Data Acquisition Using LabVIEW™ NXG

Online Participant Guide

Course Software Version 2.1
May 2018 Edition

Trademarks

Refer to the *NI Trademarks and Logo Guidelines* at ni.com/trademarks for more information on National Instruments trademarks.

ARM, Keil, and µVision are trademarks or registered of ARM Ltd or its subsidiaries.

LEGO, the LEGO logo, WEDO, and MINDSTORMS are trademarks of the LEGO Group.

TETRIX by Pitsco is a trademark of Pitsco, Inc.

FIELDBUS FOUNDATION™ and FOUNDATION™ are trademarks of the Fieldbus Foundation.

EtherCAT® is a registered trademark of and licensed by Beckhoff Automation GmbH.

CANopen® is a registered Community Trademark of CAN in Automation e.V.

DeviceNet™ and EtherNet/IP™ are trademarks of ODVA.

Go!, SensorDAQ, and Vernier are registered trademarks of Vernier Software & Technology. Vernier Software & Technology and vernier.com are trademarks or trade dress.

Xilinx is the registered trademark of Xilinx, Inc.

Taptite and Trilobular are registered trademarks of Research Engineering & Manufacturing Inc.

FireWire® is the registered trademark of Apple Inc.

Linux® is the registered trademark of Linus Torvalds in the U.S. and other countries.

Handle Graphics®, MATLAB®, Real-Time Workshop®, Simulink®, Stateflow®, and xPC TargetBox® are registered trademarks, and TargetBox™ and Target Language Compiler™ are trademarks of The MathWorks, Inc.

Tektronix®, Tek, and Tektronix, Enabling Technology are registered trademarks of Tektronix, Inc.

The Bluetooth® word mark is a registered trademark owned by the Bluetooth SIG, Inc.

The ExpressCard™ word mark and logos are owned by PCMCIA and any use of such marks by National Instruments is under license.

The mark LabWindows is used under a license from Microsoft Corporation. Windows is a registered trademark of Microsoft Corporation in the United States and other countries.

Other product and company names mentioned herein are trademarks or trade names of their respective companies.

Members of the National Instruments Alliance Partner Program are business entities independent from National Instruments and have no agency, partnership, or joint-venture relationship with National Instruments.

Patents

For patents covering National Instruments products/technology, refer to the appropriate location: **Help»Patents** in your software, the patents.txt file on your media, or the *National Instruments Patent Notice* at ni.com/patents.

Worldwide Technical Support and Product Information

ni.com

Worldwide Offices

Visit ni.com/niglobal to access the branch office websites, which provide up-to-date contact information, support phone numbers, email addresses, and current events.

National Instruments Corporate Headquarters

11500 North Mopac Expressway   Austin, Texas 78759-3504   USA   Tel: 512 683 0100

For further support information, refer to the *Additional Information and Resources* appendix. To comment on National Instruments documentation, refer to the National Instruments website at ni.com/info and enter the Info Code feedback.

Table of Contents

Student Guide

In this student guide, you will learn about the course description and the items you need to get started in this course.

Topics

- What You Need To Get Started
- Installing the Course Software
- Course Learning Map

A. What You Need To Get Started

Before you use take this course, make sure you have all of the following items:

☐ Computer running Windows 7 or later

☐ LabVIEW NXG 2.0 and NI-DAQmx 17.5

☐ *Data Acquisition Using LabVIEW NXG* course USB card, from which you install the following folders:

Directory	Table Head
`Exercises\DAQNXG\`	Contains files used in the course
`Solutions\DAQNXG\`	Contains completed course exercises

B. Installing the Course Software

File Locations

Exercise files are located in the `<Exercises>\` folder where `<Exercises>` refers to `C:\Exercises\` assuming that you installed the files on your root directory.

Solution files are located in the `<Solutions>\` folder where `<Solutions>` refers to `C:\Solution\` assuming that you installed the files on your root directory.

Note For Participant Guide updates and corrections, refer to `ni.com/info` and enter the Info Code `daqnxgerrata`.

C. Course Learning Map

		
• Analog Input • Analog Output • Digital I/O • Specific Sensors and Signals	• DAQmx API • Multiple Channels • Triggers • Timing and Synchronization • Data Logging	• Other System Considerations

1 Measuring Voltage or Current (Analog Input)

Topics

+ Resources

Exercises

Resources

Resources Refer to the following resource for more information. For locations with info codes, go to `ni.com/info` and enter the code to access the article.

Sample Rate

Article	Location
Acquiring an Analog Signal: Bandwidth, Nyquist Sampling Theorem, and Aliasing	Info code: `nyquist`

Different Types of ADCs

Article	Location
Benefits of Delta-Sigma Analog-to-Digital Conversion	Info code: `deltasigma`
Simultaneous Sampling Data Acquisition Architectures	Info code: `sampledata`

Isolation

Article	Location
Isolation Types and Considerations when Taking a Measurement	Info code: `isolationtype`
Isolation Technologies for Reliable Industrial Measurements	Info code: `isolationtech`

Article	Location
High Voltage Measurement and Isolation	Info code: `voltageisolation`
Isolation and Safety Standards for Electronic Instruments	Info code: `isolationsafety`

Filtering

Article	Location
Median and Nth Order Filtering	Info code: `median`
Digital Filtering	Info code: `filtering`

Field Wiring and Grounding

Article	Location
Field Wiring and Noise Considerations for Analog Signals	Info code: `fieldwiring`

Using Test Panels

Article	Location
Using Test Panels in Measurement & Automation Explorer for Devices Supported by NI-DAQmx	Info code: `testpanels`

Meausring current

Article	Location
How to Measure Voltage, Current, and Power	Info code: `currentpower`
Fundamentals, System Design, and Setup for the 4 to 20 mA Current Loop	Info code: `currentloop`

Article	Location
Current Measurements: How-To Guide	Info code: `currenthowto`
How to Minimize Errors for Low-Current Measurements	Info code: `currenterrors`

Determining the Accuracy of a System

Article	Location
Measurement Accuracy of a Data Acquisition Board	Info code: `ma101`
How Do I Calculate Absolute Accuracy or System Accuracy?	Info code: `sysacc`
Understanding Instrument Specifications -- How to Make Sense Out of the Jargon	Info code: `instrumspecs`
Absolute Accuracy of Dynamic Signal Acquisition Devices	Info code: `accdsa`
Understanding Frequency Performance Specifications	Info code: `performspecs`
Specifications Explained	Info code: `specexplain`

Exercise 1-1 Sample Rate and Aliasing

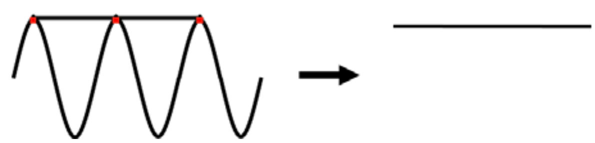
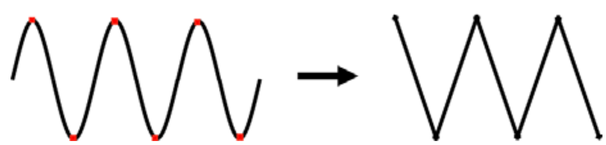
Goal

Explore the effects of sample rate on an input signal.

Scenario

The Nyquist Theorem states that you must sample at greater than two times the maximum frequency component of your signal to accurately represent the frequency of the signal. However, the Nyquist Theorem only deals with accurately representing the *frequency* of the signal. In order to properly represent the shape of your signal, you must sample between 5 to 10 times greater than the maximum frequency component of your signal.

Nyquist Frequency (f_n) is $f_n = ½ f_s$. Below is an example of a 100 Hz sine wave sampled at different frequencies.

Aliased signal		
	100 Hz Sine Wave	Sampled at 100 Hz
Adequately sampled signal for frequency only		
	100 Hz Sine Wave	Sampled at 200 Hz
Adequately sampled signal for frequency and shape		
	100 Hz Sine Wave	Sampled at 1 kHz

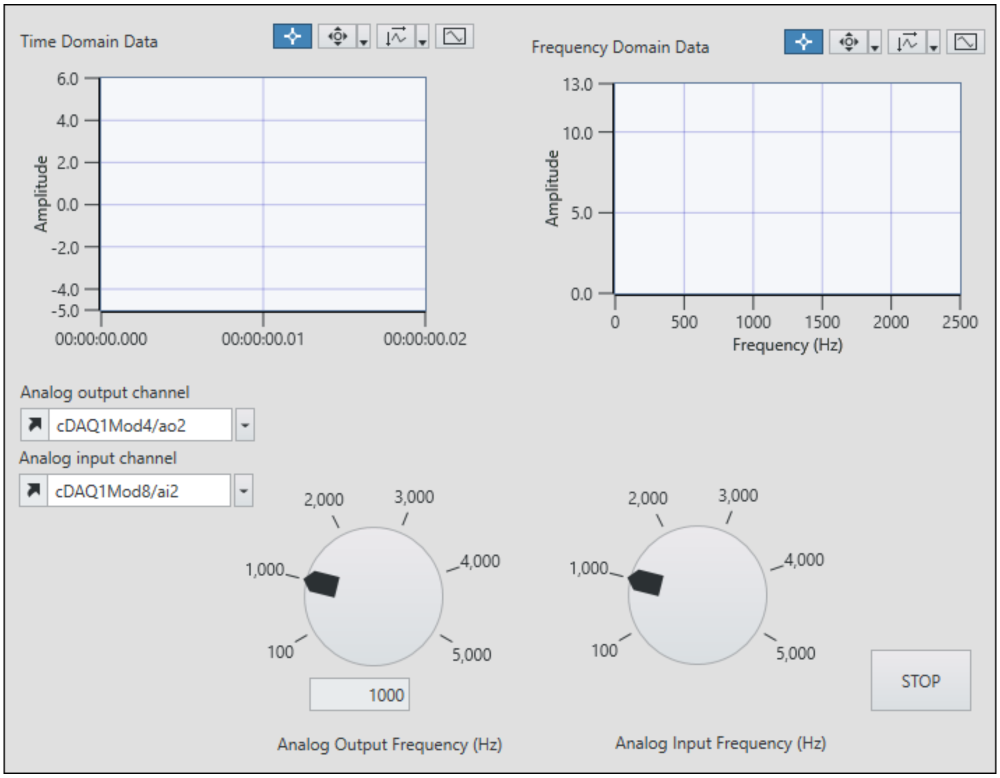
In this exercise, you use the Sample Rate Example VI to demonstrate aliasing and the effects of sample rate on an input signal.

Implementation

1. Open the Sample Rate Example project located in the `<Exercises>\DAQNXG\Aliasing\` directory.

2. Open Sample Rate Example VI.

Figure 1-1. Sample Rate Example VI Panel



This VI acquires a sine wave that is generated by the analog output circuitry on a DAQ device. The VI graphs both the time domain and frequency domain of the acquired signal.

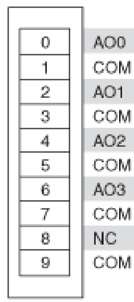
3. Using wires, connect an analog output channel to an analog input channel in the demo box.

    a. Open SystemDesigner and select the C Series Voltage Output Module (NI 9263).

    b. On the **Item** tab, click **Documentation»Pinout** to locate the pins for generating the analog output signal.

Signal Type	Module Options
Analog output	NI 9263

Figure 1-2. Device Pinout for the NI 9263

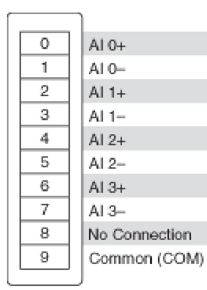


    c. In SystemDesigner and select the C Series Voltage Input Module (NI 9215).

    d. On the **Item** tab, click **Documentation»Pinout** to locate the pins for measuring the analog input signal.

Signal Type	Module Options
Analog input	NI 9215

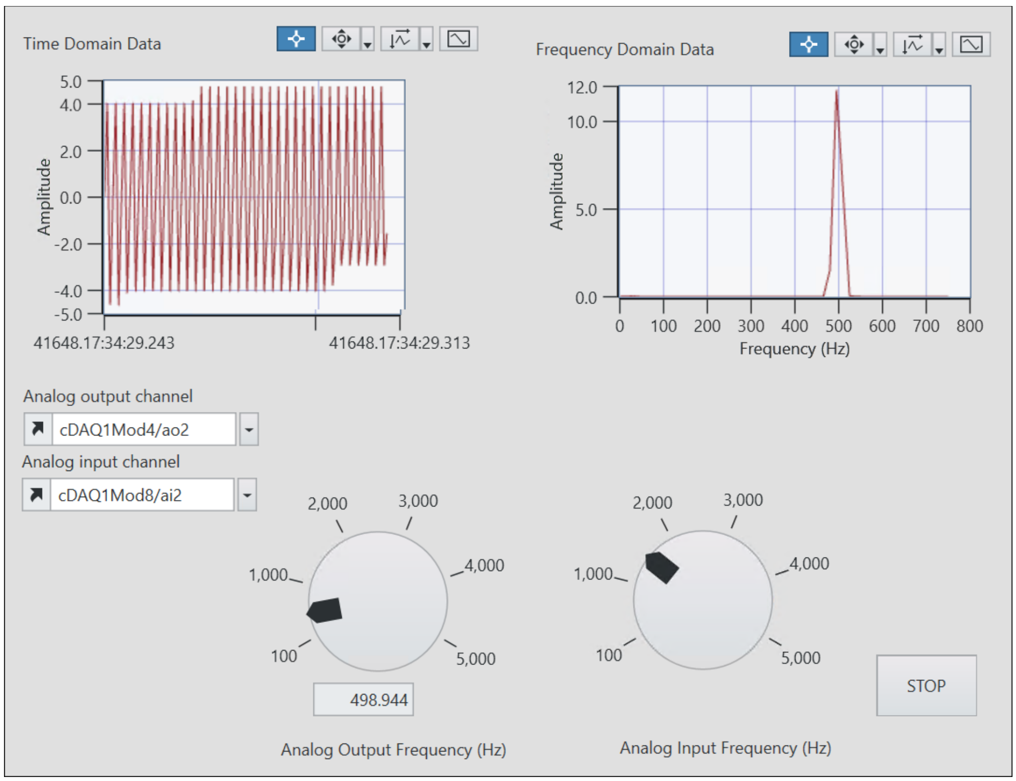
Figure 1-3. Device Pinouts for NI 9215



    e. Using a wire and screwdriver, connect AO2 on the NI 9263 module to AI2+ on the NI 9215 module.

    f. Using a different wire, connect COM on the NI 9263 module to AI2- on the NI 9215 modules.

4. Set the panel controls with the following values, where *cDAQx* is the identity of your chassis.
   - **Analog Output Channel**: `cDAQxMod4/ao2`
   - **Analog Input Channel**: `cDAQxMod8/ai2`
   - **Analog Output Frequency (Hz)**: `500`
   - **Analog Input Frequency (Hz)**: `1500`
5. Run the VI. The x coordinate of the peak you see on the Frequency plot represents the frequency of the sine wave the DAQ device generates.

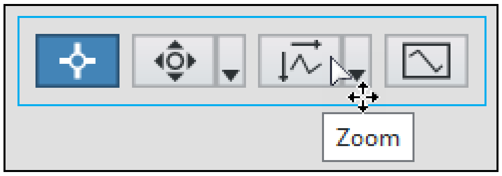


Remember that the Nyquist Frequency (f_n) is $f_n = ½ f_s$.

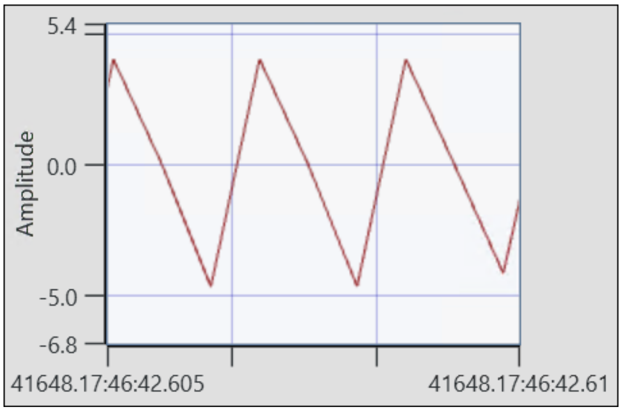
With a sample rate of 1,500 Hz, the Nyquist frequency is 750 Hz. This implies that the sample rate is sufficient to measure a sine wave up to 750 Hz. When you run the VI, you see a peak at 500 Hz, which is the analog output frequency that the DAQ device generates.

6. Stop the VI. Click the Zoom button, as shown in Figure 1-4, and drag on the Time Domain Data graph to zoom in on the data.

Figure 1-4. Graph Tools - Zoom Button



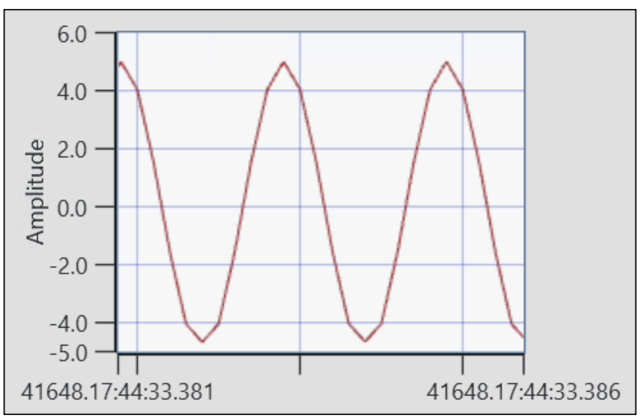
The data looks like a triangular wave. Because you are sampling three times faster than the analog output frequency, you are satisfying the Nyquist Theorem, but you are not capturing the shape of the signal. Notice on the Frequency Domain Data graph that you have captured the correct frequency of the signal.



7. Run the VI.

   a. Increase the **Analog Input Frequency** to `5,000` Hz.

   b. Stop the VI and zoom in on the Time Domain Data graph. The shape of the time domain signal looks like a smooth sine wave.



Increasing the sample rate ten times faster than the signal you are trying to acquire more accurately represents the shape of the signal.

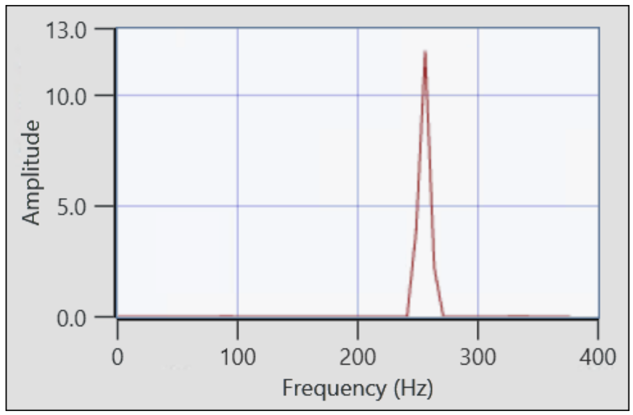
Tip In general, try to acquire a signal 5 to 10 times faster than the highest frequency in the signal you are trying to capture.

8. Run the VI to explore aliasing of a 500 Hz signal.

    a. Decrease the **Analog Input Frequency** to 750 Hz.

    b. What is the Nyquist frequency for sampling frequency of 750 Hz?__________

    c. What is the Frequency displayed on the Frequency Domain Data graph?______



    d. Does the frequency displayed on the graph match the **Analog Output Frequency** control value?____

The Nyquist frequency, f_n, is equal to 375 Hz, which is lower than the analog output signal frequency. Although the time-domain data waveform appears sinusoidal in nature, the signal measured has the incorrect frequency, as displayed on the frequency domain chart.

The alias frequency is determined by the following formula:

Alias freq. = | (closest integer multiple of the sampling freq. – signal freq.) |

Therefore, | 750 – 500 | = 250 Hz, which is what you see on the Frequency Domain Data graph.

The Frequency Domain Data graph displays the incorrect frequency because the frequency has been aliased between 0 and 375 Hz. The 500 Hz signal has been aliased to 250 Hz.

9. Stop and close the VI. Do not save changes.

Note When you select a sample rate to obtain time-domain information such as the shape of the waveform, you must oversample at a rate of at least 5x greater than the highest frequency component in the waveform. If you want to obtain only the frequency information, oversample at least 2x greater than the highest frequency component in the waveform, according to the Nyquist Theorem.

End of Exercise 1-1

Exercise 1-2 Calculating Range, Resolution, and Accuracy

Goal

Determine the optimal configuration for a data acquisition measurement system.

Scenario

In this exercise, you will review the specifications for a project and determine the most appropriate DAQ device and input range for that device.

When choosing a DAQ device, consider performance against cost. A higher resolution DAQ device costs more but provides a more accurate representation of the acquired signal.

1. First determine if the DAQ device offers an input range that meets the requirements of the project.
2. Next, determine if the desired code width is within the capability of the DAQ device.

Tip Use the code width equation.

$$\text{code width} = \frac{\text{Device Input Range}}{2^{\text{resolution in bits}}}$$

3. Finally, use the Absolute Accuracy column of the table to determine if the absolute accuracy of the DAQ device meets the measurement requirements of the project.

In this exercise you will have two project scenarios. For each scenario, select the appropriate DAQ hardware and optimum input range to maximize accuracy. Tables 1-1, 1-2, and 1-3 list the three DAQ devices you can use for the project and their resolution, input ranges, and accuracy.

Tables 1-1, 1-2, and 1-3 display the values for the analog input (AI) absolute accuracy of each nominal range for each DAQ device.

Table 1-1. DAQ Device 1 AI Absolute Accuracy

	Resolution	Nominal Range (V)		Absolute Accuracy at Full Scale (μV)
		Positive Full Scale	Negative Full Scale	
DAQ Device 1	16 Bit	10	–10	3,100
		5	–5	1,620
		1	–1	360
		0.2	–0.2	112

Table 1-2. DAQ Device 2 AI Absolute Accuracy

	Resolution	Nominal Range (V)		Absolute Accuracy at Full Scale (μV)
		Positive Full Scale	Negative Full Scale	
DAQ Device 2	16 Bit	10	–10	1,920
		5	–5	1,010
		2	–2	410
		1	–1	220
		0.5	–0.5	130
		0.2	–0.2	74
		0.1	–0.1	52

Table 1-3. DAQ Device 3 AI Absolute Accuracy

	Resolution	Nominal Range (V)		Absolute Accuracy at Full Scale (μV)
		Positive Full Scale	Negative Full Scale	
DAQ Device 3	18 Bit	10	–10	980
		5	–5	510
		2	–2	210
		1	–1	120
		0.5	–0.5	70
		0.2	–0.2	39
		0.1	–0.1	28

Practice—Project 1 (Answers on page 1-15.)

1. A thermocouple is attached to the steam drum output of a high-pressure boiler system. The thermocouple can measure a temperature range of –270 °C to 1,372 °C. With this temperature range, the thermocouple returns a voltage of –6.548 mV to 54.874 mV. Which DAQ devices offer an acceptable input range for this application?

   a. DAQ Device 1

   b. DAQ Device 2

   c. DAQ Device 3

2. To detect a change of 3.74 μV, which DAQ devices offer an acceptable code width for the project?

   a. DAQ Device 1

   b. DAQ Device 2

   c. DAQ Device 3

3. The measurements need to be within 37.4 μV of their true value. Which DAQ devices offer an acceptable analog input absolute accuracy for the project?

   a. DAQ Device 1

   b. DAQ Device 2

   c. DAQ Device 3

Practice—Project 2

1. A pressure transducer is placed on the intake manifold of an engine. The transducer outputs a voltage between –2 V and 2 V for a linear pressure range of 20 kPa to 105 kPa. Which DAQ devices offer an acceptable input range for this application?

   a. DAQ Device 1

   b. DAQ Device 2

   c. DAQ Device 3

2. This project needs to detect a change of 1.5 Pa, which would be a voltage change of 70 μV. Which DAQ devices and corresponding input ranges offer an acceptable code width for the project?

   a. DAQ Device 1

   b. DAQ Device 2

   c. DAQ Device 3

3. The measurements need to be within 5 Pa, which would be within 235 μV, of their true value. Which DAQ devices offer an acceptable analog input absolute accuracy for the project?

    a. DAQ Device 1

    b. DAQ Device 2

    c. DAQ Device 3

Answers from page 1-13

Project 1

1. In this project, the thermocouple outputs a voltage ranging from –6.548 mV to 54.874 mV. All three DAQ devices have a maximum input range of ± 10 V, which more than covers the voltage range of this project.

2. Using the code width equation, if you select:

   - DAQ Device 1 using an input range of ± 0.2 V, the code width is 6.10 μV, so this DAQ device cannot detect a change of 3.74 μV.
   - DAQ Device 2 using an input range of ± 0.1 V, the code width is 3.05 μV.
   - DAQ Device 3 using an input range of ± 0.1 V, the code width is 0.76 μV.

   DAQ Device 2 and DAQ Device 3 are both able to detect a change of 3.74 μV.

3. AI absolute accuracy by device:

   - DAQ Device 1: 112 μV using a ± 0.2 V input range
   - DAQ Device 2: 52 μV using a ± 0.1 V input range
   - DAQ Device 3: 28 μV using a ± 0.1 V input range

   Only DAQ Device 3 is accurate enough to acquire measurements within 37.4 μV of their true value.

   Therefore, DAQ Device 3, using an input range of ± 0.1 V, is the best DAQ device for this project.

Project 2

1. In this project, the transducer is linear and outputs a voltage ranging from –2 V to 2 V. All three DAQ devices have a maximum input range of ± 10 V, which more than meets the ± 2 V range of this project.

2. Using the code width equation:

   - DAQ Device 1 can detect a change of 152 μV using an input range of ± 5 V.
   - DAQ Device 2 can detect a change of 61 μV using an input range of ± 2 V.
   - DAQ Device 3 can detect a change of 15 μV using an input range of ± 2 V.

   DAQ Device 2 and DAQ Device 3 are both able to detect a change of 70 μV.

3. AI absolute accuracy by device:

    - DAQ Device 1: 1620 μV using a ± 5 V input range
    - DAQ Device 2: 410 μV using a ± 2 V input range
    - DAQ Device 3: 210 μV using a ± 2 V input range

    Only DAQ Device 3 is accurate enough to acquire measurements within 235 μV of their true value.

    Therefore, DAQ Device 3, using an input range of ± 2 V, is the best DAQ device for this project.

End of Exercise 1-2

Exercise 1-3 Connecting Signal Sources with Measurement Systems

Goal

Choose a grounding mode for a measurement system and properly connect signals to that measurement system.

Scenario

In this two-part exercise, you practice determining a correct grounding mode configuration.

Part I (Answers on page 1-19.)

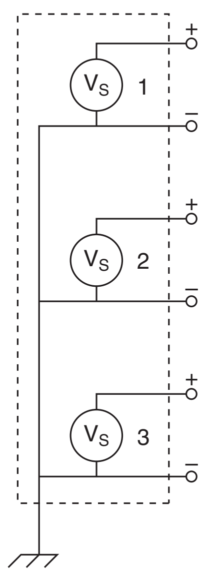
Assume you have an instrument that plugs into a standard wall outlet. The outputs of the instrument are referenced to the same ground as the instrument. Cross out any measurement system you could not use to connect the outputs of the instrument to a DAQ device. Which system would be the most desirable to use and why?

Differential RSE NRSE

Figure 1-5 shows the instrument and a 68-pin layout for a PCIe-6321. Based on your choice for the grounding mode of the measurement system, draw the connections for the following:

- voltage source 1 to analog input channel 0
- voltage source 2 to analog input channel 1
- voltage source 3 to analog input channel 2

Figure 1-5. PCIe-6321 68-pin Layout



AI 0	68	34	AI 8
AI GND	67	33	AI 1
AI 9	66	32	AI GND
AI 2	65	31	AI 10
AI GND	64	30	AI 3
AI 11	63	29	AI GND
AI SENSE	62	28	AI 4
AI 12	61	27	AI GND
AI 5	60	26	AI 13
AI GND	59	25	AI 6
AI 14	58	24	AI GND
AI 7	57	23	AI 15
AI GND	56	22	AO 0
AO GND	55	21	AO 1
AO GND	54	20	NC
D GND	53	19	P0.4
P0.0	52	18	D GND
P0.5	51	17	P0.1
D GND	50	16	P0.6
P0.2	49	15	D GND
P0.7	48	14	+5 V
P0.3	47	13	D GND
PFI 11/P2.3	46	12	D GND
PFI 10/P2.2	45	11	PFI 0/P1.0
D GND	44	10	PFI 1/P1.1
PFI 2/P1.2	43	9	D GND
PFI 3/P1.3	42	8	+5 V
PFI 4/P1.4	41	7	D GND
PFI 13/P2.5	40	6	PFI 5/P1.5
PFI 15/P2.7	39	5	PFI 6/P1.6
PFI 7/P1.7	38	4	D GND
PFI 8/P2.0	37	3	PFI 9/P2.1
D GND	36	2	PFI 12/P2.4
D GND	35	1	PFI 14/P2.6

Part II (Answers on page 1-19.)

