steps 4 through 7 replacing AO 16 (I) with `_cal_ao_current_offset`. Continue to repeat step 7 with various values in `_cal_ao_current_offset` until you have generated approximately 0.1 mA on AO 16 (I).
9. Clear the generation using DAQmxClearTask (DAQmx Clear Task VI).

| NI-DAQ Function Call | LabVIEW Block Diagram |
|---|---|
| <p>Call DAQmxClearTask with the following parameter:</p> <p>taskHandle: taskHandle</p> |  |

10. Repeat steps 4 through 9 generating 20.0 mA on AO 16 (I), and replace `_cal_ao_current_offset` with `_cal_ao_current_gain`. Continue to repeat step 7 with various values in `_cal_ao_current_gain` until you have generated approximately 20.0 mA on AO 16 (I).
11. Repeat steps 4 through 10 to perform another set of offset and gain adjustments until they are within the values listed in the *24-Hour Limits* column of Table 2 in the *NI 6703/6704 Test Limits* section.

12. Save the adjustment to the EEPROM, or onboard memory, using DAQmxCloseExtCal (**DAQmx Close External Calibration VI**). This function also saves the date, time, and temperature of the adjustment to the onboard memory.

| NI-DAQ Function Call | LabVIEW Block Diagram |
|--|--|
| <p>Call DAQmxCloseExtCal with the following parameters:</p> <p>calHandle: calHandle</p> <p>action: DAQmx_Val_Action_Commit</p> |  |

13. Disconnect the DMM from the device.

You have now adjusted the current output.

NI 6703/6704 Test Limits

The tables in this section list the accuracy specifications to use when verifying and adjusting the NI 6703/6704. The tables display the specifications for both 1-year and 24-hour calibration intervals. The 1-year ranges display the specifications that the devices should meet if it has been one year between calibrations. When a device has been calibrated with an external source, the values shown in the 24-hour tables are the valid specifications.

Using the Tables

The following definitions describe how to use the information from the tables in this section.

Test Point

The *Test Point* is the voltage or current value that is generated for verification purposes. *Value* refers to the voltage or current value to be verified and is in volts or milliamps.

24-Hour Limits

The *24-Hour Limits* column contains the *Upper Limit* and *Lower Limit* for the test point value. That is, when the device is within its 24-hour calibration interval, the test point value should fall between the upper and lower limit values. Upper and lower limits are expressed in volts.

1-Year Limits

The *1-Year Limits* column contains the *Upper Limit* and *Lower Limit* for the test point value. That is, when the device is within its 1-year calibration interval, the test point value should fall between the upper and lower limit values. Upper and lower limits are expressed in volts.

Table 1. NI 6703/6704 Voltage Output Specifications

| Test Point Value (V) | 24-Hour Limits | | 1-Year Limits | |
|----------------------|-----------------|-----------------|-----------------|-----------------|
| | Lower Limit (V) | Upper Limit (V) | Lower Limit (V) | Upper Limit (V) |
| 9.9 | 9.8991019 | 9.9008981 | 9.8989435 | 9.9010565 |
| 0.0 | -0.0007100 | 0.0007100 | -0.0007100 | 0.0007100 |
| -9.9 | -9.9008981 | -9.8991019 | -9.9010565 | -9.8989435 |

Table 2. Current Output Specifications (NI 6704 Only)

| Test Point Value (mA) | 24-Hour Limits | | 1-Year Limits | |
|-----------------------|------------------|------------------|------------------|------------------|
| | Lower Limit (mA) | Upper Limit (mA) | Lower Limit (mA) | Upper Limit (mA) |
| 0.1 | 0.0985616 | 0.1014384 | 0.09855 | 0.10145 |
| 20 | 19.997885 | 20.002115 | 19.995565 | 20.004435 |

National Instruments, NI, ni.com, and LabVIEW are trademarks of National Instruments Corporation. Refer to the *Terms of Use* section on ni.com/legal for more information about National Instruments trademarks. Other product and company names mentioned herein are trademarks or trade names of their respective companies. For patents covering National Instruments products, refer to the appropriate location: **Help>Patents** in your software, the `patents.txt` file on your CD, or ni.com/patents.