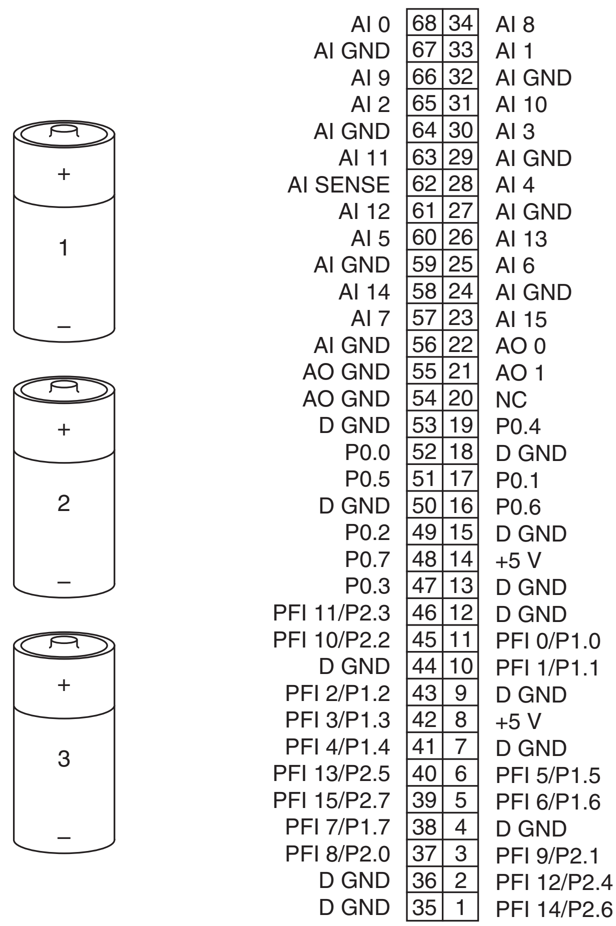
Assume you have three batteries. Cross out any measurement system you could not use to connect the outputs of the batteries to a DAQ device. Which system would be the most desirable to use and why?

Differential RSE NRSE

Figure 1-6 shows the batteries and a pin-out layout for a USB-6361. Based on your choice for the grounding mode of the measurement system, draw the connections for the following:

- battery 1 to analog input channel 5
- battery 2 to analog input channel 6
- battery 3 to analog input channel 7

Figure 1-6. USB-6361 Layout



Answers from page 1-17.

Solution to Part I

The signal source is grounded, so you cannot choose RSE. The ideal choice is differential because you do not have more than eight signals to measure. If you have more than eight signals to measure, the ideal choice is NRSE. If you chose differential, wire the positive lead from voltage source 1 to pin 68 and wire the negative lead to pin 34. Wire the positive lead from voltage source 2 to pin 33 and wire the negative lead to pin 66. Finally, wire the positive lead from voltage source 3 to pin 65 and wire the negative lead to pin 31. If you chose NRSE, wire the positive leads in the same way, but wire the negative leads to AI SENSE.

Solution to Part II

The signal source is floating, so you could choose any of the three measurement systems. Both RSE and NRSE allows you to use 16 channels, but NRSE requires you to use bias resistors, so eliminate NRSE. Because you have fewer than eight signals to measure, the best choice for a good measurement is differential. However, with differential you would need bias resistors, so the simplest choice is RSE. Assuming you chose differential, wire the positive lead from battery 1 to pin 60 and wire the negative lead to pin 26. Wire the positive lead from battery 2 to pin 25 and wire the negative lead to pin 58. Finally, wire the positive lead from battery 3 to pin 57 and wire the negative lead to pin 23. You also need bias resistors from the positive and negative terminals of each battery to AI GND. If you chose RSE, wire the positive leads in the same way, but wire all the negative terminals to AI GND. RSE does not require bias resistors.

End of Exercise 1-3

Exercise 1-4 Using NI-DAQmx to Measure Voltage

Goal

Open and run an example program to take a software-timed, analog input measurement.

Scenario

You need to make sure a solar panel is functioning properly before deploying it to the field.

In this exercise, you use the specifications of a solar panel and the NI 9215 C Series module to verify a voltage reading. Instead of building a VI from scratch, you take advantage of an NI-DAQmx shipping example to take a software-timed measurement.

Implementation

Note Make sure that all measurement panels are stopped before completing this exercise. If necessary, reset the hardware in SystemDesigner.

1. Refer to *Light Sensor Information* in Appendix A, *NI CompactDAQ Measurements Demo Box Information* to answer questions about the wiring of this sensor. (Answers on page 1-22.)

   a. Is the source sensor floating or grounded? ________________________

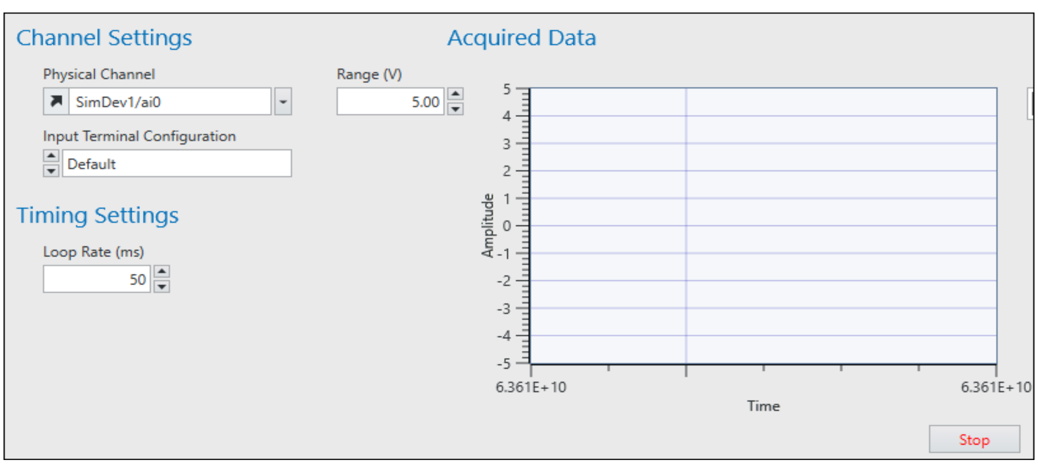
   b. What kind of terminal configuration is this wiring setup?

   c. Which device pins are wired to the solar panel?

2. On the Learning tab, select **Examples»Hardware Input and Output»NI-DAQmx» Programmatic Configuration»NI-DAQmx Analog Input.**

3. Enter `NI-DAQmx Analog Input` as the Project Name and click **Create**.

4. Open `Software-Timed Input with Voltage Probe.gvi`.



5. Update the panel so that the VI uses the NI 9215 to read a voltage from the solar panel.

Parameter	Hint	Value
Physical Channel	Which slot of the cDAQ chassis contains the NI 9215 module? What answer did you have for step 1c on page 1-20?	
Max Voltage, Min Voltage	According to the solar panel specification, what is the maximum voltage generated?	
Terminal Configuration	What answers did you have for step 1a and 1b on page 1-20?	

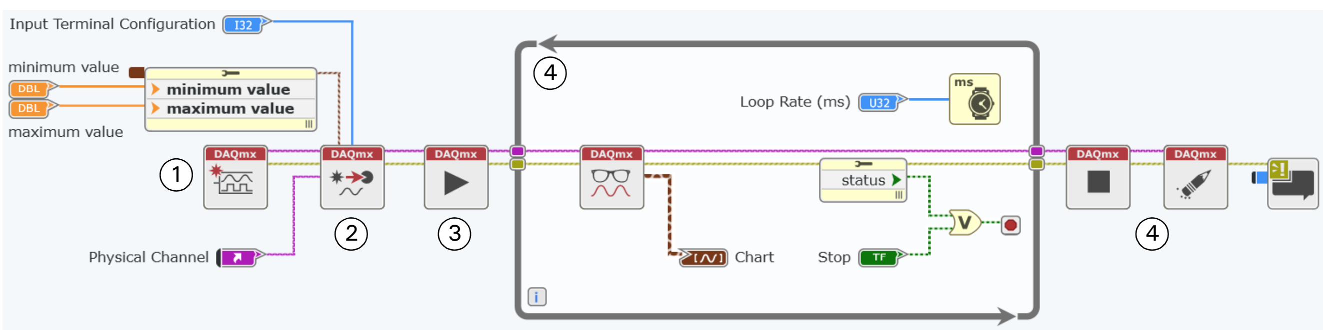
Challenge

Update the diagram and panel so that you can enter the precise minimum and maximum values for the solar panel.

6. Review the diagram to see how the VI is implemented.

   a. Right-click the **Physical Channel** control and select **Find»On diagram** from the shortcut menu.

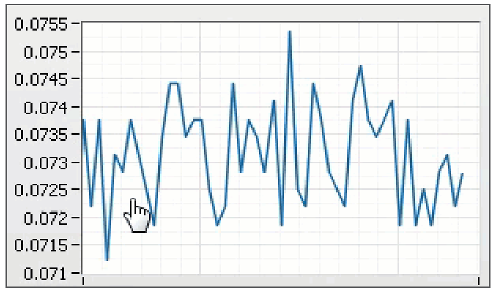
   b. Review the components of the diagram. For a software-timed acquisition application, you see the basic DAQ API structure: Create, Configure, Start, Read/Write, Stop/Clear.



1 **DAQmx Create Task**—Creates a task with the channels you specify. This node is optional.
2 **DAQmx Create Virtual Channel**—Using the Context Help for this node, you see that only the Physical Channel input is required. However, because we know the maximum and minimum values we can program them. The Terminal Configuration has a default value depending on the DAQ device, but it is important to verify that you are using the correct terminal configuration for the grounding situation you have between the sensor and the DAQ device. Because this is a simple acquisition, no further configuration is needed.
2 **DAQmx Start Task**—Starting the task means that the device is ready to take a measurement.
3 The While Loop contains the DAQmx Read node and controls the timing of the acquisition using the Wait (ms) node. This means that the timing of the acquisition is dependent on the OS.
4 **DAQmx Stop Task and DAQmx Clear Task**—This example VI uses both for completeness, but the DAQmx Clear Task node will stop the task if necessary.

Test

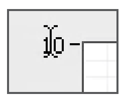
1. Run the VI. Because the y-axis on the Acquired Data graph is set to AutoScale, you see a very jagged graph.



2. Stop the VI and change the y-axis so that it does not autoscale and has a minimum value of `0` and a maximum value of `.5`.

   Tip Right-click the y-axis and select **Y Scale»Autoscale Y** to remove the checkmark and disable the autoscaling.

   Click a value on the axis and type in the desired value to specify the new scale. Editing the axis values can be tricky. Make sure nothing is selected and the cursor becomes the text tool, as shown below, before trying to click the value.



3. Run the VI again.

4. Place your hand over the light sensor to make sure the voltage drops to zero.

5. Use a flashlight or camera flash to see if you can get the voltage to reach its maximum, .5 V.

6. Stop and close the VI without saving it.

Answers from page 1-20.

1. Refer to *Light Sensor Information* in Appendix A, *NI CompactDAQ Measurements Demo Box Information* to answer questions about the wiring of this sensor.

   a. Is the source sensor floating or grounded? **Floating**

   b. What kind of terminal configuration is this wiring setup? **Differential**

   c. Which device pins are wired to the solar panel? **0 and 1**

End of Exercise 1-4

2 Generating an Analog Signal (Analog Output)

Topics

+ Resources

Exercises

Exercise 2-1 Using NI-DAQmx to Generate a Voltage

Resources

Resources Refer to the following resource for more information about topics in this lesson. For locations with info codes, go to `ni.com/info` and enter the code to access the article.

Article	Location
M-series: External AO Reference	Info code: `mseriesextref`
Outputting a Voltage Using an External Voltage Reference	Info code: `aoextref`
How Can I Maximize the Resolution of My Analog Output Signal?	Info code: `aoresolution`

Signal Conditioning: Isolation and Amplification

Article	Location
Isolation Technologies for Reliable Industrial Measurements	Info code: `isolationtech`

Output Terminal Configuration

Article	Location
Default Input/Output Terminal Configurations	Info code: `defaultioterm`
Field Wiring and Noise Considerations for Analog Signals	Info code: `fieldwiring`

Reading the Pinout

Article	Location
Connect Analog Output Signals to a DAQ Device	Info code: `connectao`

Data Transfer Mechanism for an Output Operation

Article	Location
Data Transfer Mechanisms	Info code: `datatransfermech`

Exercise 2-1 Using NI-DAQmx to Generate a Voltage

Goal

Open and run an example program to generate a software-timed, analog output signal.

Scenario

In this exercise, you use the NI 9263 C Series module to generate a tone through the speaker in the demo box. Instead of building a VI from scratch, you take advantage of a NI-DAQmx shipping example to make an on-demand, software-timed measurement.

Implementation

Note Make sure that all measurement panels are stopped before completing this exercise.

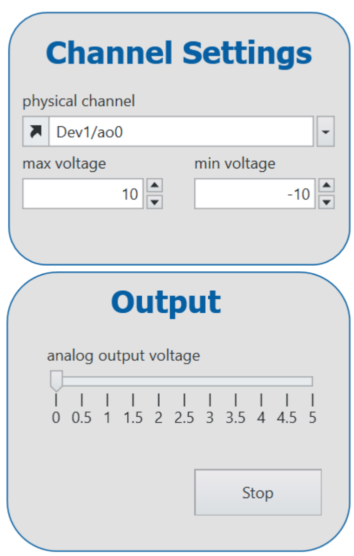
1. Refer to Appendix A, section E, *Speaker Information*, and section F, *Sound and Vibration Signal Simulator Box Information*, to answer questions about the wiring of the NI 9263 module to components in the Demo Box.

   a. Which device pins are wired to the speaker?________________________________

   b. What device pins connect the NI 9263 to the fan control of the Sound and Vibration Signal Simulator?____

2. Open `<Exercises>\DAQNXG\Analog Output\Fan Control.lvproject`.

Note This VI is based on the Example program `Voltage - On Demand Output` located in the **NI-DAQmx»Analog Output** folder.



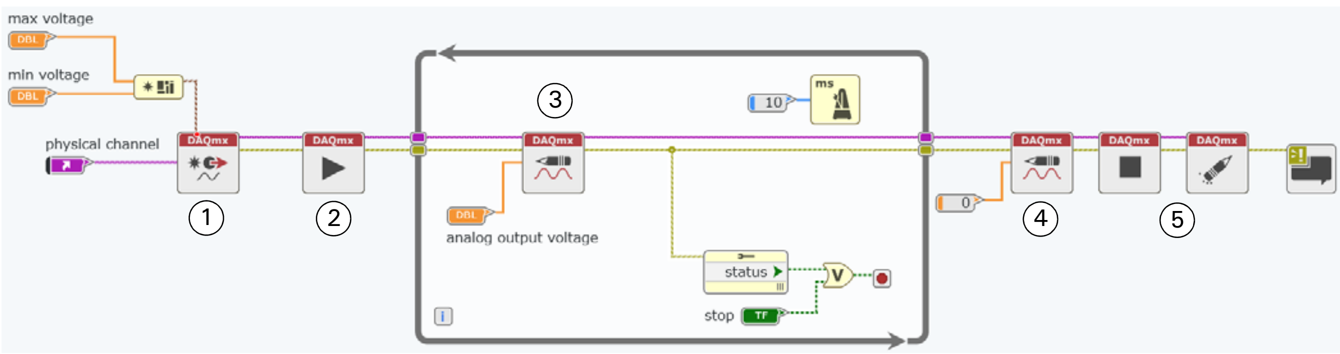
3. Update the panel so that the VI uses the NI 9263 to generate a voltage.

Parameter	Hint	Value
Physical Channel	Which slot of the cDAQ chassis contains the NI 9263 module? What answer did you have for step 1b on page 2-4?	

4. Review the diagram to see how the VI is implemented.

   a. Right-click the **Physical Channel(s)** control and select **Find»On diagram** from the shortcut menu.

   b. Review the components of the diagram. For a software-timed acquisition application, you see the basic DAQ API structure: Create, Configure, Start, Read/Write, Stop/Clear.



1	**DAQmx Create Virtual Channel**—Using the Context Help for this node, you see that only the Physical Channels input is required. However, because we know the maximum and minimum values we can program them. The **output terminal configuration** has a default value depending on the DAQ device, but it is important to verify that you are using the correct terminal configuration for the grounding situation you have between the actuator and the DAQ device. Because this is a simple voltage generation, no further configuration is needed.
2	**DAQmx Start Task**—Starting the task means that the device is ready to generate a signal. Because this task is software-timed, you do not need to write data to the buffer before starting the task.
3	**DAQmx Write**—The While Loop contains the NI-DAQmx Write node and controls the timing of the generation of a signal using the Wait Until Next ms Multiple function. This means that the timing of the signal generation is dependent on the OS.
4	Turning off the fan—By writing 0 voltage to the fan control you make sure the fan isn't still running after you stop the VI.
5	**DAQmx Stop Task** and **DAQmx Clear Task**—This example VI uses both for completeness, but the NI-DAQmx Clear Task node will stop the task if necessary.

Test

1. Run the VI.
2. Change the **Analog Output Voltage (V)** slider on the panel of the VI to change the speed of the fan.
3. Stop the VI.
4. Change the **Max Voltage** to a value less than 5 and run the VI again.
   - ☐ What happens when you specify an **Analog Output Voltage** greater than the **Max Voltage**?____

   Note If the fan continues to run after you stop the VI, you can go to SystemDesigner and reset the hardware.

5. Stop and close the VI without saving it.

End of Exercise 2-1

3 Generating or Reading a Digital Signal

Topics

+ Resources

Exercises

Exercise 3-1 Using NI-DAQmx to Generate a Digital Signal

Resources

Resources Refer to the following resource for more information about topics in this lesson. For locations with info codes, go to `ni.com/info` and enter the code to access the article.

Digital I/O

Article	Location
Choosing the Right Industrial Digital I/O Module for Your Digital Output Sensor	Info code: `industrialdio`

Article	Location
Digital I/O Sinking and Sourcing	Info code: `diosinksource`

Isolation

Article	Location
Isolation Types and Considerations when Taking a Measurement	Info code: `isolationtype`
Common Hardware Connections for Digital I/O and Counter/Timer Boards	Info code: `dioconnect`

Reducing Measurement Noise in Digital Signals

Article	Location
Five Tips to Reduce Measurement Noise	Info code: `noisetips`
High-Speed Digital I/O Logic Families	Info code: `diologicfam`

3 Types of Digital Input Configurations

Article	Location
Connecting Digital Input Signals to a DAQ Device	Info code: `connectdio`

4 Types of Digital Output Configurations

Article	Location
Connecting Digital Output Signals to a DAQ Device	Info code: `connectdout`

Exercise 3-1 Using NI-DAQmx to Generate a Digital Signal

Goal

Open and run an example program to generate a digital signal.

Scenario

In this exercise, you use the specifications for the LEDs and the NI 9472 C Series module to generate the status of a digital line. Instead of building a VI from scratch, you take advantage of a NI-DAQmx shipping example to take a software-timed measurement.

Implementation

Note Make sure that all digital tasks are stopped before completing this exercise.

1. Refer to *LEDs Information* in Appendix A, *NI CompactDAQ Measurements Demo Box Information* to answer questions about the wiring of the LEDs.

   a. Which lines from the 9472 are wired to the red LEDs?______________________

   b. Which lines are wired to the green LEDs?________________________________

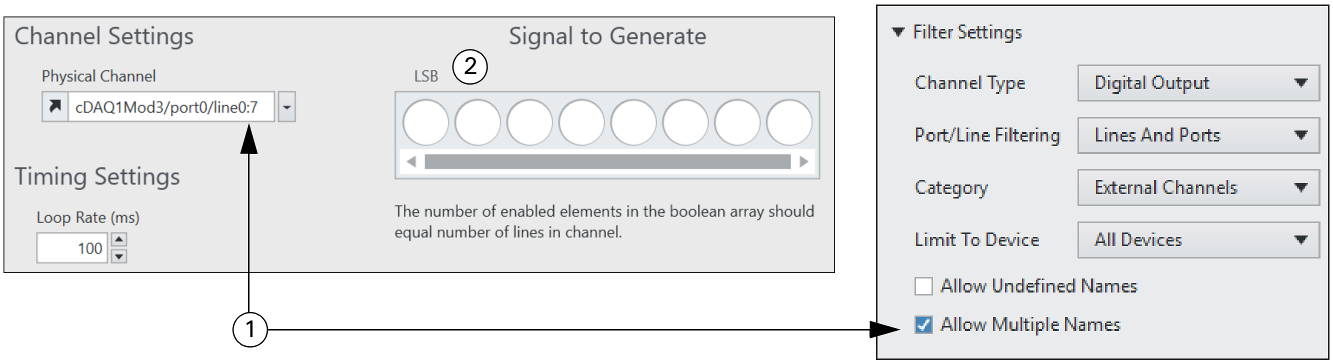
   c. Are the LEDs a sinking or sourcing device? ______________________________

2. On the Learning tab, select **Examples»Hardware Input and Output»NI-DAQmx»Programmatic Configuration»NI-DAQmx Digital Output.**

3. Enter `NI-DAQmx Digital Output` as the Project Name and click **Create**.

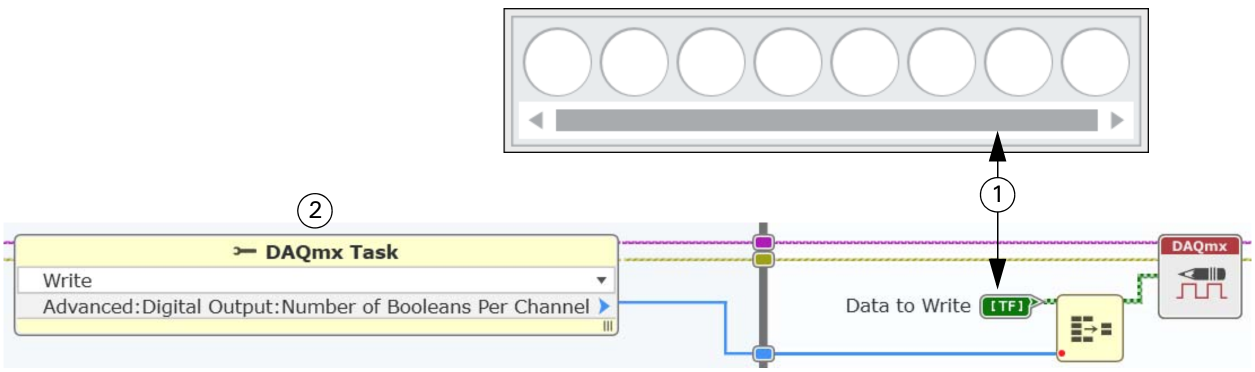
4. Open `On Demand Digital Output.gvi`.

5. Select all eight digital lines in the **Physical Channel** control. To select multiple lines, you must place a checkmark in the **Allow Multiple Names** option on the Item tab.



1 To select multiple lines you must check **Allow Multiple Names** in the Item tab. Then using the syntax *device*/port*x*/line0:7 you can specify every line in the port. On the other hand, *device*/port*x*/line0:1 only writes to the first two lines.
2 The number of lines in the indicator control must match the number of lines specified in **Physical Channel**.

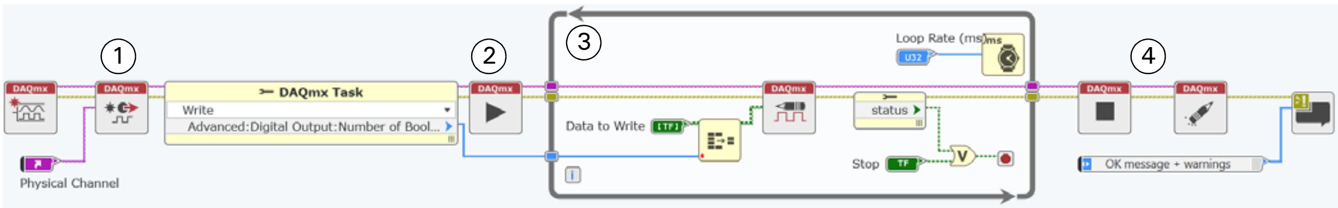
6. Right-click the Data to Write control and select **Find on panel.**



1 Notice that the control is actually a one-dimensional array of Boolean elements.
2 The NI-DAQmx Write instance is configured to accept data that is a one-dimensional array of data. Using the property node, you can determine the size of the array from the channel.

7. Review the rest of the diagram to see how the VI is implemented.

   For a software-timed acquisition application, you see the basic DAQ API structure: Create, Configure, Start, Read/Write, Stop/Clear.



1	**DAQmx Create Virtual Channel**—Using the Context Help for this node, you see that only the Physical Channel input is required. The Terminal Configuration has a default value depending on the DAQ device, but it is important to verify that you are using the correct terminal configuration for the grounding situation you have between the sensor and the DAQ device. Because this is a simple acquisition, no further configuration is needed.
2	**DAQmx Start Task**—Starting the task means that the device is ready to take a measurement.
3	The While Loop contains the NI-DAQmx Write node and controls the timing of the acquisition using the Wait (ms) function. This means that the timing of the generation of the signal is dependent on the OS.
4	**DAQmx Stop Task** and **DAQmx Clear Task**—This example VI uses both for completeness, but the DAQmx Clear Task node will stop the task if necessary.

Test

1. Run the VI.
2. Update the panel so that the VI uses the NI 9472 to generate a signal that turns on one green, one amber, and one red LED.

Parameter	Hint	Value
Physical Channel	Which slot of the cDAQ chassis contains the 9472 module? How many lines does the port on the 9472 contain?	
Data to Write	The demo box has eight LEDs, one for each line in the port. What answers did you have for step 1a and 1b on page 3-4?	

3. Update the panel so that the VI only lights green and amber LEDs.
4. Stop the VI and change the **Physical Channel** control such that you specify only five lines. What happens when you run the VI and set more than 5 lines to High?

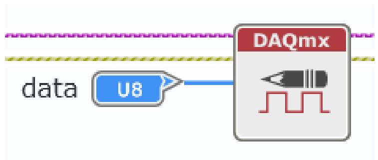
____

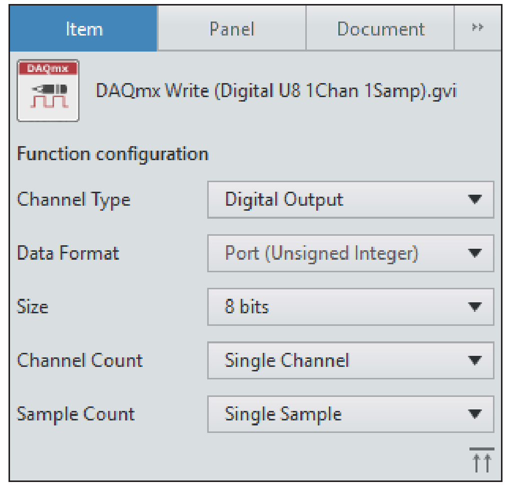
   The array of data input into the NI-DAQmx Write VI must match the number of lines you specify.

5. Stop and close the VI without saving it.

Challenge

Change the instance of the NI-DAQmx Write VI to a U8 (port format). Delete the broken wires and create a new control, **data**.





Run the VI and enter different values.

☐ Can you guess which LEDs light up before you enter the new value? For example, if you enter the value 8, which LED lights up? Why?

☐ What happens when you enter 600 into the **data** control?

Stop and close the VI without saving it.

End of Exercise 3-1

4 Choose a Signal to Explore

In this lesson you choose a signal you would like to explore more closely.

Topics

+ List of Signals with Media Guides

A. List of Signals with Media Guides

The following lessons have instructional media. Select a topic and complete the appropriate media module located in the `<Exercises>\DAQNXG\Media\` folder. Refer to the appropriate appendix for more information and exercise instructions.

- Measuring Temperature—Appendix B
- Measuring Sound, Vibration, and Acceleration (IEPE Measurements)—Appendix C
- Measuring Strain, Force, or Pressure (Bridge-Based Measurements)—Appendix D
- Measuring Position with Encoders (Counter Input)—Appendix E
- Measuring Edges, Frequency, Pulse Width, and Duty Cycle—Appendix F

5 Programming with the NI-DAQmx API

Topics

+ Resources

Exercises

Resources

Resources Refer to the following resource for more information about topics in this lesson. For locations with info codes, go to `ni.com/info` and enter the code to access the article.

Article	Location
Creating a Virtual Channel in NI-DAQmx and using it in LabVIEW	Info code: `createchannellv`
Getting Started with NI-DAQmx: Main Page	Info code: `daqmxgetstart`

What types of task configuration are possible?

Article	Location
Getting Started with NI-DAQmx: Basic Programming with NI-DAQmx	Info code: `basicdaqmx`
Getting Started with NI-DAQmx: Getting Started with NI-DAQmx Programming in LabVIEW	Info code: `basicdaqmprog`

10 Functions to Handle 80% of Your Data Acquisition Applications

Article	Location
Learn 10 Functions in NI-DAQmx and Handle 80 Percent of Your Data Acquisition Applications	Info code: `10functions`

Starting and Stopping Tasks

Article	Location
When to Use the DAQmx Start Task and DAQmx Stop Task VIs	Info code: `daqmxstartstop`

Stopping vs. Clearing Tasks

Article	Location
What is the Difference between the DAQmx Stop Task VI and the DAQmx Clear Task VI?	Info code: `diffstopclear`

What if I want even MORE control over a task?

Article	Location
Getting Started with DAQmx: Basics of DAQmx Property Nodes	Info code: `basicdaqmxprop`

Exercise 5-1 Using Digital Lines to Control Multiple LEDs

Goal

Use a digital output device to write a finite amount of digital data across a digital port.

Scenario

You must develop an application that controls the power state of eight devices. Each device can be turned on or off by connecting a digital line to its POWER input.

Design

To complete this task, you will write digital data to lines 0-7 of port 0 of the NI 9472 module in your cDAQ-9178 chassis. You do not yet have samples of the eight devices that your VI needs to control. Therefore, the values of these lines are wired to the LEDs on the NI CompactDAQ Measurements Demo Box to ensure that you are setting the line values correctly.

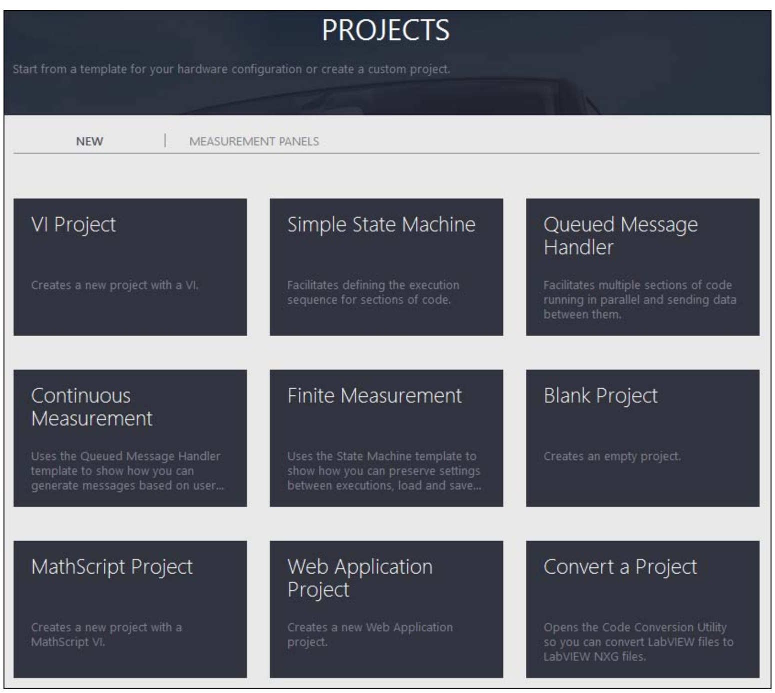
When this application runs, each digital line will be set to the on or off state. The application will write one value to each line that is included in as part of the digital channel.

Because all eight digital lines are part of the same port on the same device, you can create and use a single channel to handle writing data to all eight lines at once.

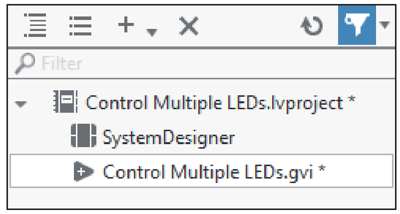
Implementation

Develop a project and VI that will use one channel to write digital data to multiple digital output lines.

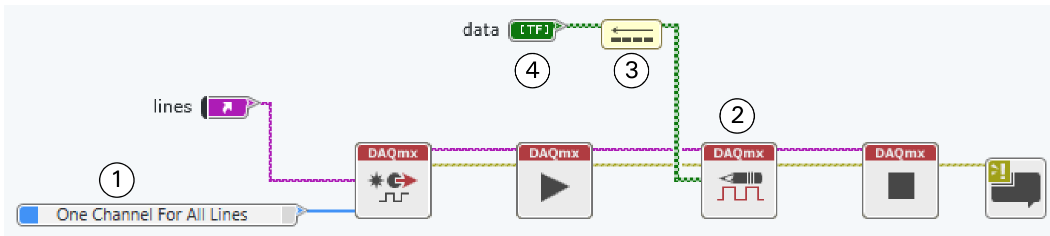
1. On the **Projects** tab, click **New»VI Project** and create a project named `Control Multiple LEDs`.



2. Change the name of the function to `Control Multiple LEDS.gvi`.

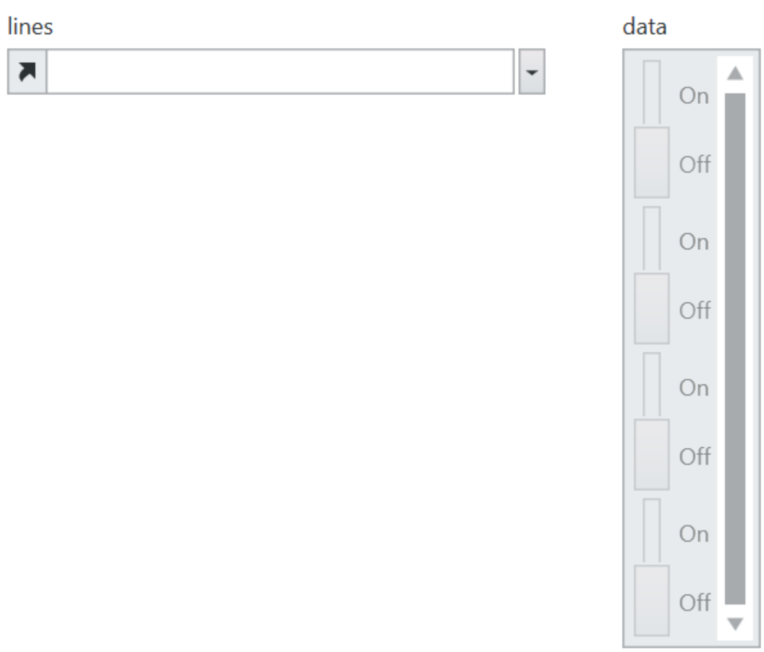


3. Create the diagram to perform a generation of a finite number of samples.

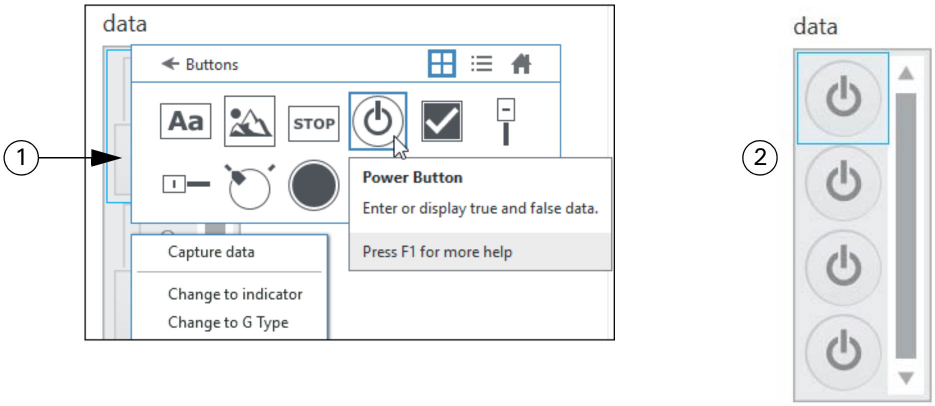


1	**Line grouping constant**—This constant configures the virtual channel to contain all of the digital lines specified in the DO Lines control.
2	**DAQmx Write**—Configure this node to write a single sample of a 1D digital data array to a single channel.
3	**Reverse 1D Array**—Rearranges the values in the Boolean array so that the orientation shown on the panel matches the orientation of the LEDs on the NI CompactDAQ Measurements Demo Box.
4	**1D array of Boolean control**—To create this 1D array, place a **True Constant** on the diagram. Right-click the constant and select **Change to Control**. Right-click the control and select **Change to Array**.

4. Switch to the panel of the VI and move the controls from the **Unplaced Items** tray to the panel.

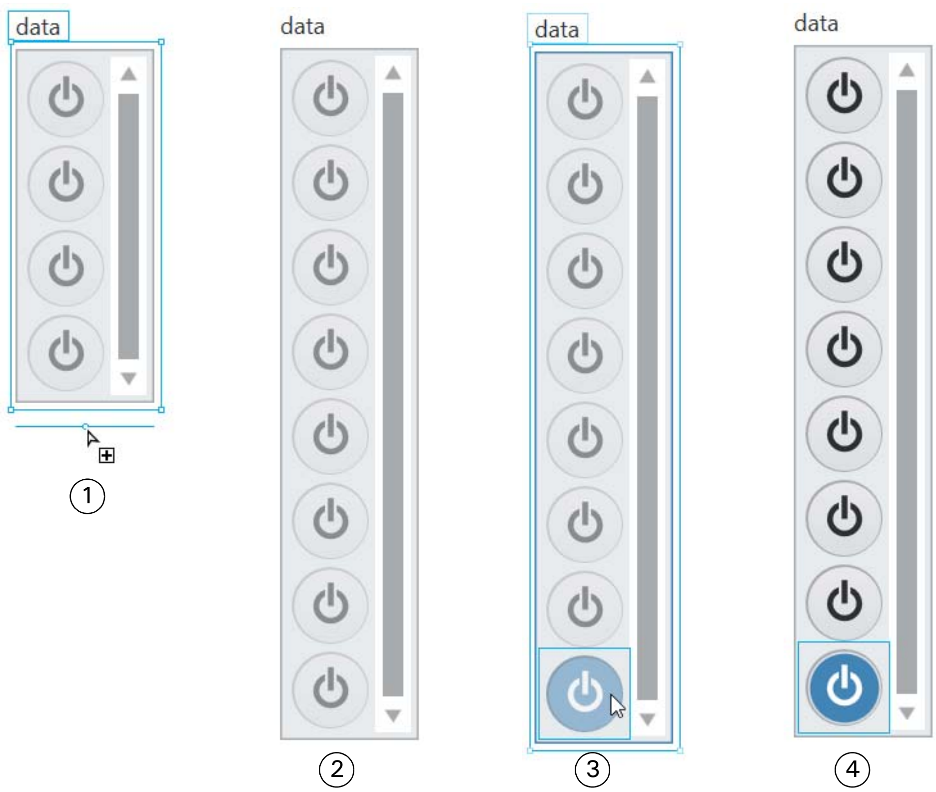


5. Switch to the panel of the VI and move the controls from the Unplaced Items tray to the panel.



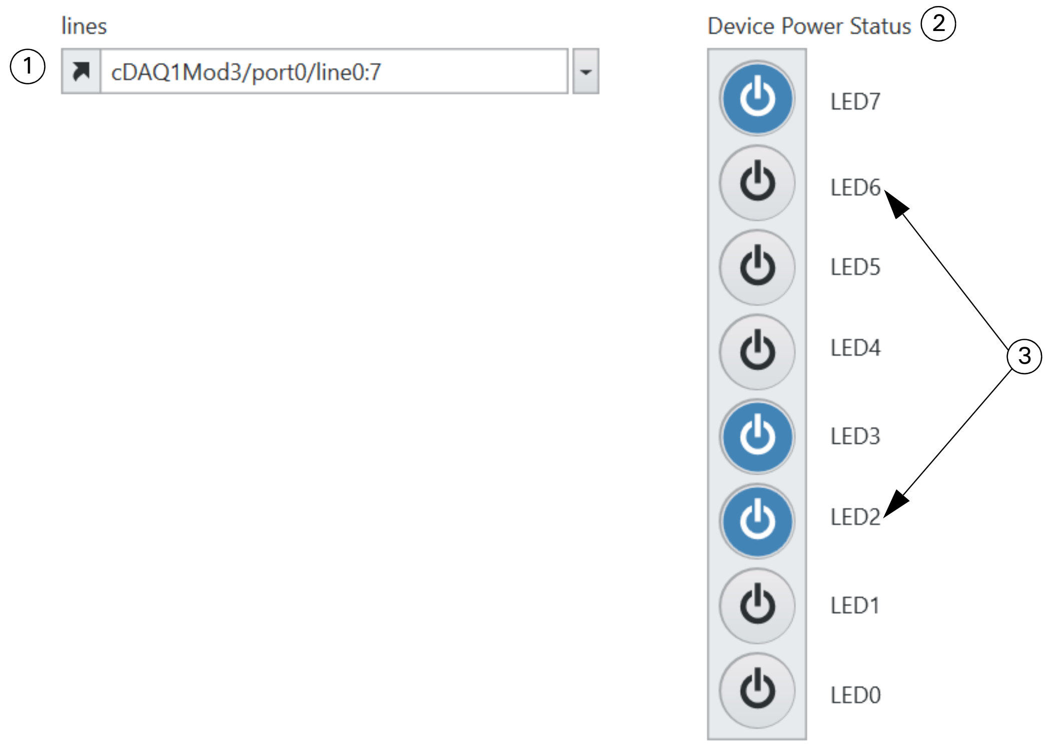
1	Right-click a control and select the Power Button from the menu.
2	All the Boolean controls in the array update when you change the look of just one.

6. Expand the **data** control and enable eight items.



1	Hover over the control until the expansion handle appears
2	Drag the expansion handle to reveal eight controls.
3	Click the eighth control to enable it.
4	Once the eighth control is enabled, the first seven controls are enabled.

7. Arrange the items on the panel and set the values.



1	**lines**—Because you are using one channel to write to multiple digital lines on the same device, you can use the format `cDAQ1Mod3/port0/line0:7` to write data to lines zero to seven of port0.
2	**Device Power Status**—Update the label and remove the vertical scrollbar from the visible parts.
3	**Labels**—Add labels to make it clear which digital line is controlling each device.

8. Save the VI and the project.

Test

Ensure that the VI is able to write digital data to LEDs zero to seven.

1. Ensure that you are able to turn on all eight LEDs.

   ☐ Set the Device Power Status for each device to ON.

   ☐ Run the VI. Each LED on the NI CompactDAQ Measurements Demo Box should turn on.

2. Ensure that you are able to selectively turn on some LEDs and turn others off.

   ☐ Set the Device Power Status for several devices to OFF.

   ☐ Run the VI. The LEDs on the NI CompactDAQ Measurements Demo Box should match the values that you specified for Device Power Status.

3. Ensure that you are able to turn off all eight LEDs.

   ☐ Set the Device Power Status for each device to OFF.

   ☐ Run the VI. Each LED on the NI CompactDAQ Measurements Demo Box should turn off.

Note It is important to run this step to turn off all eight LEDs, otherwise they will remain on.

End of Exercise 5-1

Exercise 5-2 Acquire Finite Data from a Light Sensor

Goal

Acquire data using finite buffered configuration and analyze this data for maximum and minimum values.

Scenario

Your team is developing a camera that uses a light sensor to automatically determine whether or not a flash is needed when taking photographs. You are responsible for developing code that acquires a fixed number of samples under different lighting conditions and then measures the minimum, maximum, and DC voltage levels that the light sensor generates.

Design

To complete this task, you will read data from Channel 0 of the NI 9215 module in your cDAQ-9178 chassis.

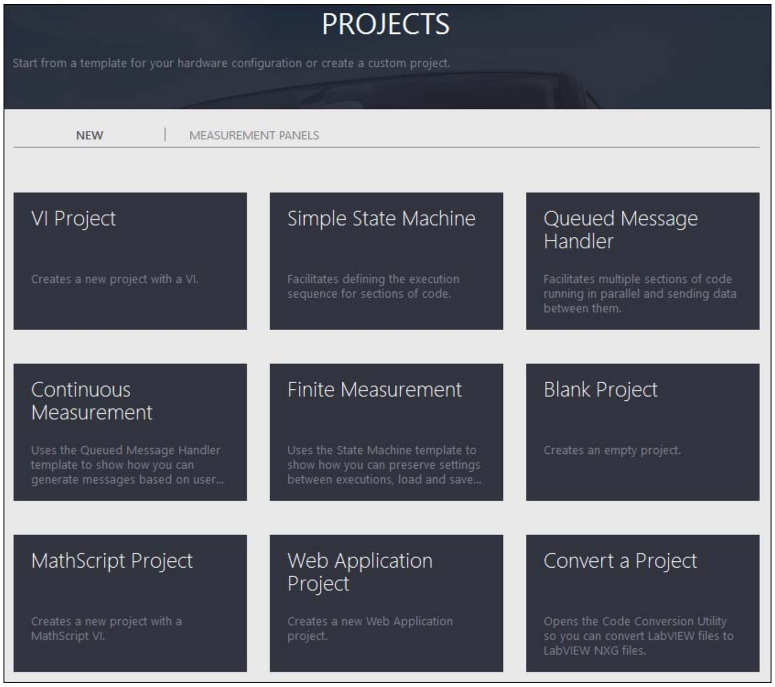
You want to acquire 1000 points of data to ensure that your maximum, minimum and DC measurements are statistically valid. You want for the acquisition to occur over the course of 2 seconds.

What sample rate should you use for this acquisition? ____________________________
(Answer on page 5-14.)

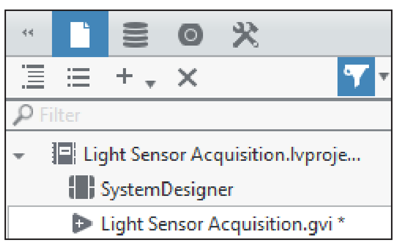
Implementation: Finite Buffered Acquisition

First, you will create a VI to ensure that you are able to acquire light sensor data and display it in a VI.

1. On the **Projects** tab, click **New»VI Project** and create a project named `Light Sensor Acquisition`.

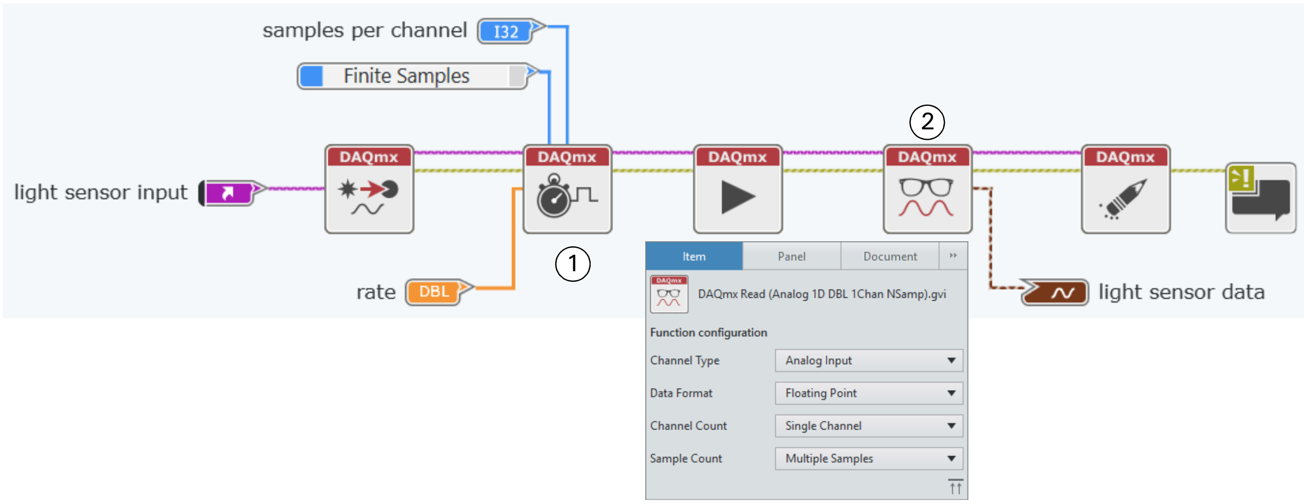


2. Change the name of the function to `Light Sensor Acquisition.gvi`.



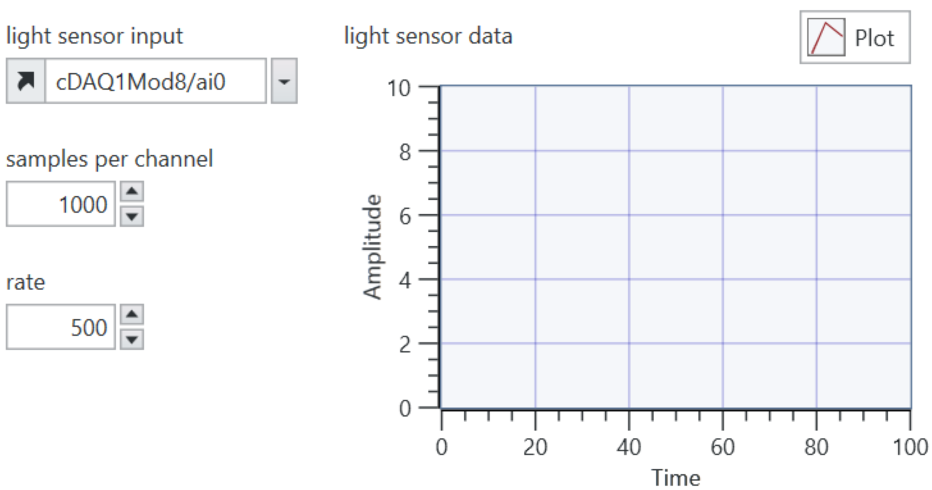
3. Select **File»Save All** and save the project in the `<Exercises>\DAQNXG\Finite Input\` directory.

4. Create the diagram to perform a buffered acquisition of a finite number of samples.



1 **DAQmx Timing**—This node configures the number of samples to acquire or generate and creates a buffer.
2 **DAQmx Read**—Make sure this node is configured as analog input, waveform, single channel, multiple samples.

5. Arrange the items on the panel and set the values.



6. Save the VI.

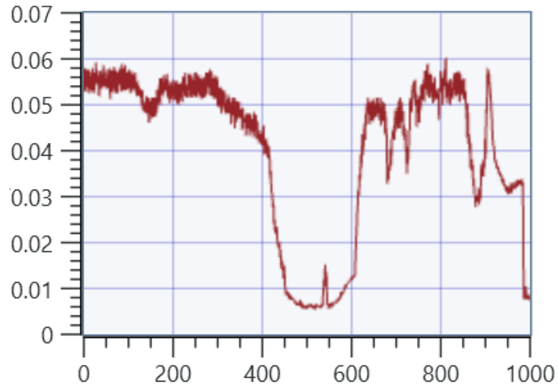
Test

Ensure that the VI is able to acquire and display light sensor data from your NI 9215.

1. Run the VI.

   ☐ While the VI is running, place your hand over the light sensor to ensure that the sensor detects the change in light level.

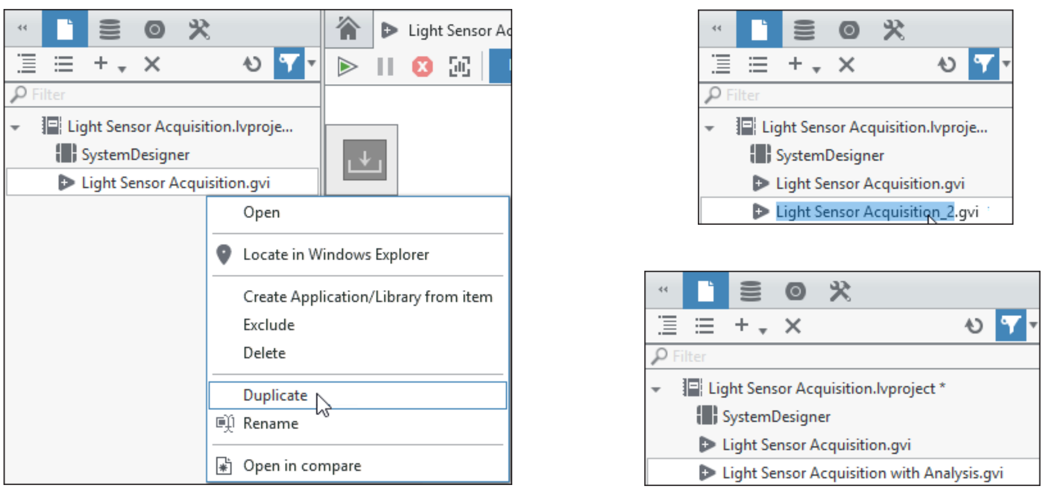
☐ Verify that the VI takes approximately two seconds to run.



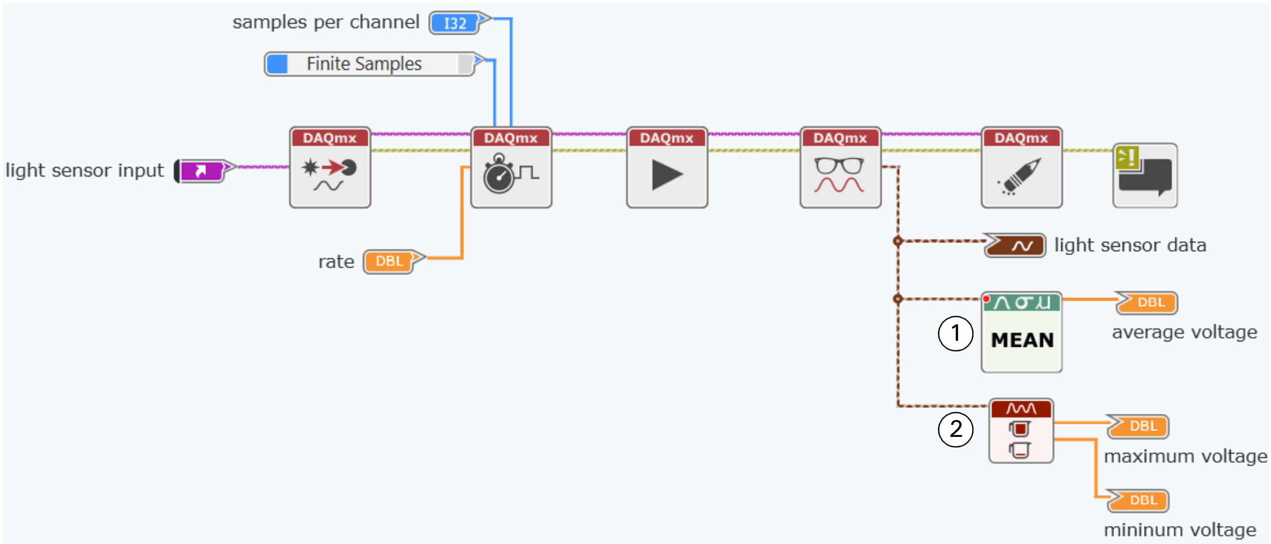
Implementation: Light Sensor Data Analysis

After ensuring that your VI is working properly, it is time to analyze the data that you acquire. In order to develop the code that controls the flash functionality, you need to characterize the voltage values acquired at different light levels. You have determined that you need to calculate the minimum, maximum, and average voltage values at each light level.

1. In the Files pane, duplicate the Light Sensor Acquisition VI, rename it `Light Sensor Acquisition with Analysis.gvi` and save it.



2. Add waveform analysis to the finite acquisition by analyzing the data after it has been acquired, as shown below.



1	**Measures of Mean**—This node calculates the average voltage value for the waveform read from the light sensor. Configure this node as **Mean**.
2	**Waveform Min Max**—This node determines the maximum and minimum values in the light sensor signal.

3. Save the VI.

Test

1. From the panel, run the VI and observe the resulting values.

   - ☐ What is the average voltage value? _____________________
   - ☐ What is the maximum voltage value? _____________________
   - ☐ What is the minimum voltage value? _____________________

2. Place your hand over the light sensor, run the VI and observe the resulting values.

   - ☐ What is the average voltage value? _____________________
   - ☐ What is the maximum voltage value? _____________________
   - ☐ What is the minimum voltage value? _____________________

3. Save and close the VI.

Answers from page 5-10.

What sample rate should you use for this acquisition?
To acquire 1000 samples in 2 seconds, you use this equation. 1000 / 2 = 500

End of Exercise 5-2

Exercise 5-3 Continuously Monitoring Fan Status

Goal

Continuously monitor an analog input channel.

Scenario

You must develop an application that can read acceleration data for the X and Y axes of a fan and determine whether it has become unbalanced as a result of a broken fan blade.

The first step in developing this type of application is to first ensure that you are able to continuously read data from accelerometers. You can then compare the results from a known good fan with those from an unbalanced fan to determine the test limits that you could use for a pass/fail test.

Design

To complete this task you will read data from Channels 0 and 1 of the NI 9234 module in your cDAQ-9178 chassis. These channels are connected to accelerometers that generate X and Y acceleration data for the balanced and unbalanced fans within the Sound and Vibration Signal Simulator.

As the fan speed increases, the differences in the acceleration data for the balanced and unbalanced fans will become more pronounced.

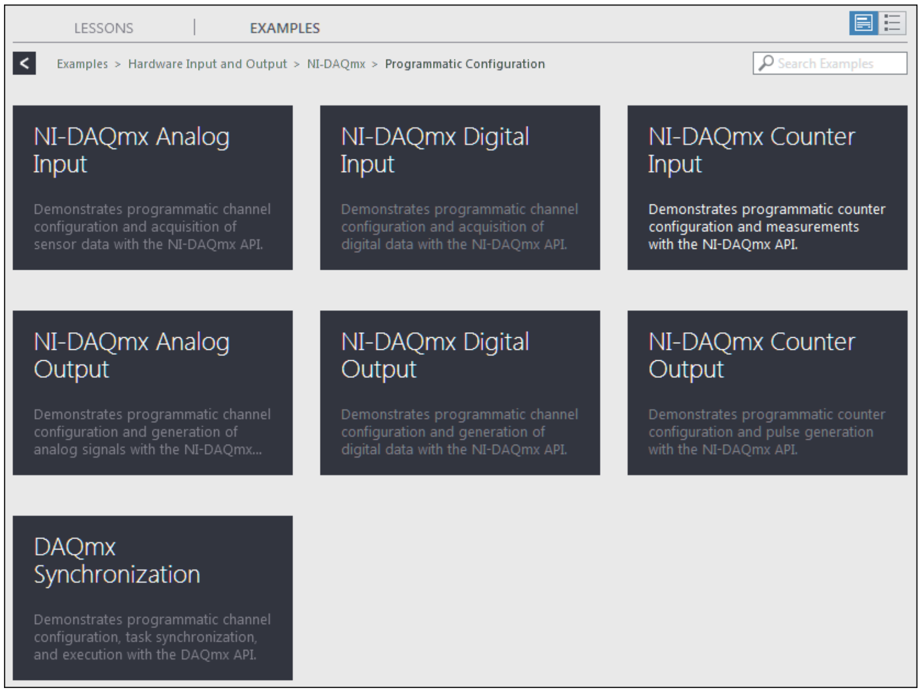
The Sound and Vibration Signal Simulator has two switches that you will use in this exercise:

- ☐ The Balanced Fan / Unbalanced Fan switch determines which fan is active and generating acceleration data.
- ☐ The BNC/DIAL switch determines whether the knob or BNC input terminal controls the speed of the selected fan.

Implementation

1. On the **Learning** tab, navigate to **Examples»Hardware Input and Output» NI-DAQmx»Programmatic Configuration**.

2. Click **NI-DAQmx Analog Input** and create a project named `Read Fan Acceleration`.

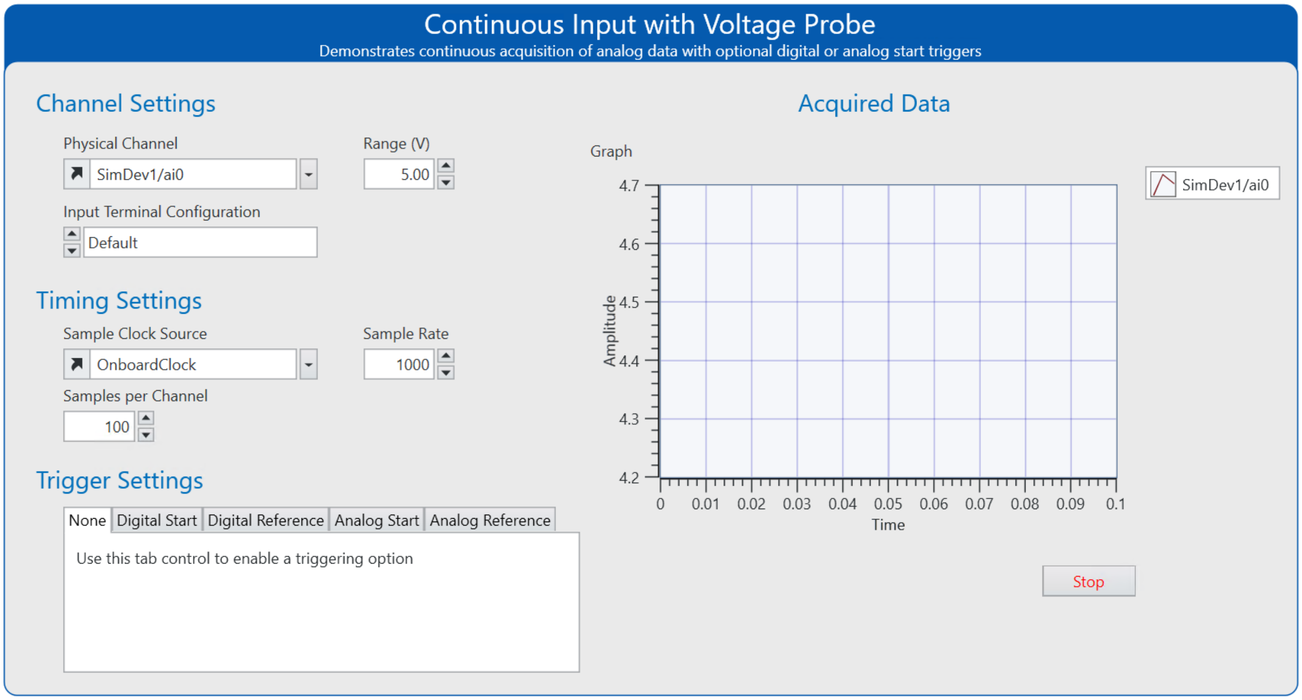


3. Duplicate **Continuous Input with Voltage Probe.gvi** so that you can modify the without affecting the source files for the example program.

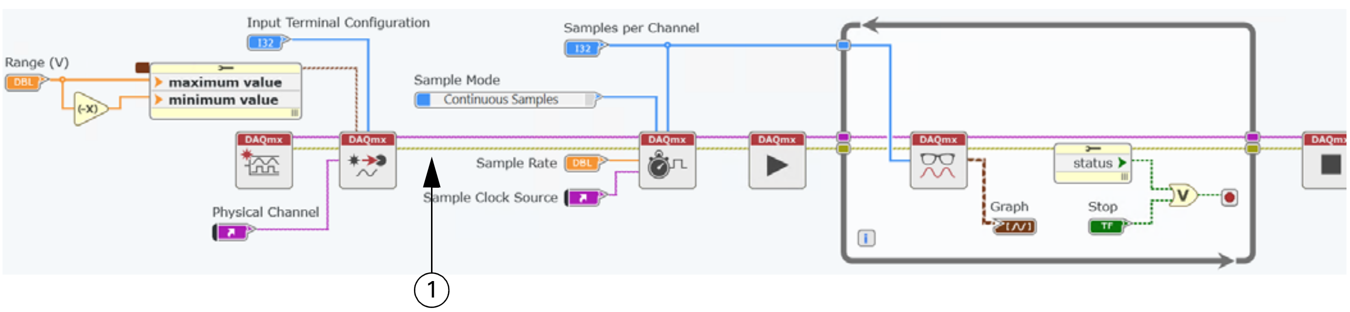
   - ☐ Right-click **Continuous Input with Voltage Probe.gvi** and select Duplicate.

   - ☐ Rename the VI in the Files pane as `Read Fan Acceleration.gvi`.

4. Open **Read Fan Acceleration.gvi.**

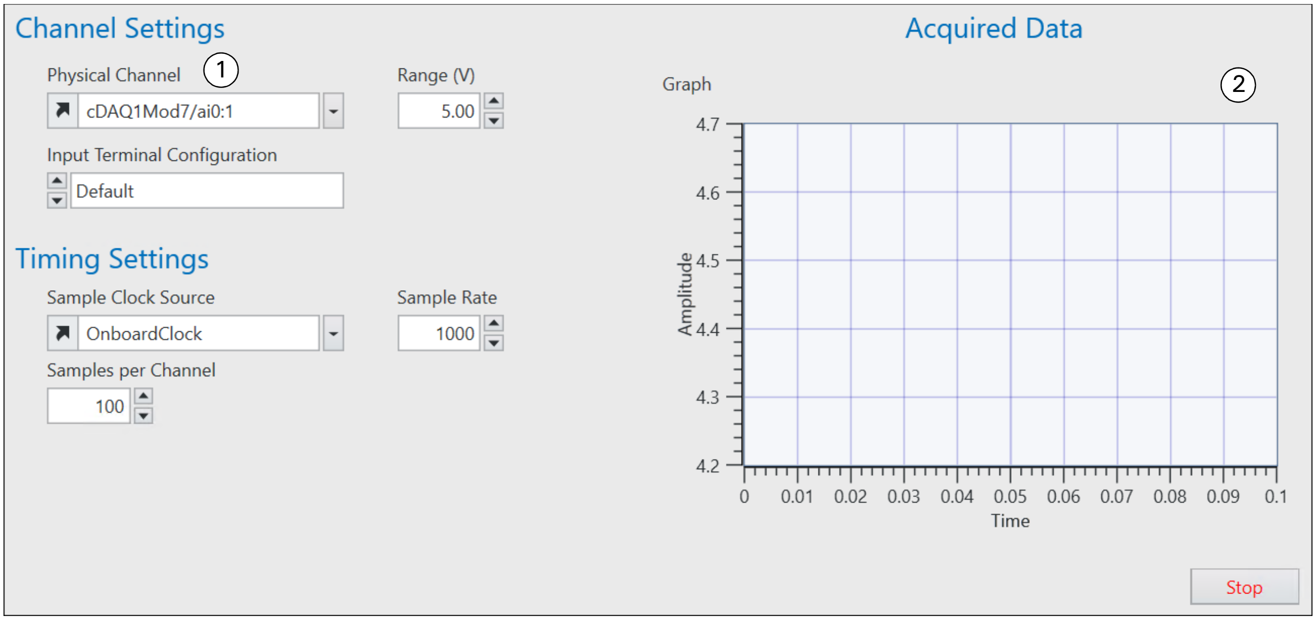


5. Modify the diagram of the VI.



1 Delete the Trigger Settings section and wire the task and error wires from the DAQmx Create Virtual Channel VI to the DAQmx Timing VI.

6. Update the panel.



1 Change the Physical Channel input to read from ai0 and ai1 of the NI 9234 (cDAQ1Mod7).
2 Show the plot legend for the waveform graph. Once you run the VI, the labels will map to the physical channels.

7. Save the VI.

Test

Generate data for both the balanced fan and the unbalanced fan. Note the difference in the data generated when the fan is at maximum speed.

1. Configure the Sound and Vibration Signal Simulator to begin testing the balanced fan.

   ☐ Ensure that the Fan Speed Control switch is set to **Dial**.

   ☐ Ensure that the switch between the fans is set to **Balanced Fan**.

   ☐ Ensure that the Fan Speed Control dial is turned all the way counterclockwise.

2. Run the VI.

3. Turn the Fan Speed Control knob clockwise and observe the impact on the X and Y acceleration plots.

   ☐ Note that the magnitude of the X and Y acceleration did not change significantly.

4. Configure the Sound and Vibration Signal Simulator to begin testing the unbalanced fan.

   ☐ Turn the Fan Speed Control dial all the way counterclockwise.

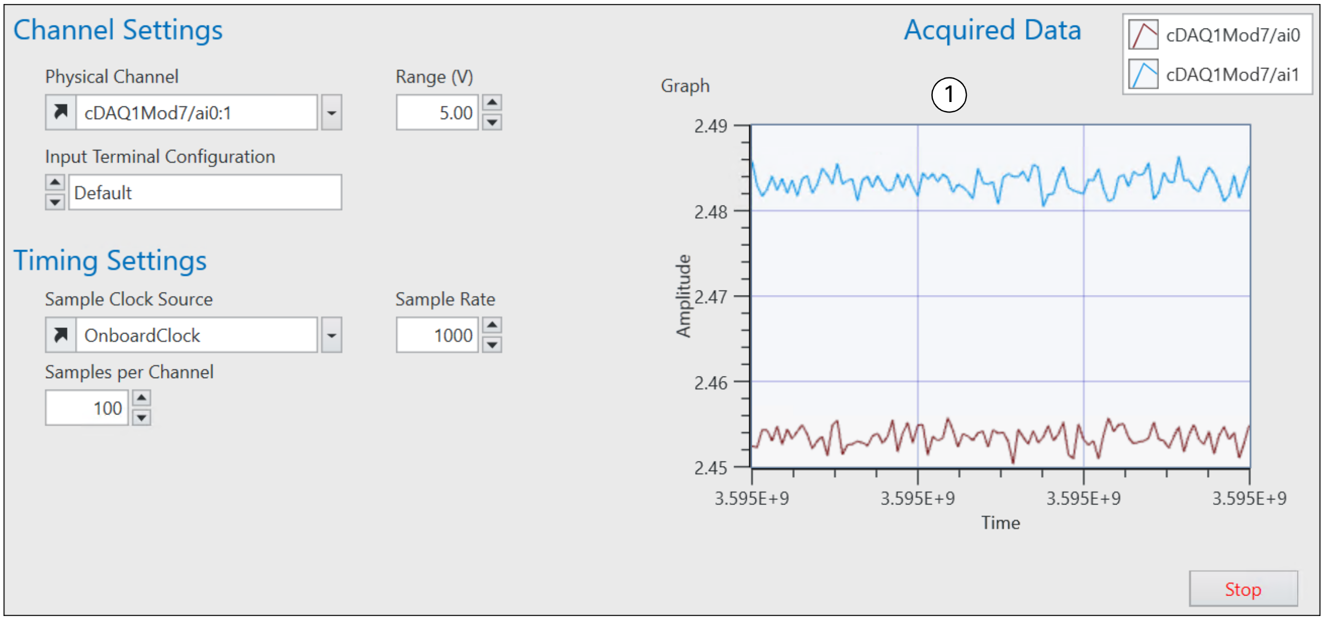
   ☐ Set the switch between the fans to **Unbalanced Fan**.

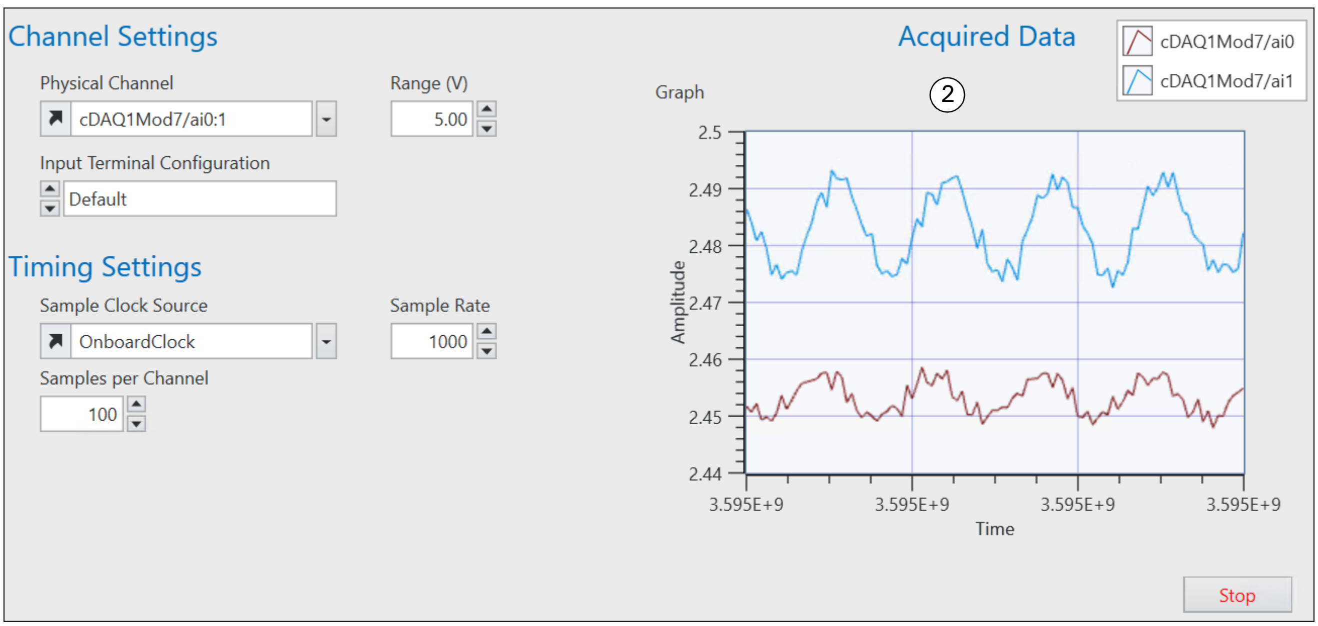
5. Slowly turn the Fans Speed Control knob clockwise and observe the impact on the X and Y acceleration plots.

   ☐ What shape do the waveforms take on? ________________________________

   ☐ What is the approximate amplitude of the resulting waveforms?

   - ai0 (X-Acceleration): _____________________________
   - ai1 (Y-Acceleration): _____________________________





1 Balanced Fan	2 Unbalanced Fan

End of Exercise 5-3

Exercise 5-4 Continuously Controlling Fan Speed

Goal

Continuously write analog data to a device.

Scenario

In the previous exercise, you developed an application that can be used to determine whether or not a fan is unbalanced. Now you must develop an application that can control the speed of a balanced fan.

Design

To complete this task, you will write analog data to channel 1of the NI 9263 analog output module in your cDAQ-9178 chassis.

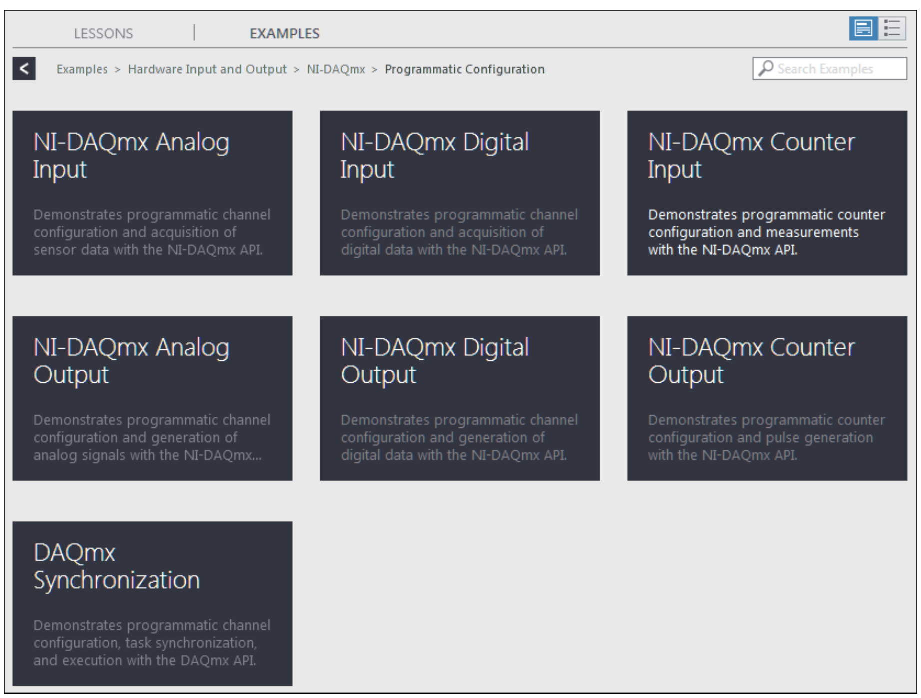
This channel is wired to the Fan Speed Control BNC on the Sound and Vibration Signal Simulator.

For this application, we can use software timing to write new fan speed values to the virtual channel when they become available. Hardware timing requires that you create a buffer of values. If a number of values reside in the buffer, that could result in a delay when you change the fan speed value on your panel.

Implementation

1. On the **Learning** tab, navigate to **Examples»Hardware Input and Output» NI-DAQmx»Programmatic Configuration**.

2. Click **NI-DAQmx Analog Output** and create a project named `Fan Speed Control`.

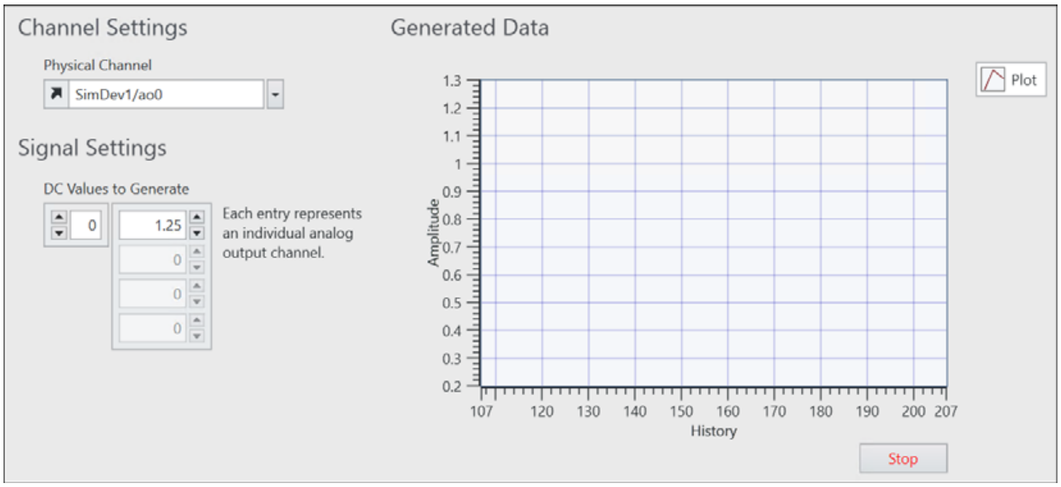


3. Duplicate **On Demand Output Voltage.gvi** so that you can modify the without affecting the source files for the example program.

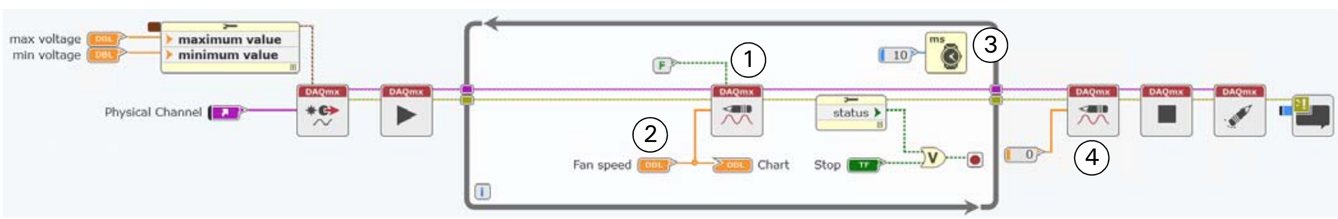
   ☐ Right-click **On Demand Output Voltage.gvi** and select **Duplicate**.

   ☐ Rename the VI in the Files pane as `Fan Speed Control.gvi`.

4. Open **Fan Speed Control.gvi.**

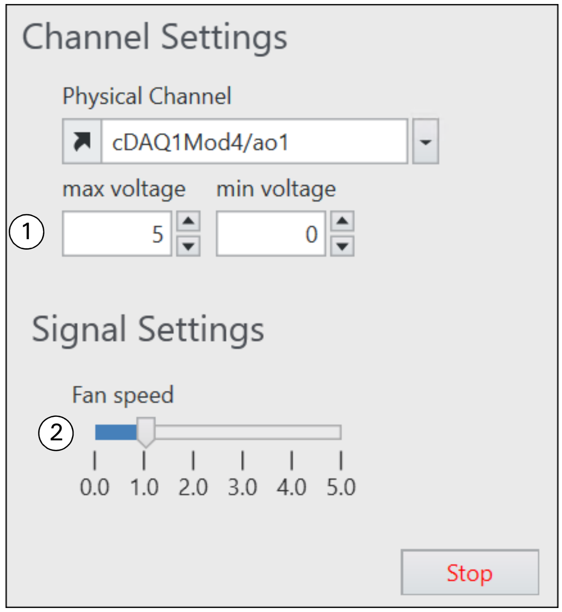


5. Modify the diagram of the VI.



1	**DAQmx Write**—Configure this node to write a single, double-precision sample to a single channel.	
2	**Fan Speed input**—Rename the control to `Fan Speed`. Right-click this control and select **Change to scalar** so that it creates a single element instead of an array.	
3	**Wait Until Next ms Multiple**—For this example, we use software timing because there is no need to create a buffer of Fan Speed values.	
4	**DAQmx Write**—Insert another DAQmx Write node after the loop to turn off the fan when you stop the VI.	

6. Modify the panel.



1	The Max Voltage and Min Voltage values correspond to the highest and lowest voltage values that you expect to measure. 0 to 5 corresponds to the voltage range that the fan uses to set the fan speed.
2	Replace the Fan Speed numeric with a Horizontal Slider. Set the maximum value to 5, since that value corresponds to the top speed for the fan.

7. Save the VI and the project.

Test

Verify that you are able to control the speed of the fan on the Sound and Vibration Signal Simulator.

1. Configure the Sound and Vibration Signal Simulator so that your VI will be able to control the fan speed.

   ☐ Ensure that the Fan Speed Control switch is set to **BNC**.

   ☐ Ensure that the switch between the fans is set to **Balanced Fan**.

2. Run the VI.

3. Adjust the value of Fan Speed on the FP. Observe that the fan speed on the Sound and Vibration Signal Simulator changes accordingly.

Challenge

1. If you completed Exercise 5-3, open `Read Fan Acceleration.lvproject` and copy `Read Fan Acceleration.gvi` to the Fan Speed Control project.

   ☐ Close the Read Fan Acceleration project.

2. Open the Read Fan Acceleration VI in the Fan Speed Control project and modify it to read tachometer values as you adjust the fan speed from `Fan Speed Control.vi`.

   ☐ Modify `Read Fan Acceleration.vi` to measure the current fan speed.

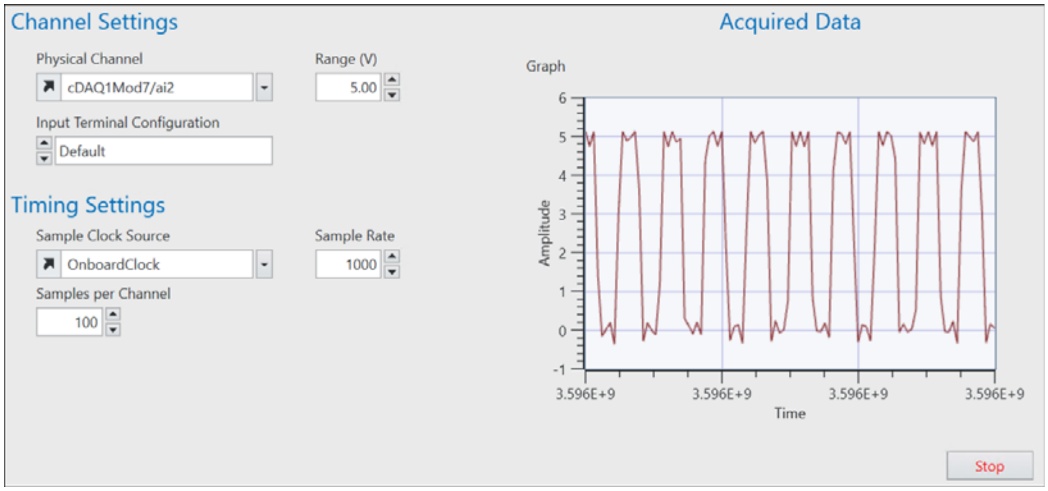
   – Instead of reading accelerometer values from AI0 and AI1 for the NI 9234, read data from AI2.

   AI2 is connected to the Tach Out terminal on the Sound and Vibration Signal Simulator. That terminal outputs data from a tachometer that reads the fan speed from the selected fan.

3. Run Fan Speed Control VI and set the fan speed to 5.

4. Run Read Fan Acceleration VI.

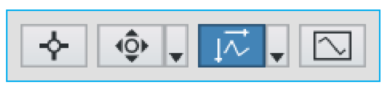
5. Stop Read Fan Acceleration VI.



6. Calculate the fan speed at max speed.

   Beneath the Tach Out terminal on the Sound and Vibration Signal Simulator, there is a label that indicates that the tachometer generates two pulses per revolution of the fan.

   – Show the Graph Tools for the waveform graph and use the Zoom tool to view approximately two cycles of tachometer data.



   – How long does it take for two cycles to complete?____________________

   – Given that two pulses correspond to one revolution of the fan, how fast is the fan spinning (in RPM)?____________________

   – Check the Sound and Vibration Signal Simulator Box Information on page A-10 to compare your findings to the specifications.

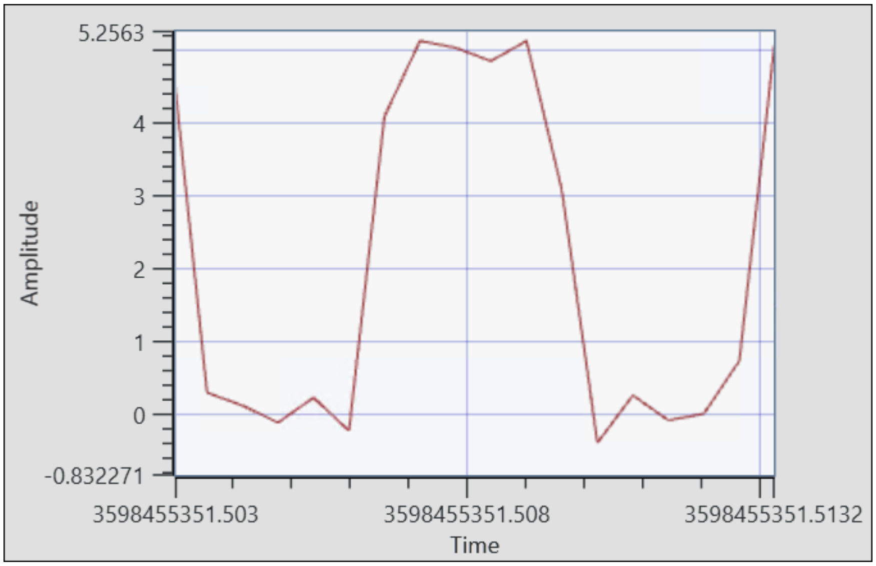
(Answer on page 5-28.)

Answers from page 5-26.

– How long does it take for two cycles to complete?

In the screenshot below, two cycles takes approximately .01 seconds to complete.

.5132 s- .503 s = .0102 s



– Given that two pulses correspond to one revolution of the fan, how fast is the fan spinning (in RPM)?

The fan is spinning approximately 1 revolution per .01 second:

$$\frac{1\,rev}{0.1\,s} = 100\frac{r}{s} = 100\frac{r}{s} \times 60\frac{s}{m} = 6000\,RPM$$

End of Exercise 5-4

6 Programming Multiple Channels

Topics

+ Resources

Exercises

Exercise 6-1 Measuring Voltage and Temperature in a Single Task

Resources

Resources Refer to the following resource for more information about topics in this lesson. For locations with info codes, go to `ni.com/info` and enter the code to access the article.

Selecting More than One Channel on a Device

Article	Location
Physical Channel Syntax	Info code: `channelsyntax`

Use Different Modules in Same Task

Article	Location
Channel Expansion Explained	Info code: `channelexp`

When to Use Channel Expansion

Article	Location
Multidevice Tasks— • C Series • S Series • DSA, SC Express, and X Series	Info code: `multitask`

Exercise 6-1 Measuring Voltage and Temperature in a Single Task

Goal

Create a VI that communicates over multiple channels of the same type.

Scenario

You need to make sure that the solar panel controller does not overheat while testing the solar panels under maximum voltage. You want to write an application to measure the temperature and voltage generation.

In this exercise, you create a VI that measures both temperature and voltage using two different C Series modules but only one task.

Requirements

- The solar panel temperature shall not exceed 30 °C.
- A J-type thermocouple is placed over the solar panel to measure the temperature.
- Solar panel wiring information is located in Appendix A.

Implementation

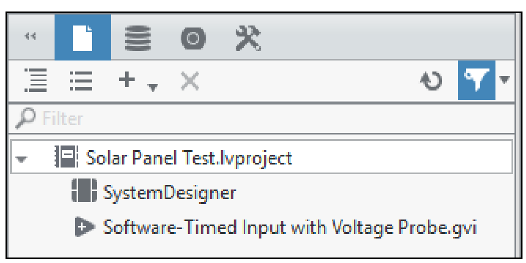
1. Create a new LabVIEW project containing an analog input VI.

    a. On the Learning tab, select **Examples»Hardware Input and Output» NI-DAQmx»Programmatic Configuration»NI-DAQmx Analog Input.**

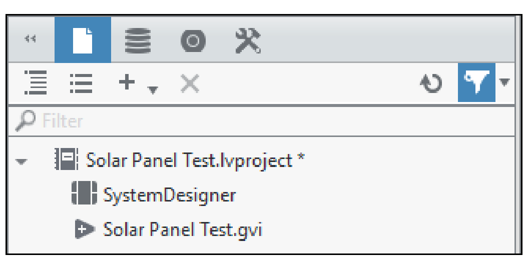
    b. Enter `Solar Panel Test` as the Project Name and click **Create**.

    c. Delete every VI in the project except `Software-Timed Input with Voltage Probe.gvi`.

    d. Save the project.

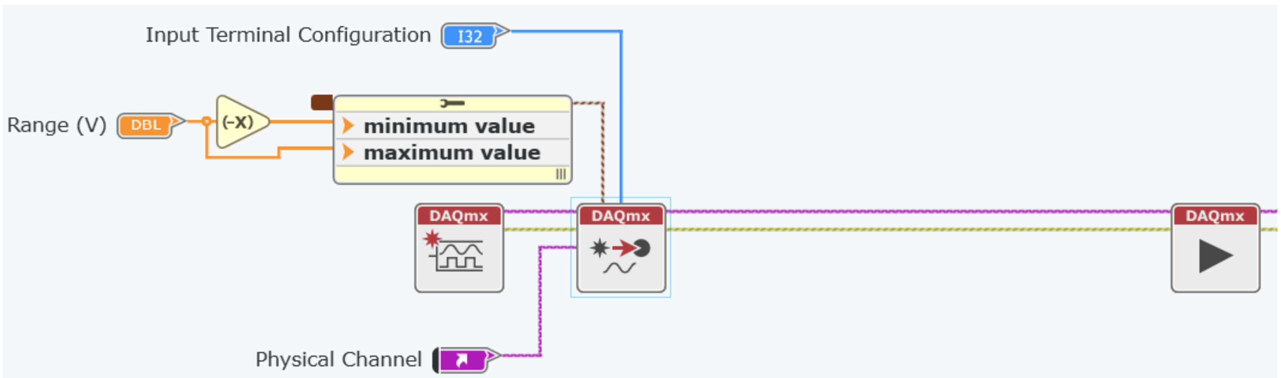


    e. Rename the VI `Solar Panel Test.gvi` and open it.

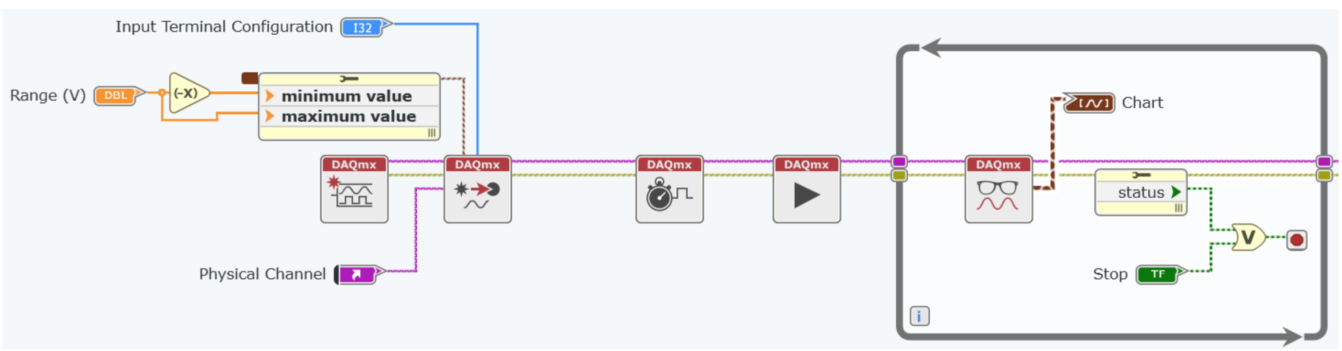


2. On the diagram, make room between the DAQmx Create Virtual Channel VI and the DAQmx Start Task VI.

Tip Press the <Ctrl> key while dragging the cursor from right to left to easily make room.

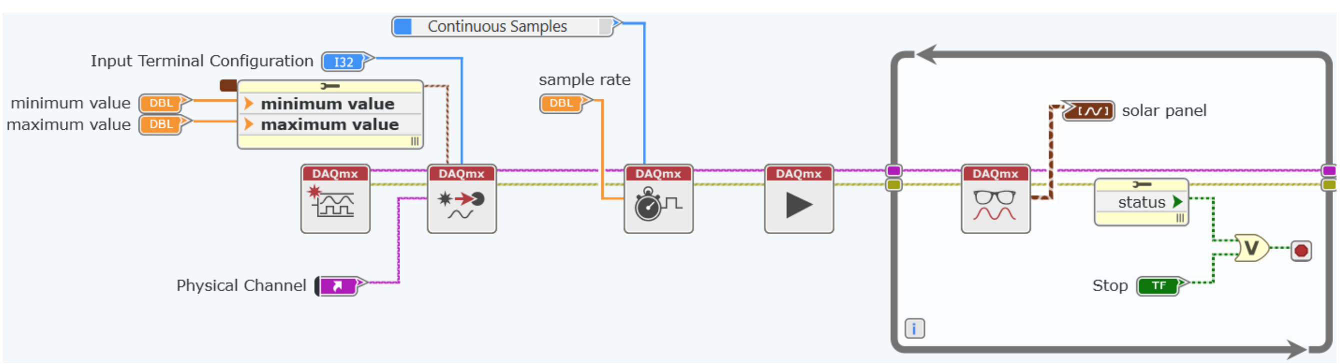


3. Change the timing of the VI from software to hardware by removing the Wait (ms) function and Loop Rate control and adding an NI-DAQmx Timing VI.



4. Update the diagram to take continuous samples from the solar panel and display the values on a waveform chart.

   a. Add or update the controls and indicators as shown in the following diagram.



b. Run the VI to make sure it is taking proper voltage values from the solar panel.

☐ On the panel, enter the values for **Physical Channel**, **Maximum Voltage**, **Minimum Voltage**, and **Input Terminal Configuration**.

☐ Turn off auto-scaling on the y-axis for the waveform chart and set the min and max voltages according to your specifications for the solar panel.

☐ Change the **Sample rate** to `10` Hz.

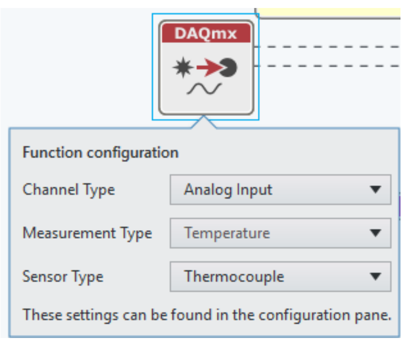
c. Stop the VI.

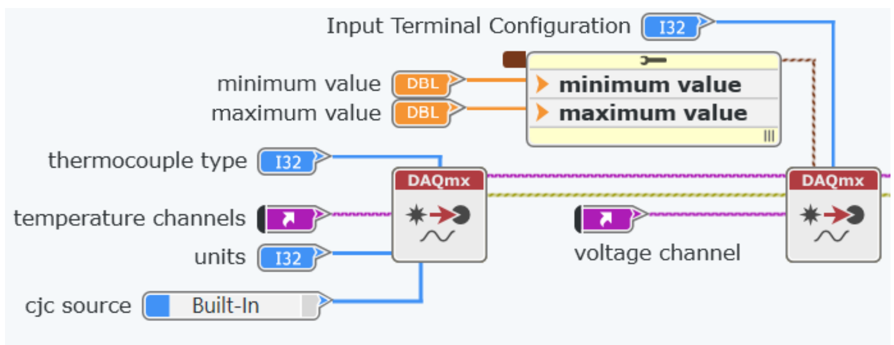
5. Now that the task can take a voltage measurement, add a virtual channel to the task to measure temperature.

   a. Which instance of the DAQmx Create Virtual Channel VI should you select to take a temperature measurement from your hardware kit?_________________

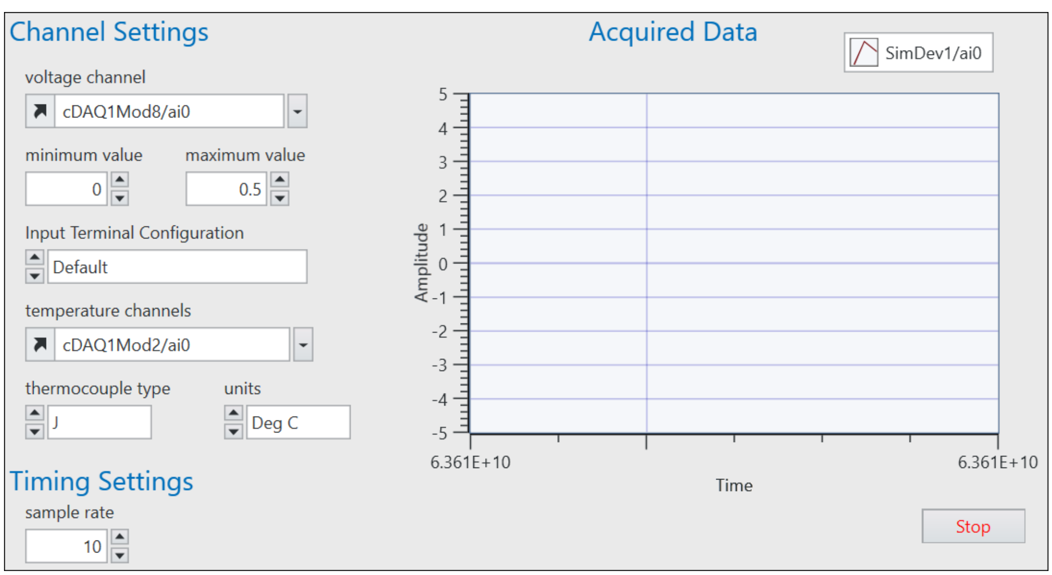
   b. From the specs for the NI 9213, what **cjc source** should you use?____________

   c. You do not have to create constants or controls for all of the input to the DAQmx Create Virtual Channel VI, but make sure your diagram contains the following controls and constants.



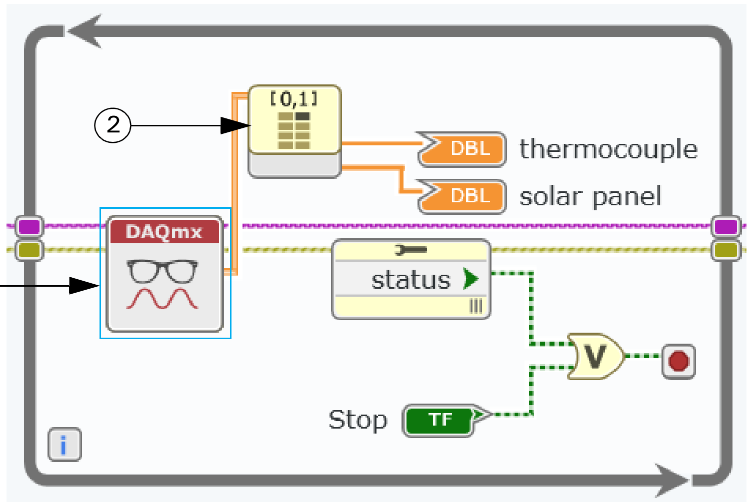


   d. Clean up the panel to accommodate the new controls.



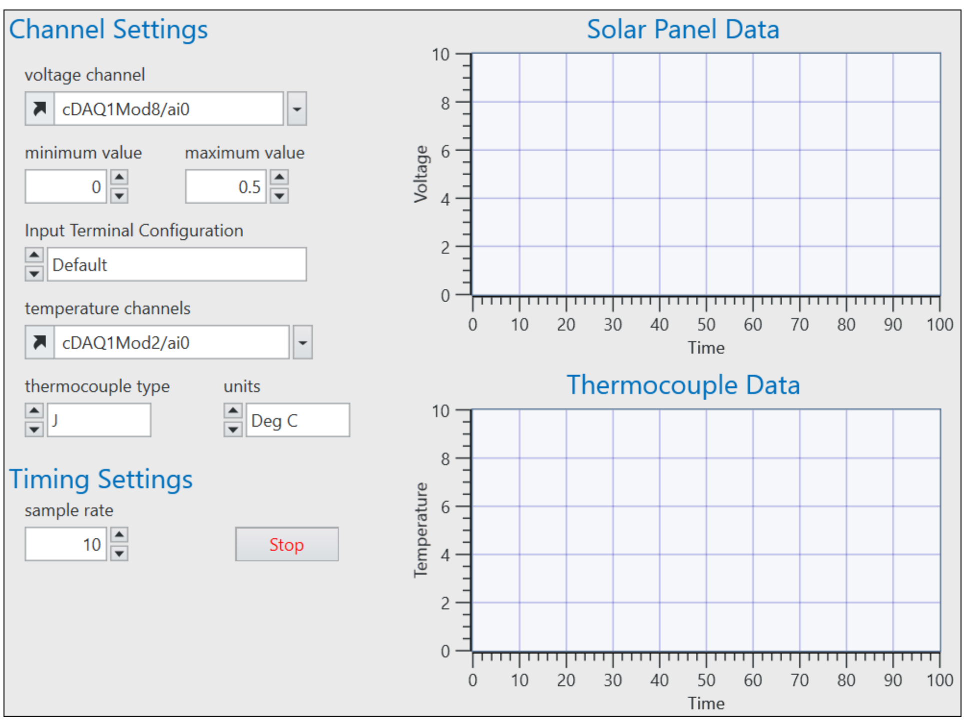
6. Update the diagram to include a second chart to map the temperature reading.

   a. You'll need to include an Index Array to split the output. Make sure you adjust the DAQmx Read node.



1	**DAQmx Read**—Change to Analog Input, Floating Point, Multiple Channels, Single Sample.
2	**Index Array**—Splits the output from the two sensors.

   b. Clean up the panel to see the new charts.



7. Run the VI to make sure it is taking proper temperature values from the thermocouple.

   ☐ On the panel, enter the values for **Temperature, Thermocouple Type**, and **Temperature Units**.

   ☐ Turn off auto-scaling on the y-axis of the Thermocouple waveform chart and set the min and max amplitude according to your estimate of the range of temperature values.

   ☐ Change the **Sample rate** to `10` Hz, if necessary.

Test

Verify the VI can acquire both temperature and voltage data.

1. Verify that the input values on the panel are correct and run the VI.

   Tip Refer to Appendix A, Section C, *Thermocouple Information* for information about the wiring of the thermocouple.

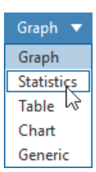
2. Hold the thermocouple between your fingers to increase the temperature.

3. Cover the light sensor or use a flash to decrease and increase the sensor lighting.

4. Stop the VI after acquiring some data.

Now you want to see the data values for analysis.

1. Right-click in either the Solar Panel Data or Thermocouple graphs and select **Export»Capture Data**.



2. On the Capture Data tab, double-click a capture to view the data. Change the view by selecting an option on the top-right of the document workspace.

3. How many points of solar panel data were collected? How many points of thermocouple data were collected?____________________________________

4. Right-click a capture in the Captured Data tab and select **Export**. Note that you can export as CSV or TDMS files.

Challenge

You want modify your application to measure the temperature and voltage generation and notify the technician of an overheating situation. Update your Solar Panel Test VI according to the following requirements.

- Add a control that allows the user to set the temperature at which a warning notification displays.
- Display an indicator the easily communicates the status of the solar panel temperature. For example, the indicator could be green during normal operation and red when the temperature exceeds the limit set by the user.

End of Exercise 6-1

7 Triggering on a Specific Condition

Topics

+ Resources

Exercises

Exercise 7-1 Adding Hardware Triggers

Resources

Resources Refer to the following resource for more information about topics in this lesson. For locations with info codes, go to `ni.com/info` and enter the code to access the article.

Triggering to Synchronize

Article	Location
Synchronizing Analog Input C Series Modules with NI-DAQmx	Info code: `synchcseries`

Implementing a Pause Trigger

Article	Location
Tips and Techniques in Data Acquisition Triggering	Info code: `techniqdaqmx`

Analog vs Digital Triggering

Article	Location
Tips and Techniques in Data Acquisition Triggering - NI-DAQmx	Info code: `techniqdaqmx`

Exercise 7-1 Adding Hardware Triggers

Goal

Use the DAQ modules in your suitcase to generate a tone when a digital trigger is received on PFI0.

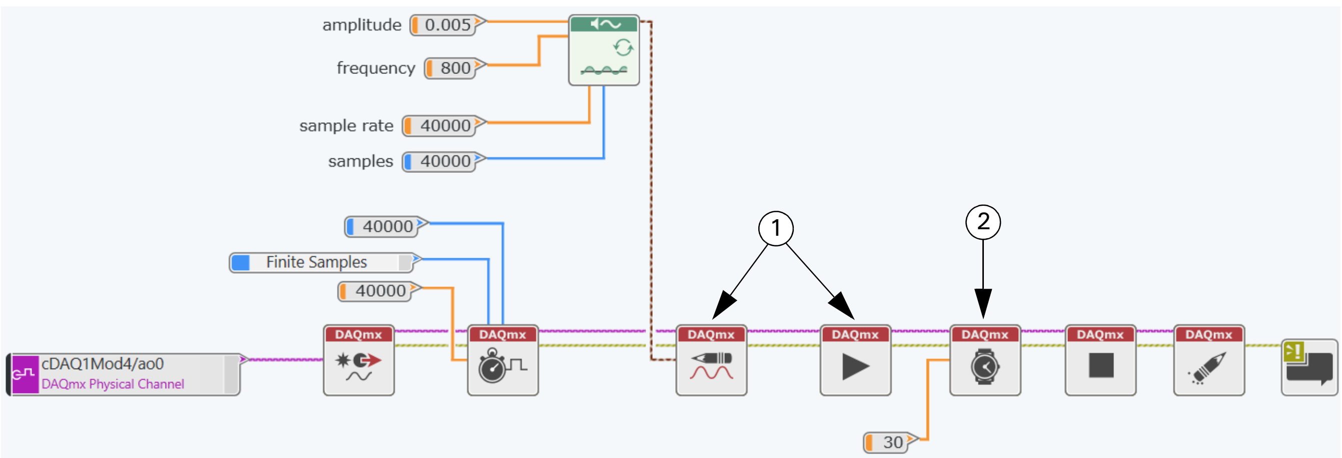
Scenario

You have a VI that generates a tone at a given frequency through a speaker.

In this exercise, you add a hardware trigger to the VI so that the tone only occurs after you push the trigger button on the cDAQ Measurements Demo Box.

Implementation

1. Open the Tone Generation VI, located in the `<Exercises>\DAQNXG\Tone Generation` folder and display the diagram.



1 For buffered write operations, the DAQmx Write must run before the DAQmx Start Task to ensure that the buffer has data to write when the task starts.
2 **DAQmx Wait Until Done**—This node ensures that the tasks is not accidentally cleared before all samples are written.

2. Explore the diagram.

   a. Which module should you specify to generate the signal?________________

   b. Is this a finite or continuous signal generation?________________________

   c. How long will the tone last?____

   d. Why does the DAQmx Write VI come before the DAQmx Start VI in this task?

   ____

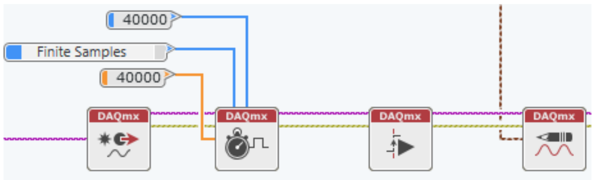
   ____

   ____

   e. Which DAQmx VI ensures that the specified operation is complete before stopping the task?____

3. Update the diagram so that the signal generation waits until you trigger it with the Trigger button on the cDAQ Measurements Demo Box.

   a. Insert the DAQmx Trigger node between the DAQmx Timing node and the DAQmx Write node.

   b. Select the correct instance of the DAQmx Trigger VI. Consider the following questions to determine the correct instance.

      - Is the Trigger button an analog or digital control?____________________

      - Do you want the tone to sound before or after you release the Trigger button? ____

4. Create a constant or control to specify the **source** of the DAQmx Start Trigger.

5. Enter an appropriate value for the source of the trigger.

Tip What line is the Trigger button on the demo box connected to?

____

(Refer to Table A-1, *Connections in the NI CompactDAQ Measurements Demo Box*, in Appendix A, to determine which channel is connected to the Trigger button.)

Test

1. Verify that all controls and constants refer to valid channel in the demo box.

2. Click the run button on the VI and verify that no sound is generated.

3. Press the Trigger button on the demo box to hear the generated sound.

End of Exercise 7-1

8 Timing and Synchronization Methods

Topics

+ Resources

Exercises

Resources

Resources Refer to the following resource for more information about topics in this lesson. For locations with info codes, go to `ni.com/info` and enter the code to access the article.

What clocks does your device have?

Article	Location
Clocks	Info code: `clocks`

Jitter

Article	Location
Digital Timing: Clock Signals, Jitter, Hysteresis, and Eye Diagrams	Info code: `jitter`

Using External Clock Signals

Article	Location
Reference Clock Synchronization	Info code: `clocksynch`

Master Timebase Implementation

Article	Location
Master Timebase Synchronization	Info code: `synchtimebase`

Synchronizing Specific Hardware Series

Article	Location
Synchronizing Analog Input C Series Modules with NI-DAQmx	Info code: `synchcseries`
M Series Synchronization with LabVIEW and NI-DAQmx	Info code: `synchmseries`
Synchronization Explained	Info code: `synchexplain`
Timing and Synchronization Systems	Info code: `synchsystems`

Exercise 8-1 Simultaneously Started Analog Input and Output

Goal

Compare simultaneous data acquisition tasks with tasks using different triggers.

Scenario

In this exercise you explore the problems with using software to trigger different tasks and you create VIs using different methods to simultaneously generate and measure analog signals.

- Part A: Relying on Software to Trigger Tasks
- Part B: Using the Start Trigger to Trigger Tasks
- Part C: Using a Hardware Trigger to Trigger Tasks

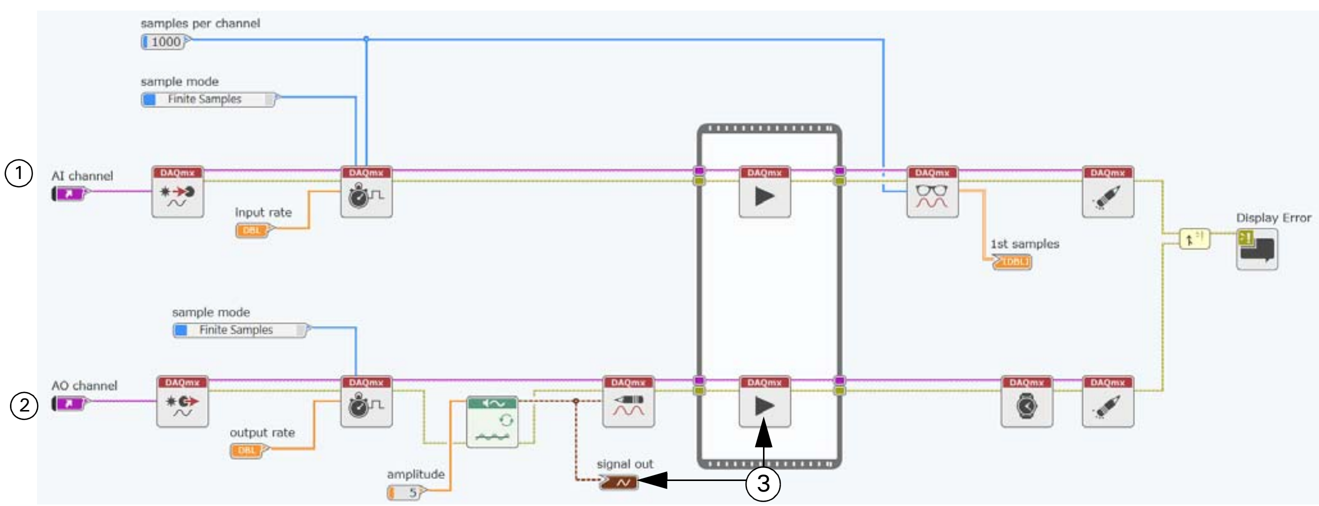
Hardware Configuration

Using two wires, connect an analog output channel from the NI 9263 to an analog input channel of the NI 9215. Be sure to note which channel you use for output and which channel you use for input.

Part A: Relying on Software to Trigger Tasks

1. Open the Start Trigger Synchronization project in the `<Exercises>\DAQNXG\Start Trigger Synchronization` directory.

2. Open `Simultaneous Start.gvi` and examine the diagram.



1 The top row is a task that measures a finite, buffered analog input acquisition and displays the acquired data on the **1st samples** waveform graph.
2 The bottom row is a task that generates a finite, buffered analog output signal. The output waveform is generated by the Sine Waveform node and is displayed on the **signal out** graph.
3 Note that the **signal out** graph displays the waveform before the waveform is generated on the analog output device.

3. Set the panel controls for the Simultaneous Start VI.

- **AI Channel**: Specify the channel you wired in *Hardware Configuration*.
- **Input Rate**: `1000`
- **AO Channel**: Specify the channel you wired in *Hardware Configuration*.
- **Output Rate**: `1000`

4. Run the VI several times and compare what you see in the two graphs.

☐ Does the **1st samples** graph display the same waveform every time it runs?

____

☐ What causes the flat line at the beginning or end of the **1st samples** plot?

____

What's Happening

Because the VI is using the Flat Sequence Structure and the two NI-DAQmx Start Task VIs to trigger and begin the data acquisition, the OS is controlling when the analog output and analog input tasks begin. As you see from the graphs, the analog input often begins before the analog output. This causes the analog input to finish after all of the output signal has been generated. The flat line at the end of the graph indicates that the buffer was empty when the analog input read its final samples.

Part B: Using the Start Trigger to Trigger Tasks

1. Modify the VI to start two tasks at the same time by using the implicit `StartTrigger` signal which the NI-DAQmx Start VI calls.

   a. Right-click `Simultaneous Start.gvi`, duplicate it, and rename it `Shared Start Trigger.gvi`

   b. Before updating the diagram, answer the following questions:

☐ Which task, analog input (top row) or analog output (bottom row), is the master task and which task is the slave task? Why?

____

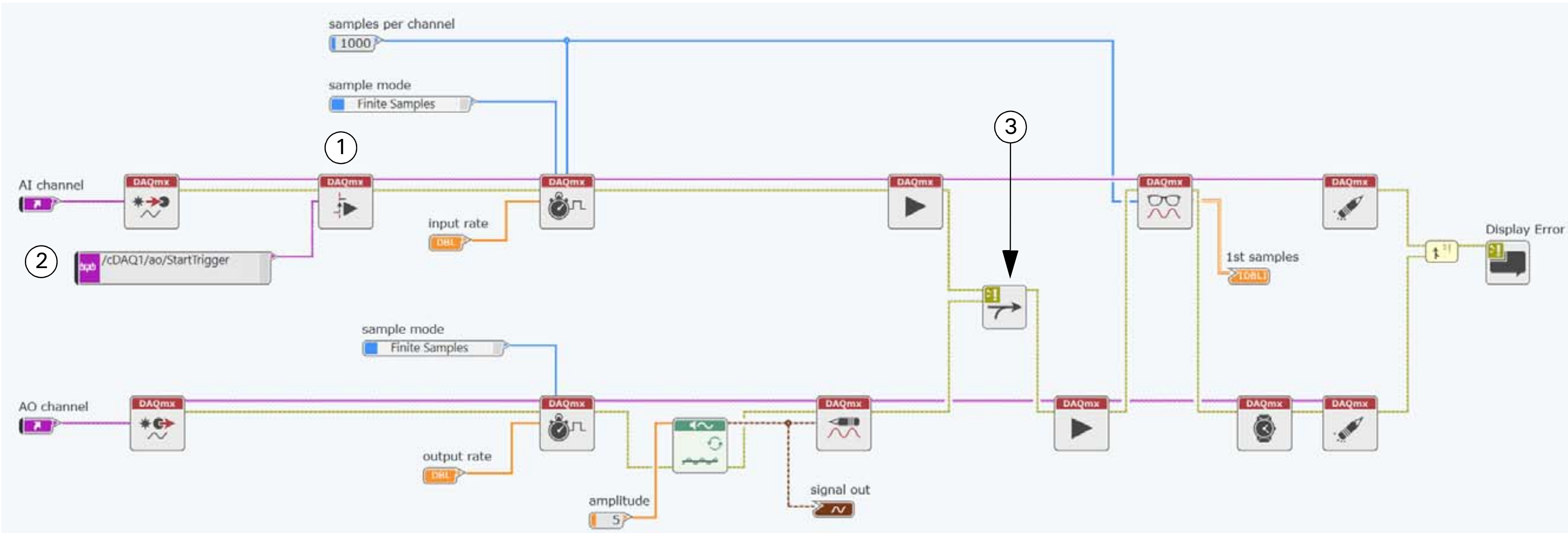
____

☐ Because you must complete the configuration of all slave tasks before starting the master task, which Start Trigger should you use in this VI? `ai/StartTrigger` or `ao/StartTrigger`?

____

Tip From the *NI-DAQmx Help*: To perform Start Trigger synchronization, configure the start triggering on all slave tasks by setting the trigger source to the internal Start Trigger terminal of the master task, such as `ao/StartTrigger`.

c. Complete the diagram as shown below.



1 **DAQmx Trigger**—Configure as **Start** and **Digital Edge**.
2 ***/cDAQ1*/ao/StartTrigger**—By specifying the `ao/StartTrigger` as the trigger source, you ensure that the acquisition does not begin before the output task begins generation because the analog input task is dependent on the analog output trigger. *cDAQ1* refers to your device.
3 **Multiplex Errors**—Merging the analog input task data flow with the analog output task data flow ensures that the analog input task does not start before the analog output task is ready.

2. Save the VI.

Test

1. Set the panel controls for the Shared Start Trigger VI.

- **AI Channel**: Specify the channel you wired in *Hardware Configuration*.
- **Input Rate**: `1000`
- **AO Channel**: Specify the channel you wired in *Hardware Configuration*.
- **Output Rate**: `1000`

2. Run the VI.

3. Run the VI several times and compare what you see in the two graphs.

   ☐ Does the **1st samples** graph display the same waveform every time it runs?

   ____

   ☐ Is the **1st samples** plot the same as the **signal out** plot?

   ____

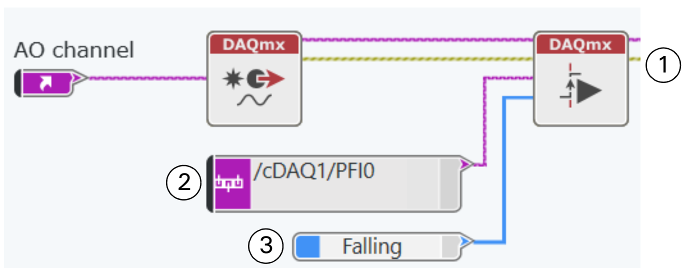
What's Happening

By specifying the `ao/starttrigger` for the analog input task, you guarantee that both tasks begin at the same time because the analog output NI-DAQmx Start Task VI implicitly calls the `ao/starttrigger` if no other trigger is specified. In other words, on a hardware level, both tasks begin on the exact same clock pulse.

Part C: Using a Hardware Trigger to Trigger Tasks

1. Right-click `Shared Start Trigger.gvi`, duplicate it, and rename it `Multiple Triggers.gvi`

2. Add a hardware trigger to start two tasks simultaneously, as shown below.



1	**NI-DAQmx Trigger**—Select the **Start»Digital Edge** instance.
2	**source** constant—Right-click the **source** input, select **Create»Constant** and specify `/cDAQ1/PFI0` as the value.
3	**edge** constant—Right-click the **edge** input, set the edge constant to `Falling` so that it triggers when the trigger button on the demo box is pushed.

3. Save the VI.

Test

1. Run the VI.

2. Press the TRIGGER button on the demo box.

   When you press the TRIGGER button, it produces a falling edge on the PFI 0 line of the cDAQ-9178 chassis. The falling edge received on the PFI 0 line triggers the analog output task to begin. The analog input task is configured to begin on the first edge of the analog output sample clock, so both analog input and analog output tasks will start simultaneously.

3. Stop the VI.

End of Exercise 8-1

Exercise 8-2 Sharing a Sample Clock

Goal

Synchronize the analog input acquisition on a cDAQ device to the analog output generation on a separate DAQ device using a shared sample clock and hardware trigger.

Hardware Configuration

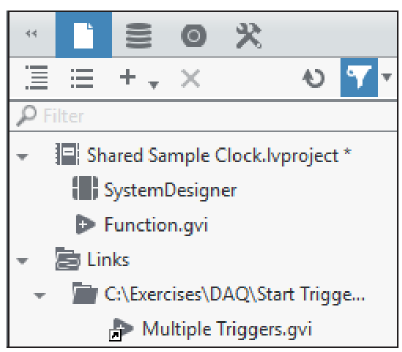
Make sure that the NI 9263 (analog output) module is directly wired to the NI 9215 (analog input) module. You can use the same configuration you used in Exercise 8-1 on page 4.

Scenario

In this exercise, you modify the Simultaneous Start (3) VI from the previous exercise to synchronize the analog output and analog input by sharing the same sample clock.

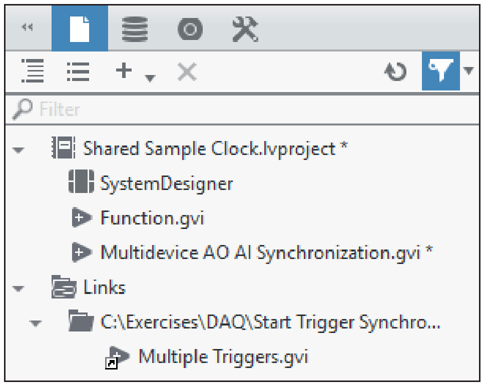
Implementation

1. From the Project tab on the LabVIEW NXG home screen, select **New»VI Project** and create a new project named `Shared Sample Clock`.
2. Select **File»Save all** and save the project to the `<Exercises>\DAQNXG\` directory.
3. Right-click **Shared Sample Clock.lvproject** in the Files pane and select **Add File**.
4. Add `<Exercises>\DAQNXG\Start Trigger Synchronization\Multiple Triggers.gvi` to the project.



5. Right-click **Multiple Triggers.gvi** and select **Duplicate**.

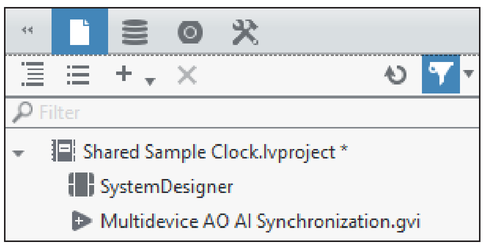
6. Change the name to `Multidevice AO AI Synchronization.gvi`.



7. Clean up the project.

   a. Right-click the link to **Multiple Triggers.gvi** and select **Exclude**.

   b. Right-click **Function.gvi** and select **Delete** to delete it from disk.



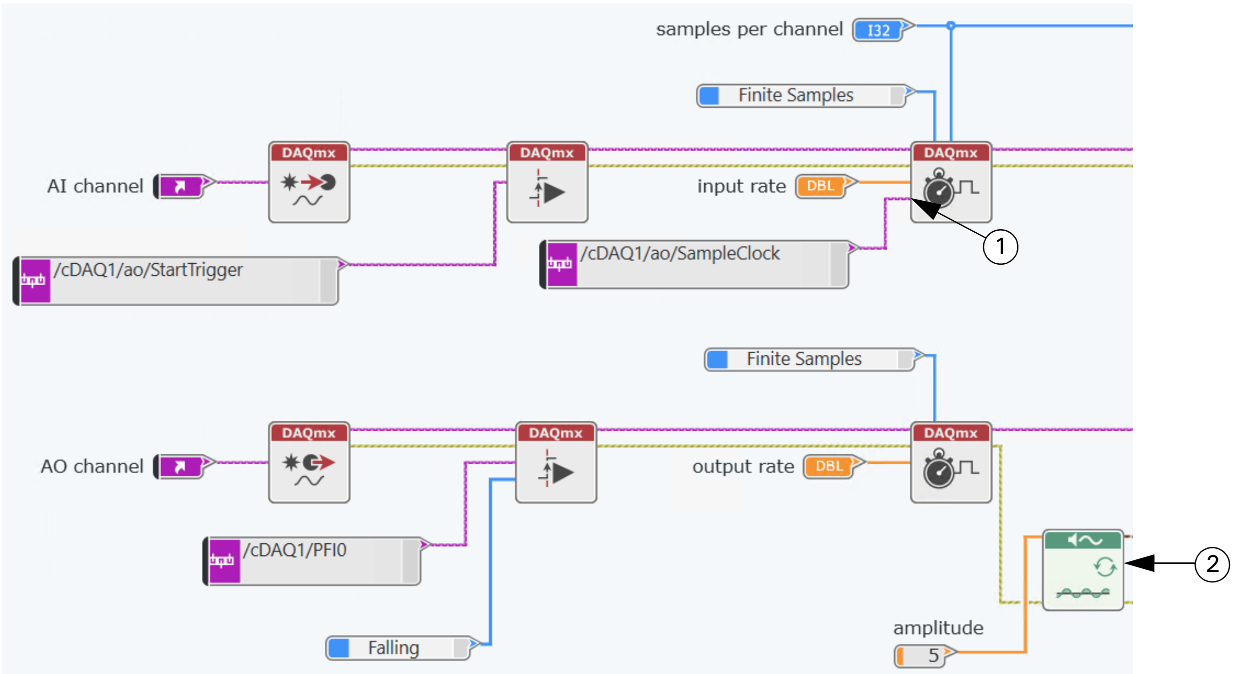
8. Open the diagram of `Multidevice AO AI Synchronization.vi` and answer the following questions.

   a. What is the syntax for specifying that the analog input task use the start trigger from the analog output device?________________________________

   b. Using the same syntax, how do you think you specify the **source** of the analog input DAQmx Timing VI to use the analog output sample clock?____

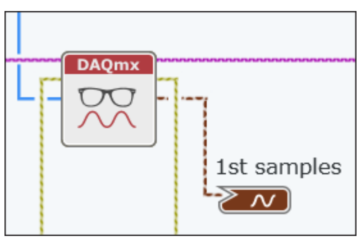
   ____

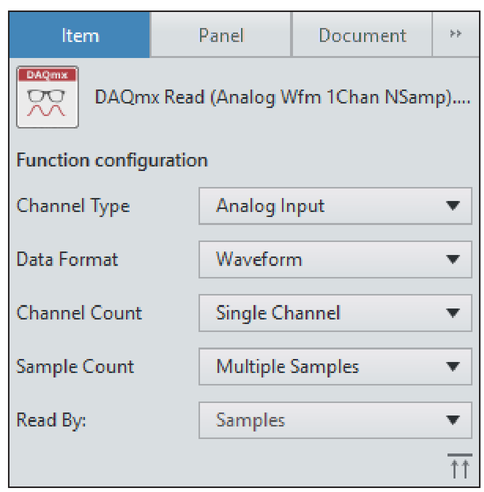
c. Modify the diagram as shown below to share the analog output sample clock.



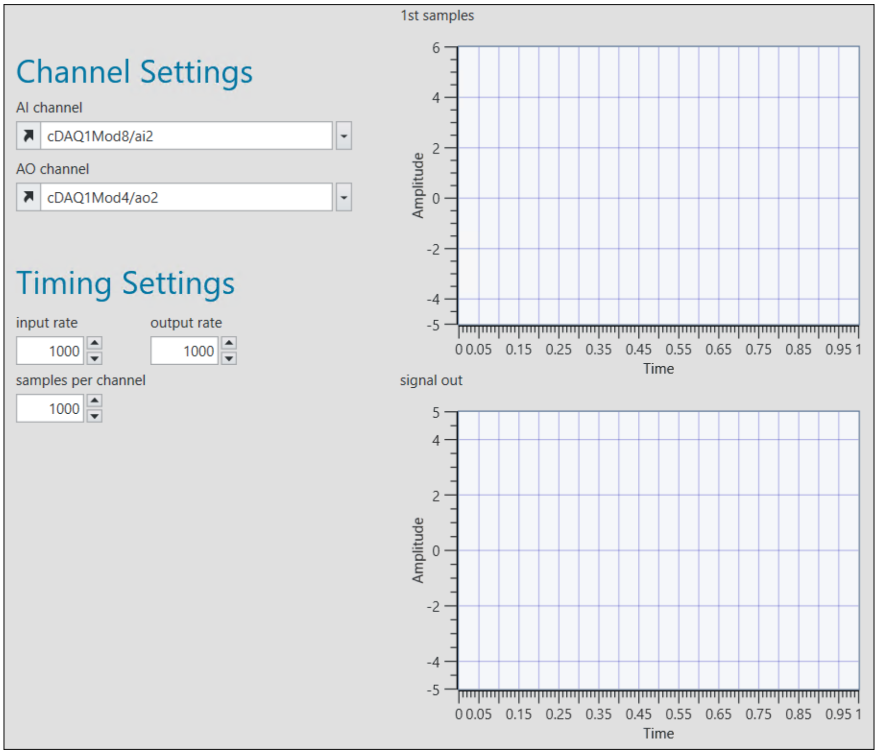
1 Right-click the **source** input of the analog input DAQmx Timing node and select **Create»Constant**. Use the drop down list to select the analog output sample clock, **cDAQ1/ao/SampleClock**.
2 **Wave Generator**—Note that by default, this node generates a sine wave of 1000 samples at 1000 S/s (samples per second).

9. Update the DAQmx Read node to output data in a waveform data type so that the data contains a timestamp.





10. Change the **1st samples** chart on the panel into a waveform graph and adjust the display to match the image below.



11. Save the VI.

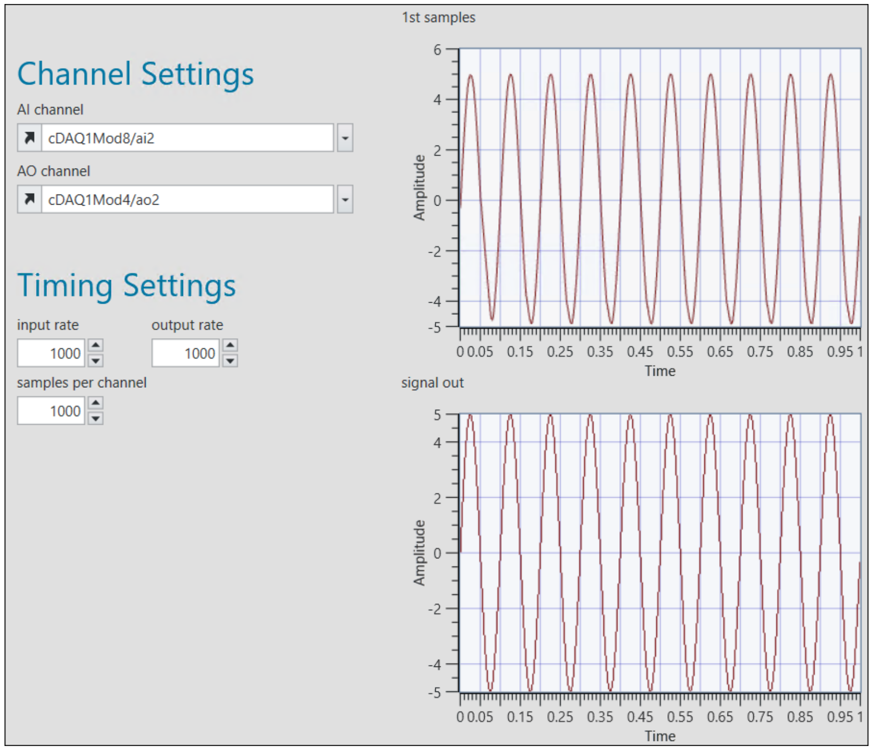
Test

1. Set the panel controls for the analog input and analog output channels you used in Exercise 8-1.

2. Make sure the **input rate** and **output rate** are `1000`.

Tip Right-click a chart on the panel and select **Clear data** to reset the chart.

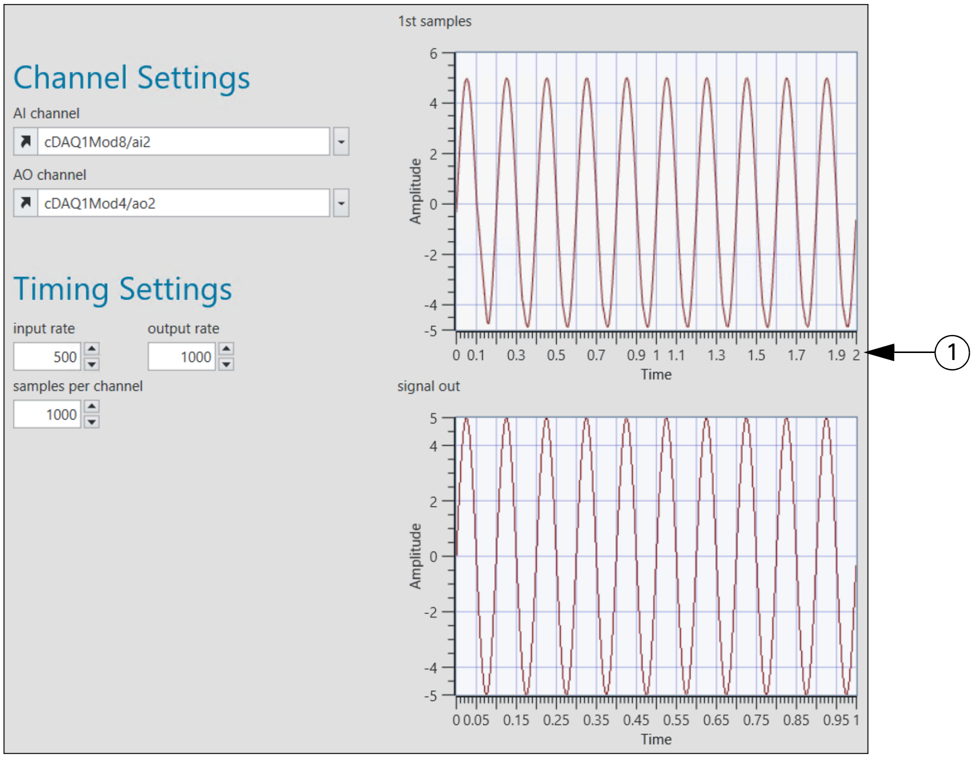
3. Run the VI. After pressing the trigger on the demo box, the panel updates.



Note that the **1st samples** graph updates one second after pressing the trigger button on the demo box. The graph updates in one second because the analog output is generating 1000 samples at 1000 samples per second. (Refer to the diagram notes in step 8 on page 9.)

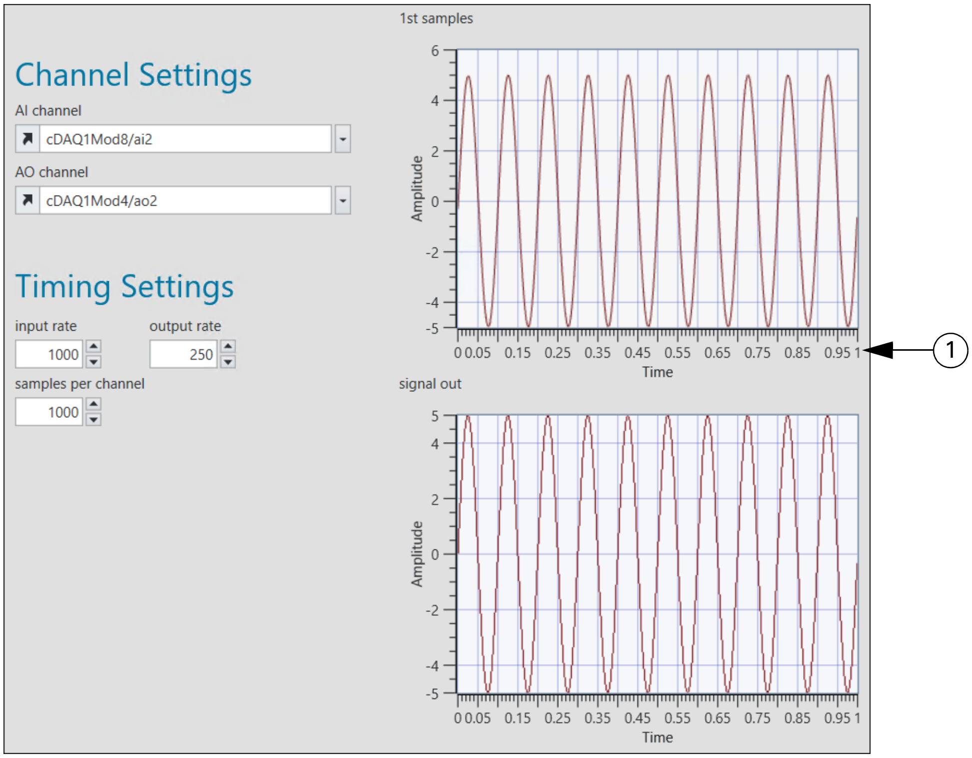
4. Right-click the **signal out** chart and select **Clear data** to reset the chart.

5. Change the **input rate** to `500` and run the VI again.



1 Note that the **1st samples** graph updates at the same rate as before, but the time stamp displays 2 seconds instead of 1. Sharing a sample clock does not allow you to run the acquisition at different rates.

6. Change the **input rate** to `1000`, change the output rate to `250`, and run the VI again. Count the number of seconds it takes to update the **1st samples** chart after you press the trigger button the demo box.



1 Note that the **1st samples** graph time displays 1 second. However, by counting seconds after pressing the trigger button on the demo box, you know that the acquisition actually took four seconds. This chart is displaying inaccurately synchronized data.

7. What happens to the graphs when you change the **output rate** to `1000` and the **input rate** to `2000`? Write your guess here before running the VI to find out.

____

____

____

8. Save and close the VI and project.

End of Exercise 8-2

Exercise 8-3 Synchronizing with a Master Timebase

Goal

Synchronize two operations the acquire data at different rates by using a reference clock as a clock source.

Scenario

You are monitoring how hot a solar panel gets as it is generating voltage over the course of an afternoon. Because temperature does not change as quickly as the voltage generated, you want to sample at a slower rate to save storage space.

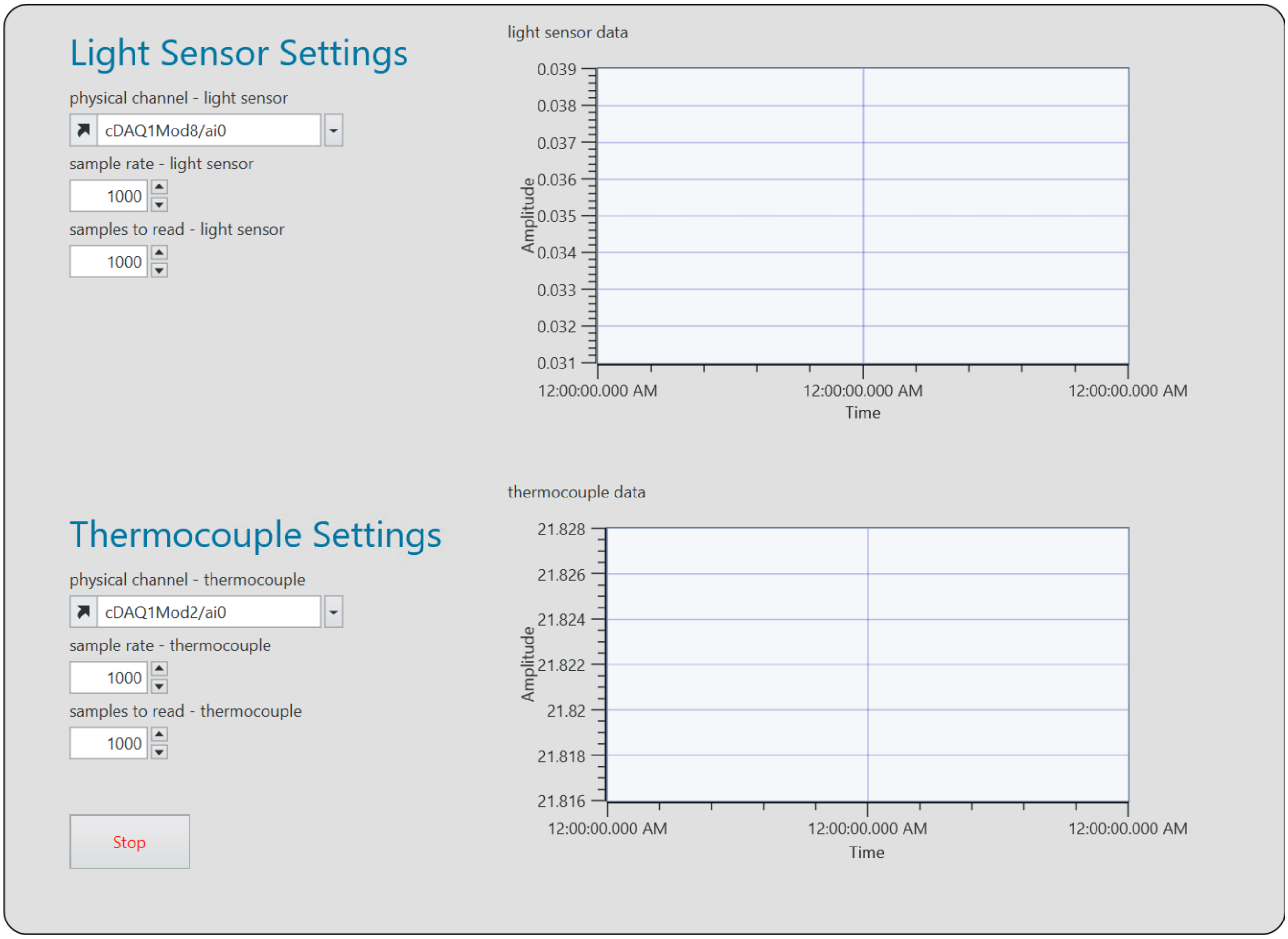
In this exercise you synchronize two C Series modules using a master timebase on the backplane of the cDAQ chassis.

- NI 9213—Delta-Sigma ADC using a scanning sampling mode (slow sample device)
- NI 9215—SAR ADC using successive approximation

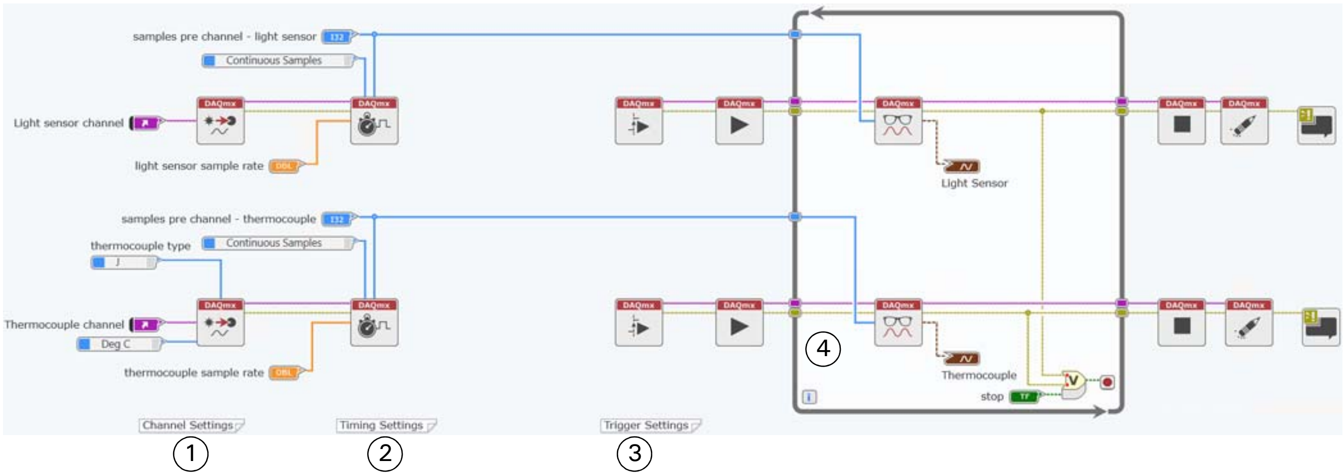
Implementation

1. Open the Timebase Synchronization project in the `<Exercises>\DAQNXG\Timebase Synchronization` directory.

2. From the **Files** tab in the **Navigation** pane, open **Timebase Synchronization.gvi**.

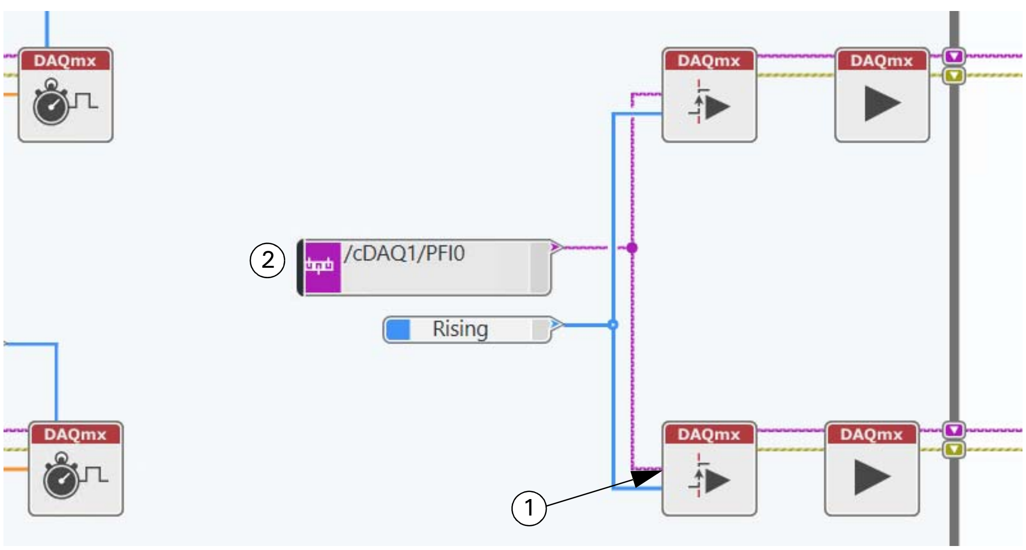


3. Examine the diagram. Part of the VI is already coded for you.



1	Channel settings—The DAQmx Create Virtual Channel VIs are already set up to accept a channel for the light sensor task and the thermocouple task.
2	Timing settings—The DAQmx Timing VIs are set up to take continuous samples at a specified sample rate.
3	Trigger settings—The DAQmx Start Trigger VIs still need a trigger source defined.
4	While Loop—The While Loop already contains code to read data from the light sensor and thermocouple tasks and display that data on a graph on the panel. The number of **samples per channel** input should match the sample rate specified in the DAQmx Timing VI.

4. Update the trigger settings portion of the application so that both tasks start when the trigger button on the demo box is pressed.

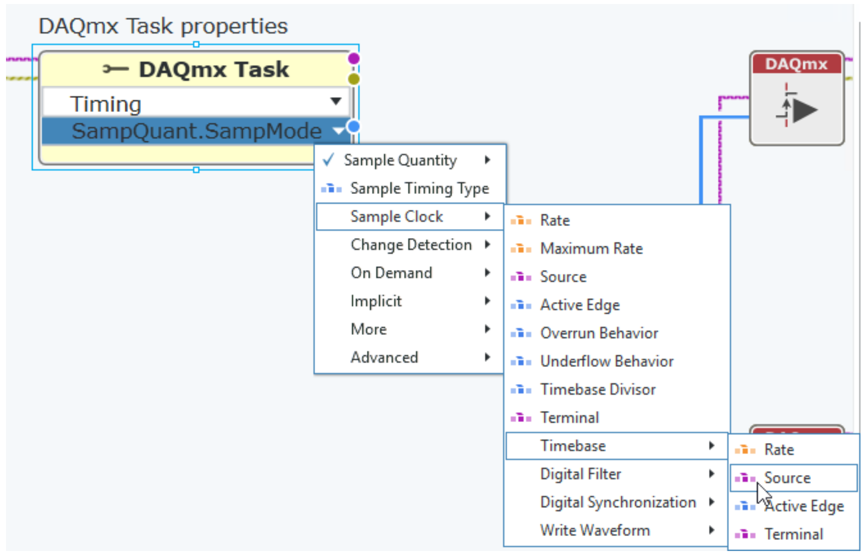


1	Right-click the **source** terminal and select **Create»Constant** from the shortcut menu to create a constant.
2	Click the constant and specify the **PFI0** line for the cDAQ chassis. Refer to Table A-1 to confirm which PFI line is wired to the physical trigger on the box.

Note Because both tasks are using the same external hardware trigger to begin the task, so you do not have to designate one task as slave and one as master.

5. Place a Property Node on the diagram and wire the **task in** and **error** wires from the DAQmx Timing node.

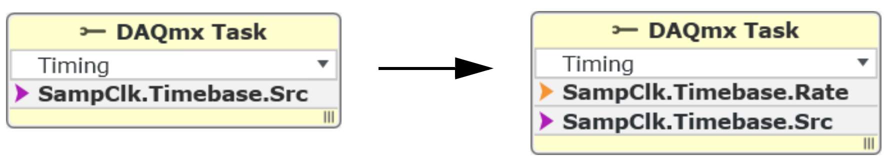
6. Update the node to use the sample clock timebase source property.



Click the terminal and select **Sample Clock»Timebase»Source** to specify the property.

7. Right-click the **SampClk.Timebase.Src** terminal and select **Change to write**.

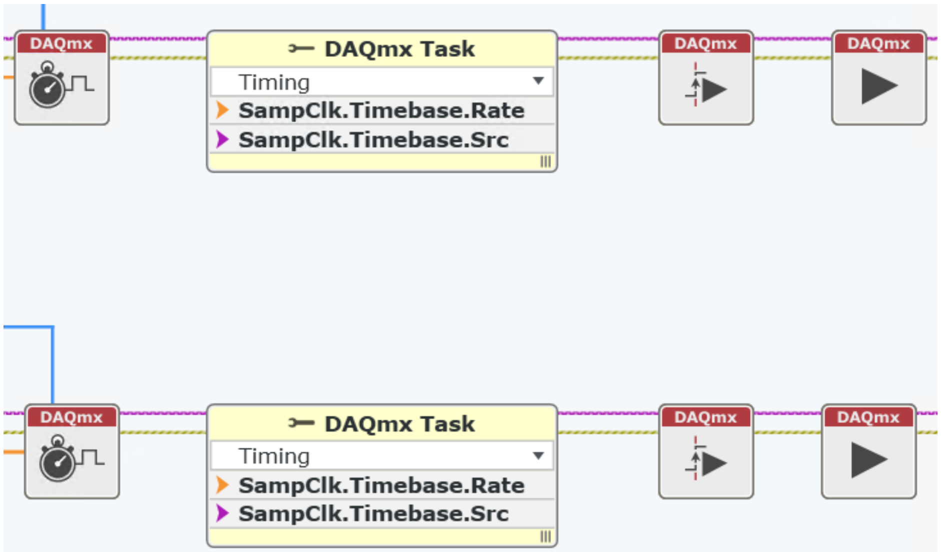
8. Expand the node and display the **SampClk.Timebase.Rate** property, as shown below.



Note Whenever you specify an external source for a timebase, you must also specify the rate of the source.

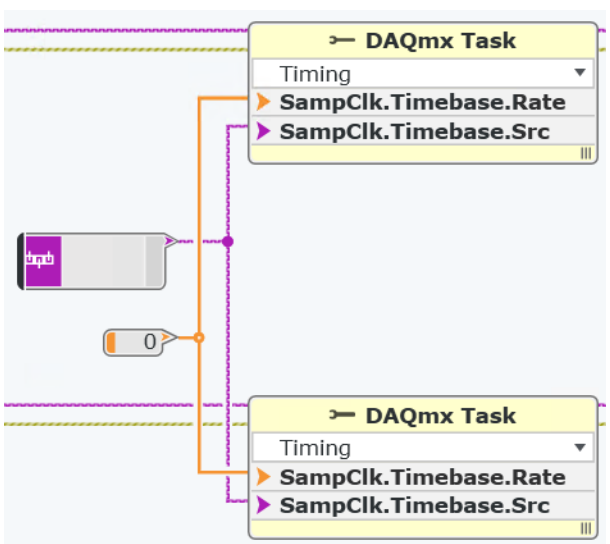
9. Copy and paste the configured timing node so that you have two nodes on the diagram.

10. Wire one Property node in between the DAQmx Timing VI and DAQmx Start Trigger VI for each task, as shown below.



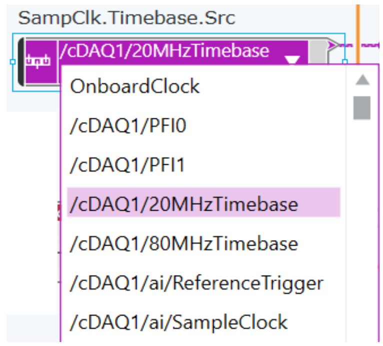
11. Create a constant for both **SampClk.Timebase.Rate** and **SampClk.Timebase.Src** properties and wire them to each node, as shown below.

Tip Right-click the property terminal and select **Create»Constant** from the shortcut menu to create a constant.

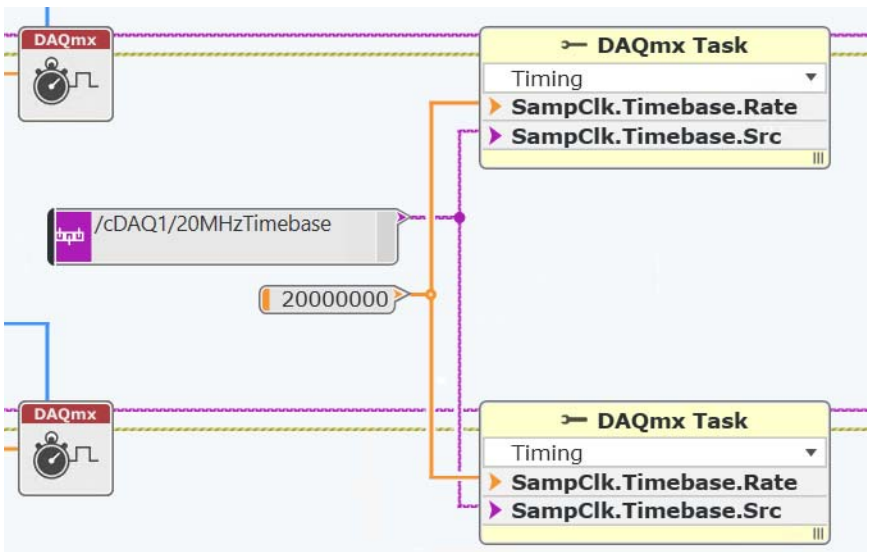


Because you wire the same value to both nodes, each task will use the same source to create its sample clock.

12. Click the I/O Name constant and select a timebase for the sample clock. In this exercise, select the **20 MHz Timebase** from the cDAQ chassis as shown below.



13. Update the constant for the **SampClk.Timebase.Rate** property to be `20000000`.



Note Whenever you specify an external source for a timebase, you must also specify the rate of the source.

14. Save the VI after confirming that the **Run** button is not broken.

Test

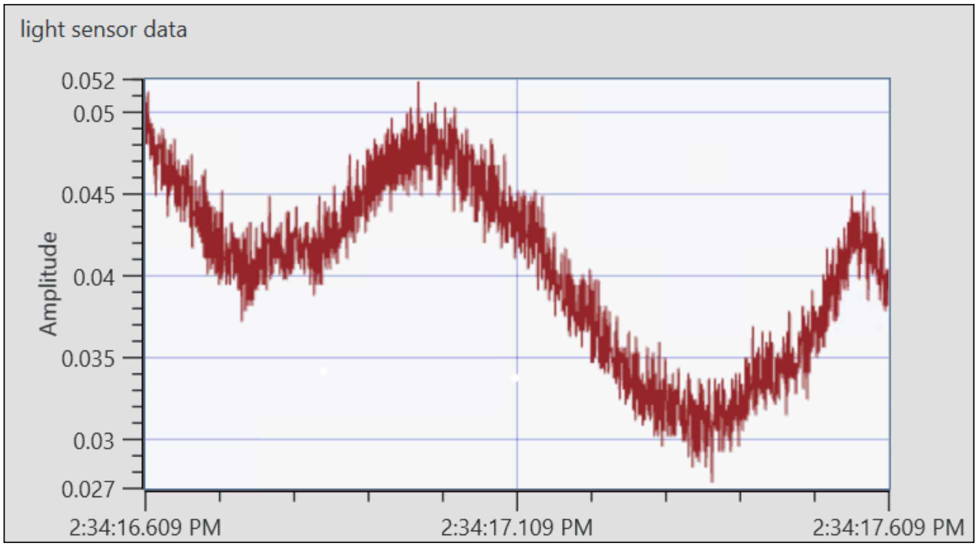
In this test you compare measurements taken at the same sample rate and measurements taken at different sample rates.

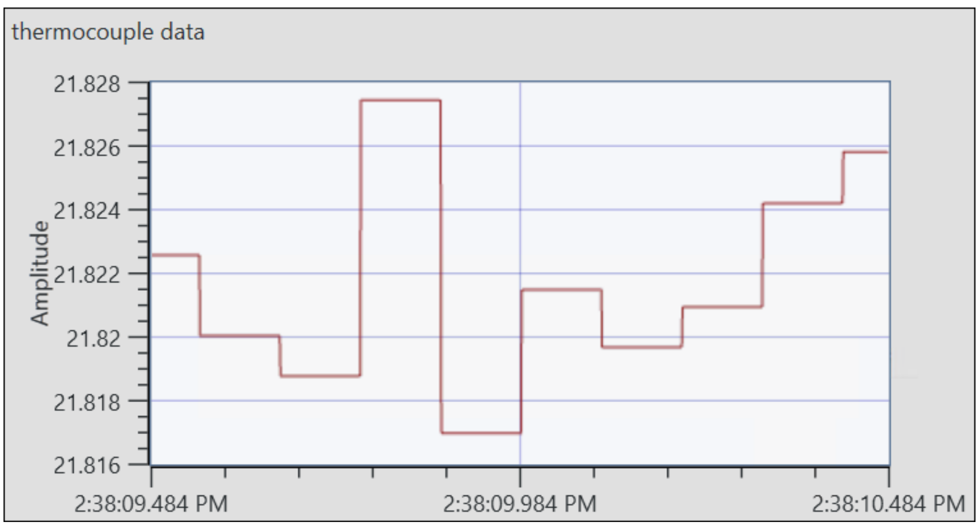
1. Set the controls to the following values:

Control	Value
Physical channel - light sensor	*Channel connected to solar panel in demo box*
Sample rate - light sensor	1000
Samples to read - light sensor	1000
Physical channel - temperature	*Channel connected to thermocouple in demo box*
Sample rate - temperature	1000
Samples to read - temperature	1000

2. Run the VI and press the trigger button on the demo box when you are ready to start your measurements.

3. Adjust the temperature and light reaching the light sensor and then press stop on the panel to stop the VI. Your graphs may be similar to the image below.





Notice that the light sensor data fluctuates rapidly, but the thermocouple data changes only approximately every 0.1 seconds.

Because the temperature does not change as quickly as the voltage from the light sensor, you can save storage space by decreasing the sample rate for the temperature analog input task.

4. Set the **Sample rate - thermocouple** control to `100` and leave the **samples to read - thermocouple** control at `1000`.

   Tip We select a sample rate of 100 because from the graph above we see that the temperature samples change only 10 times. 1000 divided by 10 equals. 100.

5. Run the VI and press the trigger button on the demo box when you are ready to start your measurements.

   Why does the VI time out? ____

   ____

6. Set the **sample rate - thermocouple** control to `100` and set the **Samples to read - thermocouple** control to `100`.

7. Run the VI and press the trigger button on the demo box when you are ready to start your measurements.

8. Send a snapshot of data to the Captured Data tab to compare samples.

   a. Right-click the **light sensor data** graph and select **Capture data**.

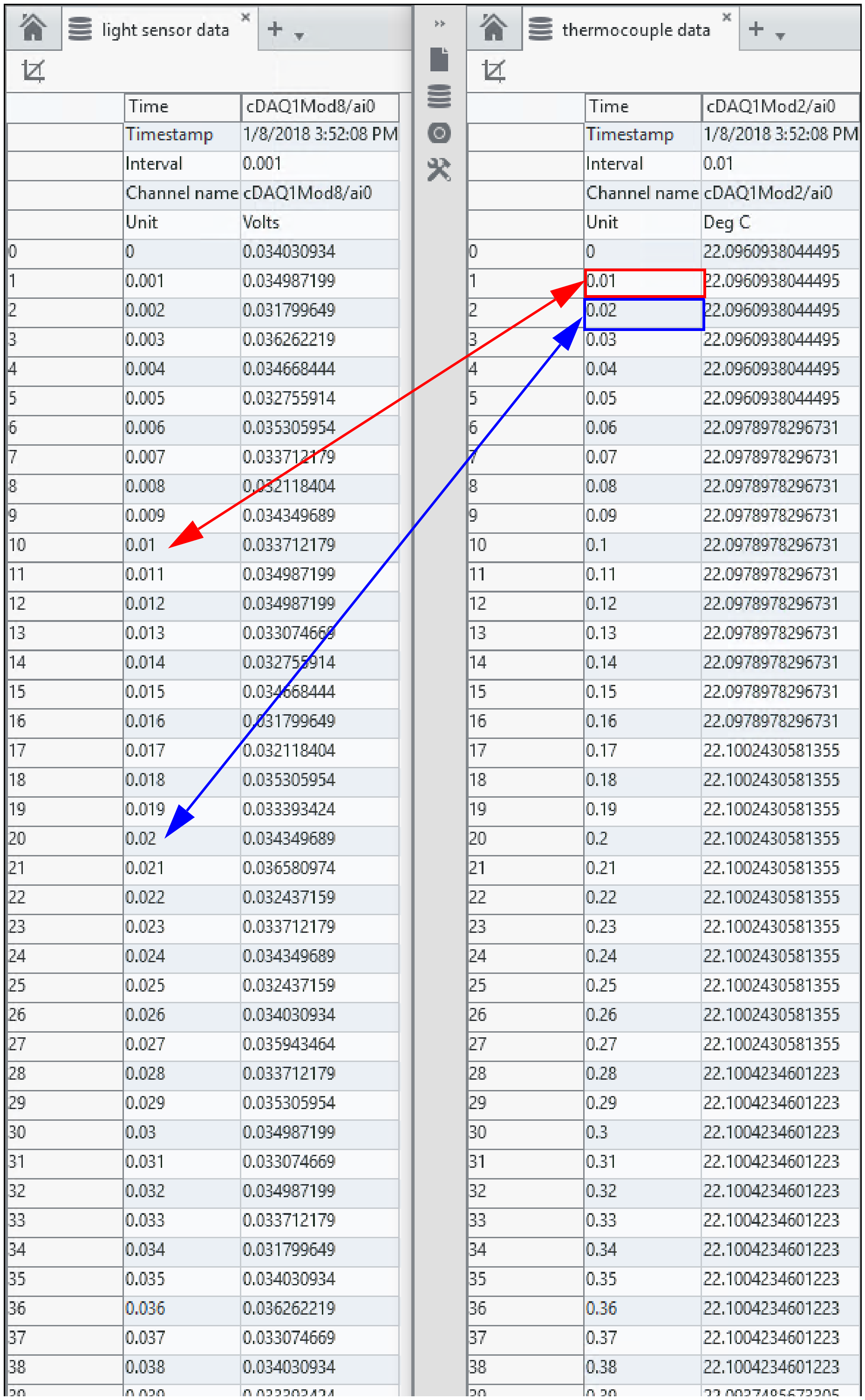
   b. Right-click the **thermocouple data** graph and select **Capture data.**

9. Compare the data in the spreadsheet.

   a. How many data points are in the light sensor data snapshot? _______________

   b. How many data point are in the thermocouple data snapshot?______________

Notice that the light sensor data column has nine data points between each temperature data point, as shown below.



	Time	cDAQ1Mod8/ai0
	Timestamp	1/8/2018 3:52:08 PM
	Interval	0.001
	Channel name	cDAQ1Mod8/ai0
	Unit	Volts
0	0	0.034030934
1	0.001	0.034987199
2	0.002	0.031799649
3	0.003	0.036262219
4	0.004	0.034668444
5	0.005	0.032755914
6	0.006	0.035305954
7	0.007	0.033712179
8	0.008	0.032118404
9	0.009	0.034349689
10	0.01	0.033712179
11	0.011	0.034987199
12	0.012	0.034987199
13	0.013	0.033074669
14	0.014	0.032755914
15	0.015	0.034668444
16	0.016	0.031799649
17	0.017	0.032118404
18	0.018	0.035305954
19	0.019	0.033393424
20	0.02	0.034349689
21	0.021	0.036580974
22	0.022	0.032437159
23	0.023	0.033712179
24	0.024	0.034349689
25	0.025	0.032437159
26	0.026	0.034030934
27	0.027	0.035943464
28	0.028	0.033712179
29	0.029	0.035305954
30	0.03	0.034987199
31	0.031	0.033074669
32	0.032	0.034987199
33	0.033	0.033712179
34	0.034	0.031799649
35	0.035	0.034030934
36	0.036	0.036262219
37	0.037	0.033074669
38	0.038	0.034030934

	Time	cDAQ1Mod2/ai0
	Timestamp	1/8/2018 3:52:08 PM
	Interval	0.01
	Channel name	cDAQ1Mod2/ai0
	Unit	Deg C
0	0	22.0960938044495
1	0.01	22.0960938044495
2	0.02	22.0960938044495
3	0.03	22.0960938044495
4	0.04	22.0960938044495
5	0.05	22.0960938044495
6	0.06	22.0978978296731
7	0.07	22.0978978296731
8	0.08	22.0978978296731
9	0.09	22.0978978296731
10	0.1	22.0978978296731
11	0.11	22.0978978296731
12	0.12	22.0978978296731
13	0.13	22.0978978296731
14	0.14	22.0978978296731
15	0.15	22.0978978296731
16	0.16	22.0978978296731
17	0.17	22.1002430581355
18	0.18	22.1002430581355
19	0.19	22.1002430581355
20	0.2	22.1002430581355
21	0.21	22.1002430581355
22	0.22	22.1002430581355
23	0.23	22.1002430581355
24	0.24	22.1002430581355
25	0.25	22.1002430581355
26	0.26	22.1002430581355
27	0.27	22.1002430581355
28	0.28	22.1004234601223
29	0.29	22.1004234601223
30	0.3	22.1004234601223
31	0.31	22.1004234601223
32	0.32	22.1004234601223
33	0.33	22.1004234601223
34	0.34	22.1004234601223
35	0.35	22.1004234601223
36	0.36	22.1004234601223
37	0.37	22.1004234601223
38	0.38	22.1004234601223

10. Save and close the project when you are done exploring.

End of Exercise 8-3

9 Logging Measurement Data to Disk

Topics

+ Resources

Exercises

Exercise 9-1 Streaming Fan Acceleration Data to Disk

Resources

Resources Refer to the following resource for more information about topics in this lesson. For locations with info codes, go to `ni.com/info` and enter the code to access the article.

TDMS File I/O VIs vs. DAQmx TDMS VIs

Article	Location
The NI TDMS File Format	Info code: `tdmsfileformat`

Customize Logging Behavior With Property Nodes

Article	Location
DAQmx Write Properties	Info code: `daqmxwriteprop`
DAQmx Read Properties	Info code: `daqmxreadprop`

Use TDMS Metadata to Avoid File Format Pitfalls

Article	Location
Writing Data-Management-Ready TDMS Files	Info code: `datamanfiles`

Offline Processing

Article	Location
TDM Excel Add-In for Microsoft Excel Download	Info code: `exceladdin`
Moving Beyond Microsoft Excel for Measurement Data Analysis and Reporting	Info code: `datareport`

Review: Data Best Practices for DAQ

Article	Location
Best Practices for Saving Measurement Data	Info code: `loggingdata`

Exercise 9-1 Streaming Fan Acceleration Data to Disk

Goal

Open an existing data acquisition VI and modify it to write measurement data to a TDMS file.

Scenario

Earlier in this course, you created a VI that read accelerometer data from a fan to assist in determining whether or not the fan was damaged. The end-user has provided feedback that they would like to be able to view that data independently of the application.

In addition to the numeric data, they would like the file to also include header information to provide context for that data.

Design

You decide to use the TDMS file format so that you can stream at high speeds and use the benefits of the built-in file and channel attributes that come with that format.

Implementation

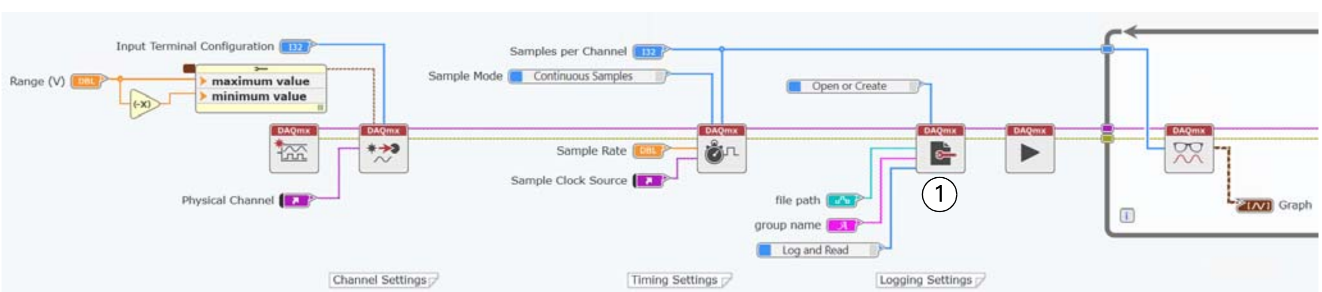
1. Open `Read Fan Acceleration.lvproject`.

   - ☐ If you completed Exercise 5-3, navigate to `<Exercises>\DAQNXG\Continuous Input\` and open `Read Fan Acceleration.lvproject`.

   - ☐ If you did not complete Exercise 5-3, navigate to `<Solutions>\DAQNXG\Exercise 5-3\` and open `Read Fan Acceleration.lvproject`.

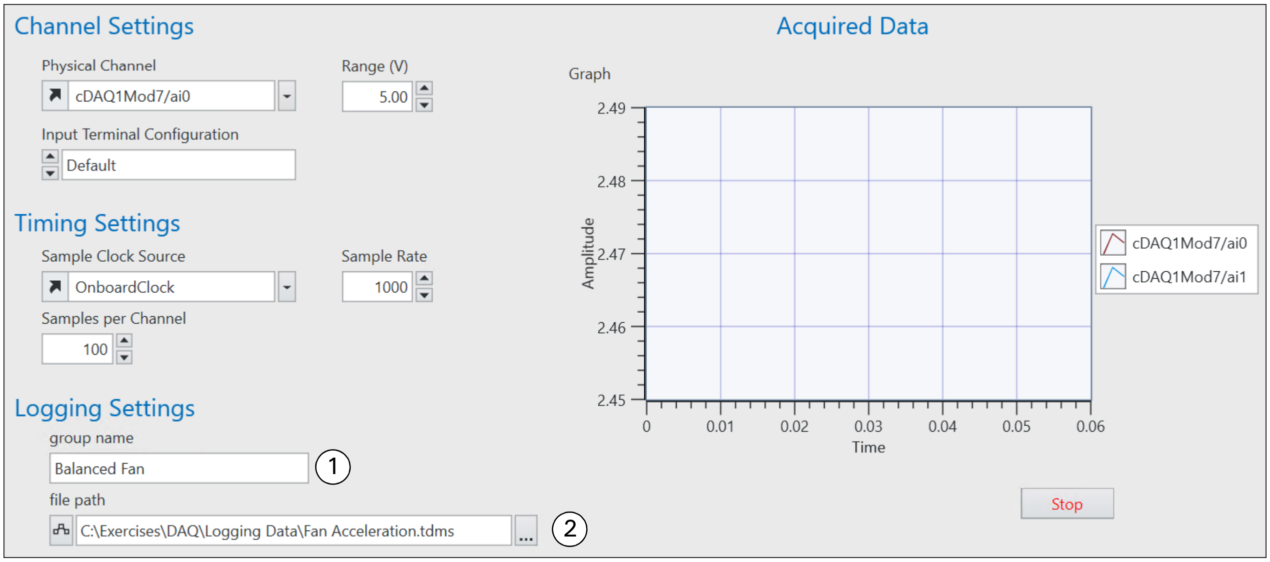
2. Duplicate and rename `Read Fan Acceleration.gvi` to `Read and Log Fan Acceleration.gvi`.

3. Modify the diagram to stream the acquired data to a TDMS file.



1 **DAQmx Configure Logging VI**—This VI modifies the NI-DAQmx task to write acquired data directly to the specified TDMS File. Set the logging mode to **Log and Read** so that you can read data from the file in this same VI. Group Name creates a common grouping in the TDMS file for both acceleration channels.

4. Modify the panel to group the timing controls together and set the control values.



1	Set **Group Name** to `Balanced Fan`. The Group Name determines how the data will be grouped within the TDMS file.
2	Set **File Path** to `<Exercises>\DAQNXG\Logging Data\Fan Acceleration.tdms`.

5. Save the VI.

Test

1. Acquire and log data for a balanced fan.

   ☐ Configure the Sound and Vibration Signal Simulator.

      – Set the Fan Speed Control to DIAL.

      – Toggle the switch to acquire data for the BALANCED FAN.

      – Turn the Fan Speed Control Knob clockwise to the maximum speed for the fan.

   ☐ Run the VI.

   ☐ Acquire approximately five seconds of data and click **Stop**.

2. Review the data acquired and logged to the TDMS file.

   ☐ On the Captured Data tab of the Navigation pane, import `<Exercises>\DAQNXG\Logging Data\Fan Acceleration.tdms`.

   ☐ Note that the data set name is Balanced Fan. This is based on the name of the group.

   ☐ Switch between Table, Graph, and Statistics views to view the data in different ways. You can see the data read from each channel during this acquisition.

3. Acquire and log data for an unbalanced fan.
   - ☐ Configure the Sound and Vibration Signal Simulator.
     – Toggle the switch to acquire data for the UNBALANCED FAN.
   - ☐ On the panel, change the group name to `Unbalanced Fan`.
   - ☐ Run the VI.
   - ☐ Acquire approximately five seconds of data and click **Stop**.
4. Review the data acquired and logged to the TDMS file.
   - ☐ On the Captured Data tab of the Navigation pane, import `<Exercises>\DAQNXG\Logging Data\Fan Acceleration.tdms`.
   - ☐ Note that two data sets are imported: Balanced Fan and Unbalanced Fan. These names are based on the name of the group.
   - ☐ Switch between Table, Graph, and Statistics views to view the data in different ways. You can see the data read from each channel during this acquisition.

Challenge

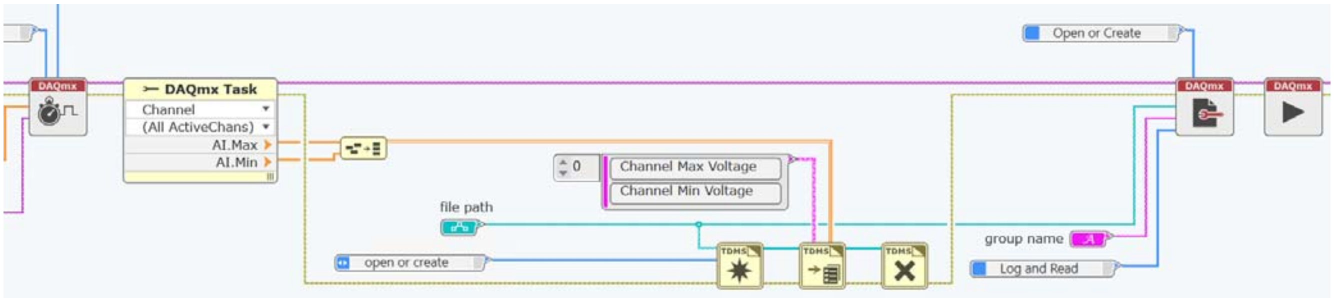
Modify the Read and Log Fan Acceleration VI to write additional property data to the TDMS file.

Tips:

- Use a property node to programmatically obtain information about a channel or task.
- Use the TDMS functions to open, set properties, and close the file.
- You will need to write this property data before you call the NI-DAQmx Configure Logging VI, because that function reserves the file for the duration of the acquisition.

Answers from page 9-6

Below is a diagram of a potential solution to the Challenge problem.



Because you write the channel properties, AI.Max and AI.Min, to the TDMS files, you can find that information in Excel when you import the TDMS file.

1

A	B	C	D	E	F	G	H	I
Root Name	**Title**	**Author**	**Date/Time**	**Groups**	**Description**	*Channel_Max_Voltage*	*Channel_Min_Voltage*	
Fan Acceleration with Att				1		5	-5	
Group	**Channels**	**Description**						
Balanced Fan	1							
Balanced Fan								
Channel	**Datatype**	**Unit**	**Length**	**Minimum**	**Maximum**	**Description**	*NI_ChannelName*	*NI_UnitDescription*
cDAQ1Mod7/ai0	DT_DOUBLE	Volts	7500				cDAQ1Mod7/ai0	Volts

1 Channel Max Voltage and Channel Min Voltage were specified in the **property names** input of the TDMS Set Parameters node.

End of Exercise 9-1

10 System Considerations

Topics

+ Resources

Resources

Resources Refer to the following resource for more information about topics in this lesson. For locations with info codes, go to `ni.com/info` and enter the code to access the article.

Bus Comparison

Article	Location
How to Choose the Right Bus	Info code: `choosebus`

When Do You Need Real-Time OS?

Article	Location
Do I Need a Real-Time System?	Info code: `needrt`

A NI CompactDAQ Measurements Demo Box Information

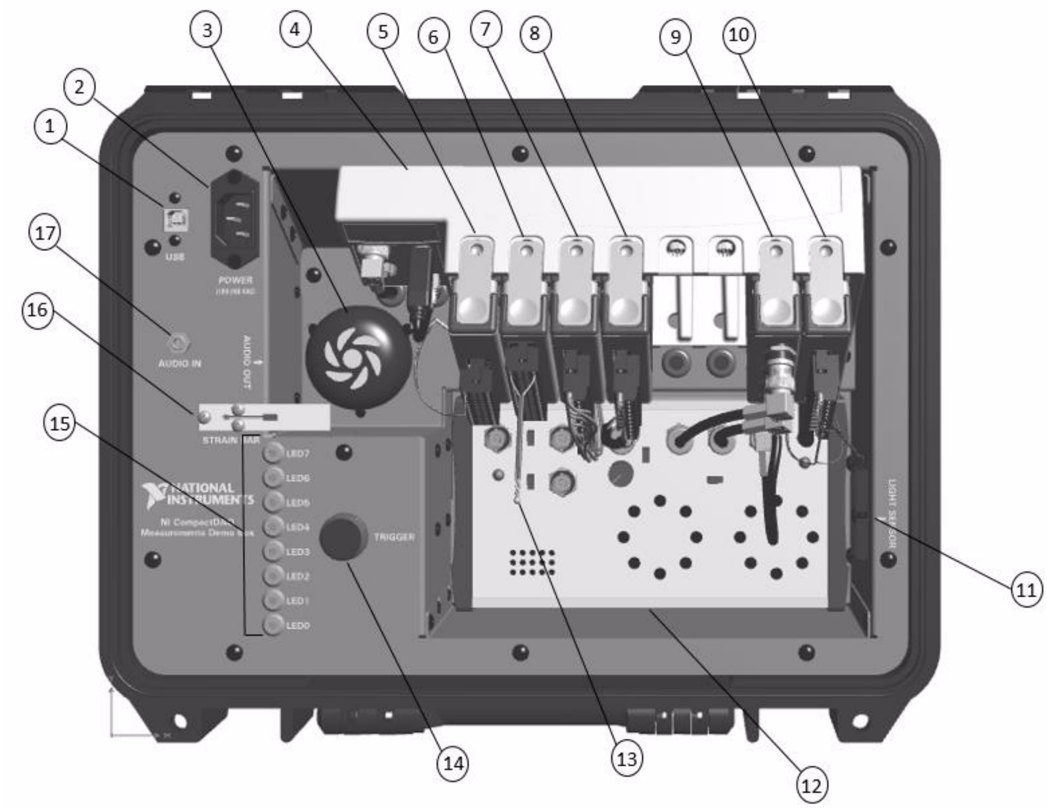
This section information about the components and wiring of the NI CompactDAQ Measurements Demo Box used in this course.

Topics

A. NI CompactDAQ Measurements Demo Box Overview

The following figure and table describe the demo box contents and connections.

Figure A-1. Overview of CompactDAQ Measurements Demo Box Components



1	USB Input	7	NI 9472 (Digital I/O)	12	Sound and Vibration Signal Simulator Box
2	Demo Kit Power	8	NI 9263 (Analog Output)	13	J-Type Thermocouple
3	Speaker	9	NI 9234 (Analog Input—IEPE)	14	Trigger Button
4	NI cDAQ-9178 Chassis	10	NI 9215 (Analog Input—Voltage)	15	LEDs (0-7)
5	NI 9236 (Analog Input—Strain)	11	Solar Panel	16	Strain Gauge Bar
6	NI 9213 (Analog Input—Temperature)			17	Audio Input Port

Table A-1. Connections in the NI CompactDAQ Measurements Demo Box

Module	Channel	I/O
cDAQ-9178	PFI 0	Trigger
	USB	Kit USB Port
	VDC	Power
NI 9213	AI 0	J-Type Thermocouple
NI 9215	AI 0	Solar Panel/Light sensor
	AI 1	Audio In Aux Port
NI 9234	Ch 0	X Accelerometer
	Ch 1	Y Accelerometer
	Ch 2	Tachometer Out
NI 9236	AI 0	Strain bar
NI 9263	AO 0	Speaker
	AO 1	Fan Speed Control
NI 9472	Port 0	LEDs 0:7

B. Light Sensor Information

The light sensor on the right side of the demo box is connected to the NI 9215 module using one red and one black wire.

Solar Panel Description and Specification

The following information is provided by the manufacturer.

Type/Brand	Solar Mini-Panel/Solar Made
Voltage	0.5 V
Current	100 mA
Dimensions	1.75 inches x 1 inch
Weight	1 oz

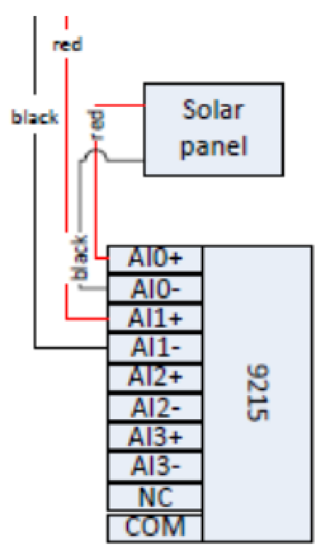
Encapsulated mini-panels utilize solar cells similar to those that power our satellites in space. Mini-panels allow for handling without the normal breakage and cell damage associated with fragile solar cells. Available in thirteen outputs with four case sizes to fit your every need. 6" red/black lead wires attached. Modules may be connected to create different outputs.

All mini panels are tested VOC (voltage open circuit), ISC (current short circuit), using one full sun condition.

General helpful hints - Most applications require charging batteries (the solar operating voltage must be higher than the battery voltage). Direct powering of devices requires knowing the voltage and current of your device. Science Projects: Light bulbs - use light emitting diodes (LED) which require 3 volts, plus 50mA per LED. Motors - Most important for operating miniature motors is the "current", most other motors will require at least 200mA to 500mA of current with the voltage being less important.

Wiring Diagram for the Solar Panel and the NI 9215

The following figure illustrates how the solar panel is connected to the NI 9215.



C. Thermocouple Information

The thermocouple is attached to the NI 9236 module in the middle of the demo box using one red and one white wire.

Thermocouple Description and Specification

The following information is provided by the manufacturer.

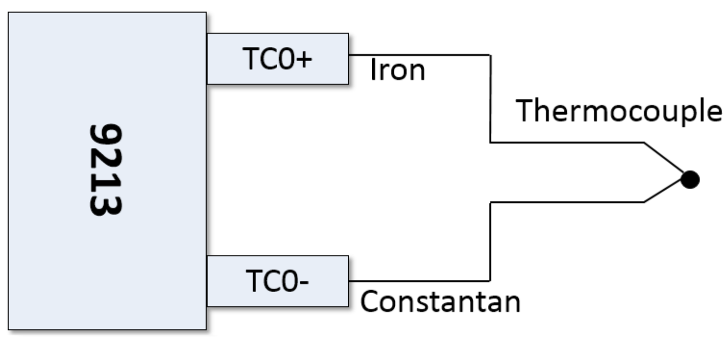
For cost-sensitive applications, NI offers ready-made thermocouples—individual packets of thermocouple wire with the measuring junction provided at one end. Ready-made thermocouples are ideal for starter or educational applications.

J-Type Thermocouples Wire, Fiberglass (32 deg F to 900 deg F) 1 m

	Conductor			Limits of Error
Calibration	**Positive**	**Negative**	**Temp Range**	**(whichever greater)**
J-type	Iron (white)	Constantan (Red)	32 to 900 °F (0 to 482 °C	±4.0°F (2.2 °C) or ±0.75%

Wiring Diagram for the Thermocouple and the NI 9213

The following figure illustrates how the solar panel is connected to the NI 9215.



D. Strain Gage Information

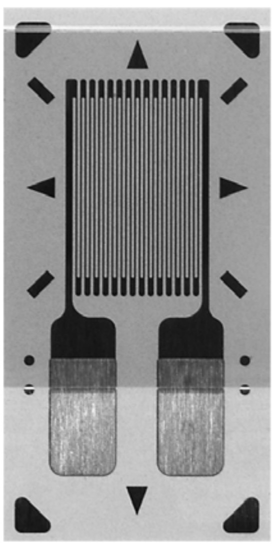
The strain gage is mounted on a metal bar on the left side of the demo box and is connected to the NI 9236 module using one red, one white, and one black wire.

Strain Gage Description and Specification

The following information is provided by the manufacturer. The demo box uses the CEA-XX-125UN-350 general purpose strain gage.

> Universal general-purpose strain gages. Constantan grid completely encapsulated in polyimide, with large, rugged copper-coated tabs. Primarily used for general purpose static and dynamic stress analysis. Nominal gage factor for constantan is 2.10.

GAGE PATTERN DATA



actual size

GAGE DESIGNATION See Note 1	RESISTANCE (OHMS)	OPTIONS AVAILABLE See Note 2
CEA-XX-125UN-120	120 ± 0.3%	**P2**
CEA-XX-125UN-350	350 ± 0.3%	**P2**

DESCRIPTION

General-purpose gage with narrow geometry. Exposed solder tab area 0.06 x 0.05 in [1.5 x 1.1 mm]. See also 125UW pattren.

GAGE DIMENSIONS

Legend: ES = Each Section; S = Section (S1 = Sec 1); CP = Complete Pattern; M = Matrix

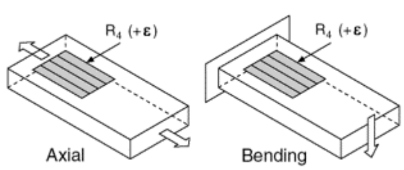
inch / millimeter

Gage Length	Overall Length	Grid Width	Overall Width	Matrix Length	Matrix Width
0.125	0.275	0.100	0.120	0.38	0.19
3.18	6.99	2.54	3.05	9.7	4.8

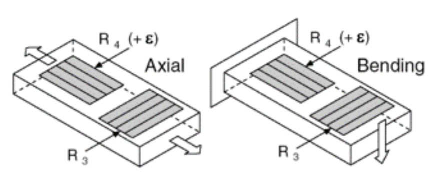
GAGE SERIES DATA See Gage Series data sheet for complete specifications.

Series	Description	Strain Range	Temperature Range
CEA	Universal general-purpose strain gages.	±5%	–100° to +350°F [–75° to +175°C]

Based on the image in the specification, you can see that the strain gage is a Quarter Bridge 1 type.



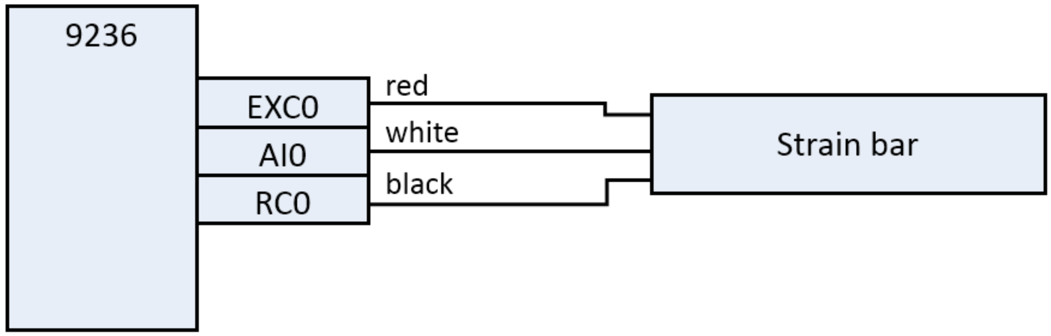
Quarter Bridge Type 1



Quarter Bridge Type 2

Wiring Diagram for the Strain Gage and the NI 9236

The following figure illustrates how the strain gage is connected to the NI 9236.



E. Speaker Information

The speaker is located on the left side of the demo box is connected to the NI 9263 module using one red and one black wire.

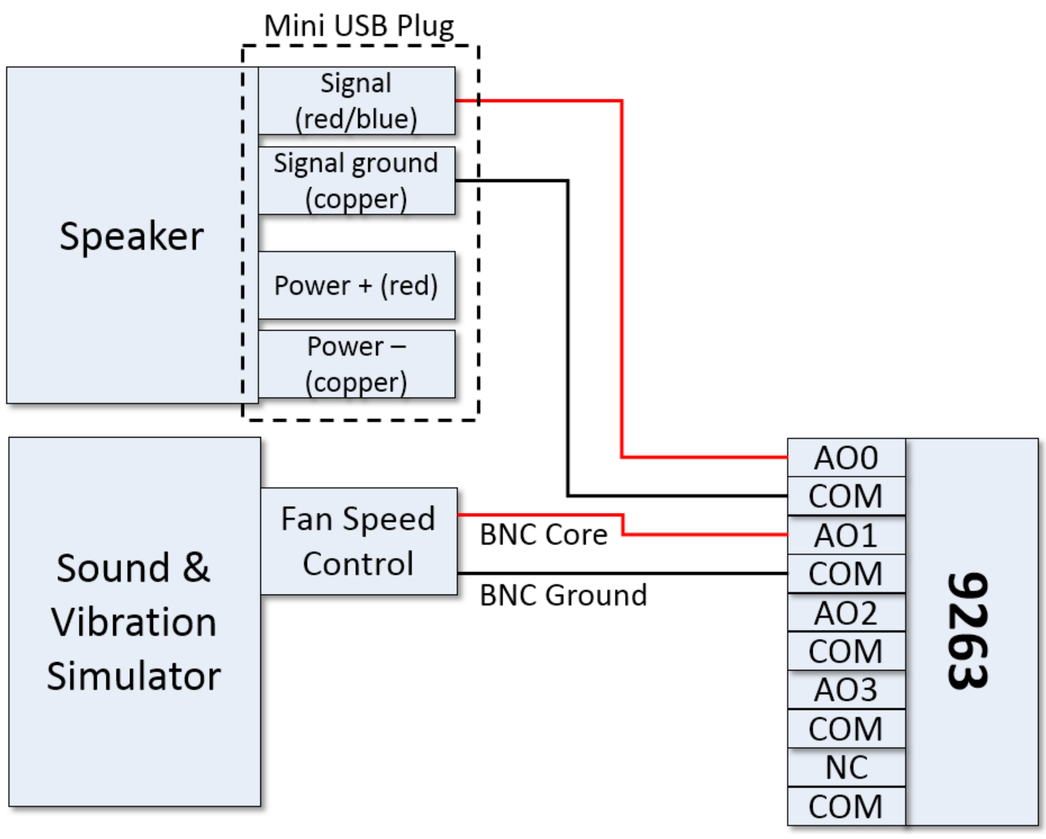
Speaker Description and Specifications

The following information is provided by the manufacturer.

> The speaker measures only 2 1/8" in diameter and about 1 1/2" tall when closed (about 2 1/8" tall when opened).The speaker comes with a combination power/signal cable. Both the audio and power are carried to the speaker through a USB mini-B connection.
>
> Power Output 2.0W
>
> AMP Freq Range .... 280Hz – 16KHz
> Signal to Noise Ratio .... ≥80dB
> Charge voltage .... 5V ± 0.5V

Wiring Diagram for the Speaker and the NI 9263

The following figure illustrates how the NI 9263 module is connected to the Sound and Vibration Signal Simulator box and the speaker.



F. Sound and Vibration Signal Simulator Box Information

The Sound and Vibration Signal Simulator box is located in the front right side of the demo box is connected to the NI 9263 module and the NI 9234 module using wires and BNC connectors.

Sound and Vibration Signal Simulator Box Description and Specifications

The following information is provided by the manufacturer.

The signal simulator box provides a basic rotor kit with two selectable vibration sources, a balanced fan and an unbalanced fan, plus internal 2-axis accelerometers. The signal accessory also includes a tachometer for the fans which can be used in applications requiring order analysis and tracking.

Fan Speed Input (Analog Input)

Expected: .... 0 to 5 VDC

Absolute Rating:........................ ± 10V

Maximum fan speed.................... 6000 RPM

Analog Outputs:

Accelerometer Outputs .... 2.5 V peak, 2.5V DC offset

Tachometer Output:.................... 0 to 5 V square wave, 2 pulses per revolution

Accelerometer Specifications

The following information is provided by the manufacturer.

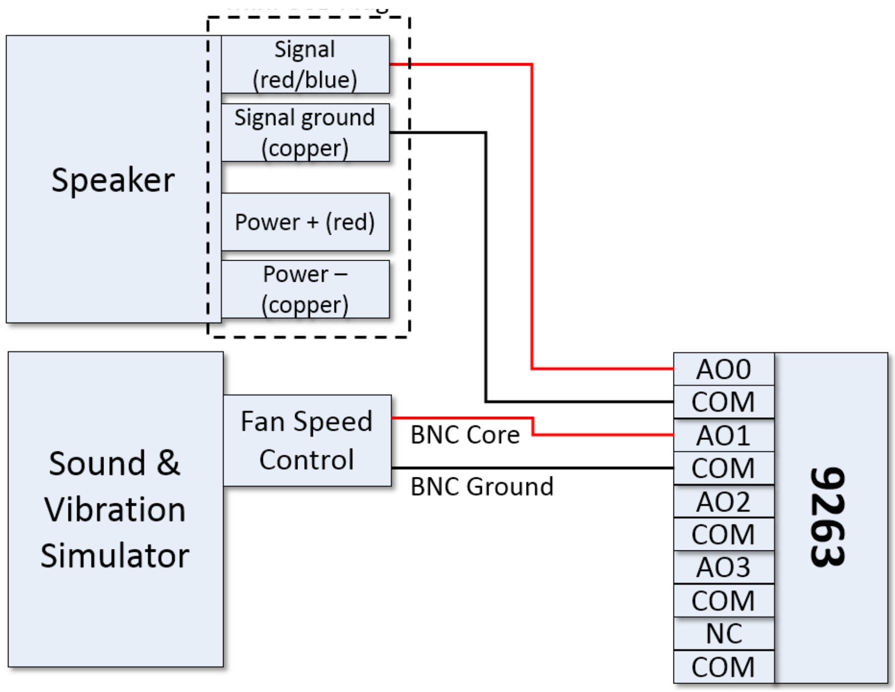
Parameter	Conditions	Min	Typ	Max	Unit
SENSOR INPUT	Each axis				
Measurement Range			±5		*g*
Nonlinearity	% of full scale		±0.2		%
Package Alignment Error			±1		Degrees
Alignment Error	X sensor to Y sensor		±0.1		Degrees
Cross Axis Sensitivity			±2		%
SENSITIVITY (RATIOMETRIC)[2]	Each axis				
Sensitivity at X_{OUT}, Y_{OUT}	$V_S = 3$ V	156	174	192	mV/*g*
Sensitivity Change due to Temperature[3]	$V_S = 3$ V		0.01		%/°C
ZERO *g* BIAS LEVEL (RATIOMETRIC)	Each axis				
0 *g* Voltage at X_{OUT}, Y_{OUT}	$V_S = 3$ V	1.3	1.5	1.7	V
0 *g* Offset Versus Temperature			±0.6		m*g*/°C

Wiring Diagrams for Sound and Vibration Signal Simulator Box and the NI 9263 and NI 9234

The Sound and Vibration Signal Simulator box is wired to both the NI 9263 and the NI 9234.

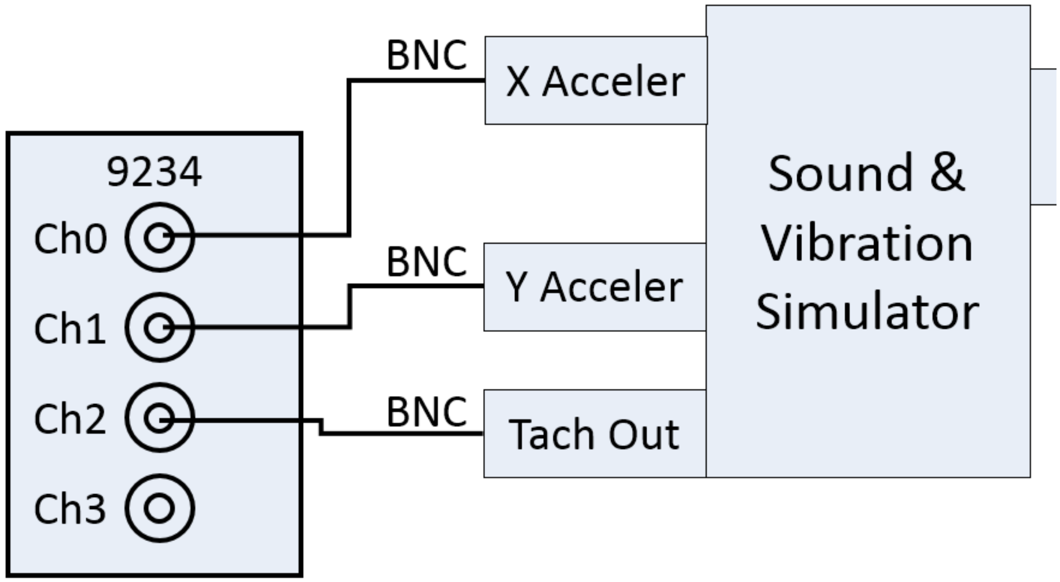
Wiring Diagram to the NI 9263

The following figure illustrates how the NI 9263 module is connected to the Sound and Vibration Signal Simulator and the speaker.



Wiring Diagram to the NI 9234

The following figure illustrates how the NI 9234 module is connected to the Sound and Vibration Signal Simulator. The accelerometer in the Sound and Vibration Signal Simulator connects to channel 0 and channel 1.



G. LEDs Information

The eight LEDs are located on the left side of the demo box and are connected to the NI 9472 module.

Aluminum LED Indicator Light Specifications

The following information is provided by the manufacturer.

> An Indicator Light for flush mounting on a panel. Can be used to indicate that status of a circuit.

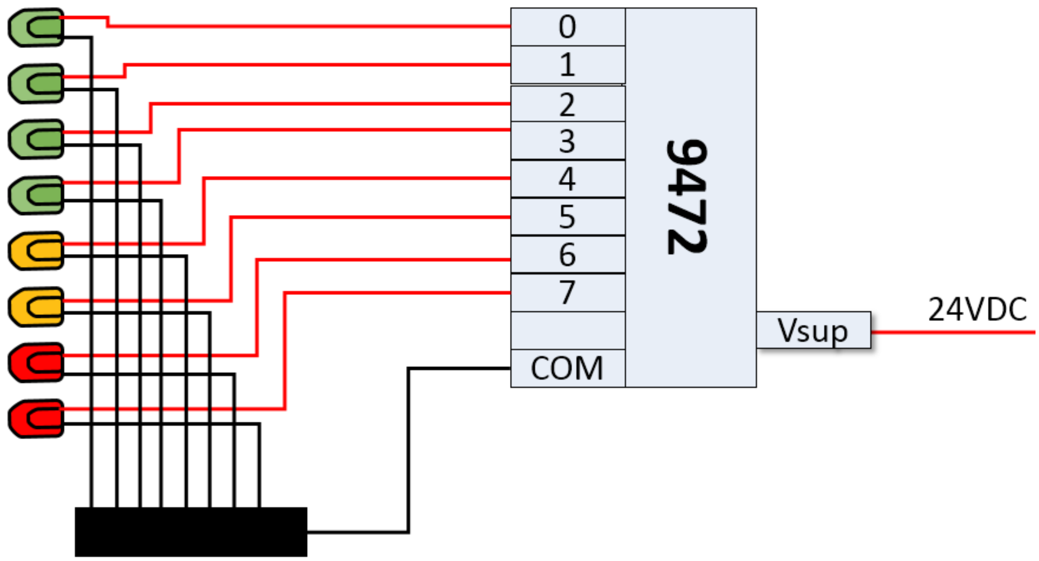
Beam angle	25 degree
Connection Type	2 red/black wires
Dimmable	Yes
IP Rating	IP20
LED Type	3mm Super Bright LED
Polarity Sensitive	Yes
Operating Voltage Range	12 VDC
Wire Gauge	22 AWG
Weight	13 grams
Housing	Aluminum
Diameter	8mm

The demo box uses the following:

Color	CCT(K)Wavelenth(nm)	Lumen(lm)	Current(mA)
Green	525	4	3
Amber	590	3	3
Red	625	3	3

Wiring Diagram to the NI 9472

The following figure illustrates how the 9472 module is connected to the eight LEDs in the demo box.



B Measuring Temperature

Topics

Exercises

Resources

Resources Refer to the following resource for more information about topics in this lesson. For locations with info codes, go to `ni.com/info` and enter the code to access the article.

Article	Location
Overview of Temperature Sensors	Info code: `tempsensors`
Making a Thermocouple Measurement With NI LabVIEW	Info code: `measurethermo`
Temperature Measurements with Thermistors: How-To Guide	Info code: `measguide`
Taking Temperature Measurements with RTDs: How-To Guide	Info code: `temperatureguide`

Exercise B-1 Using NI-DAQmx to Measure Temperature with a Thermocouple

Goal

Open and run an example program to take a software-timed, temperature measurement with the NI 9213 and a thermocouple.

Scenario

In this exercise, you use the specifications of a thermocouple and the NI 9213 C Series module to verify a temperature reading. Instead of building a VI from scratch, you take advantage of a NI-DAQmx shipping example to take a software-timed measurement.

Implementation

Note Make sure that all tasks are stopped before completing this exercise. If you need to reset the device, select the device from the SystemDesigner and click the **Reset** button on the Configuration pane.

1. Refer to *Thermocouple Information* in Appendix A, *NI CompactDAQ Measurements Demo Box Information* to answer questions about the wiring of this sensor.

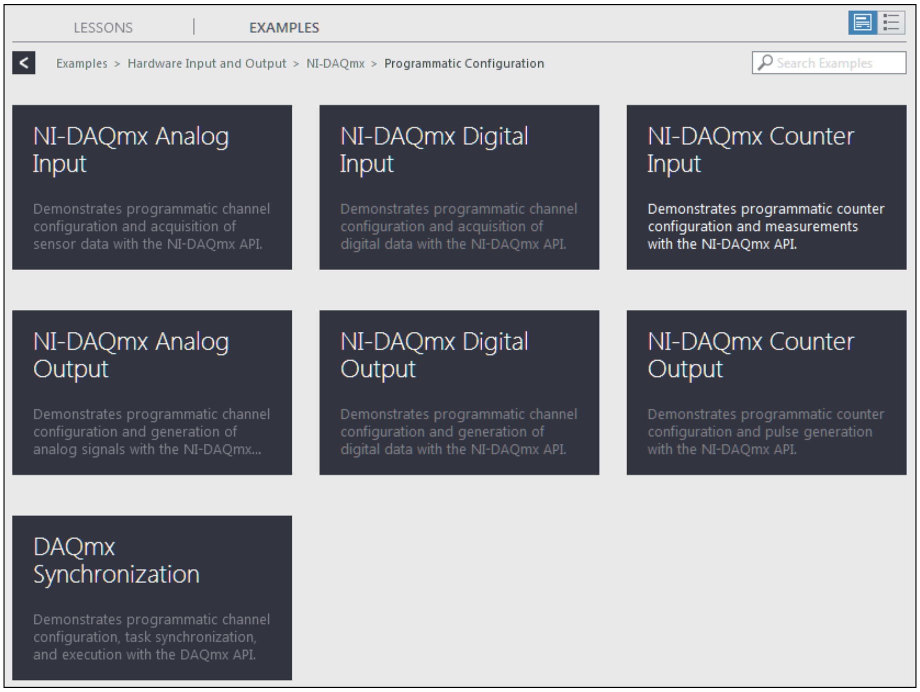
   a. Is the source sensor floating or grounded? ______________________________

   b. What kind of terminal configuration is this wiring setup?

   c. Which device pins are wired to the thermocouple?

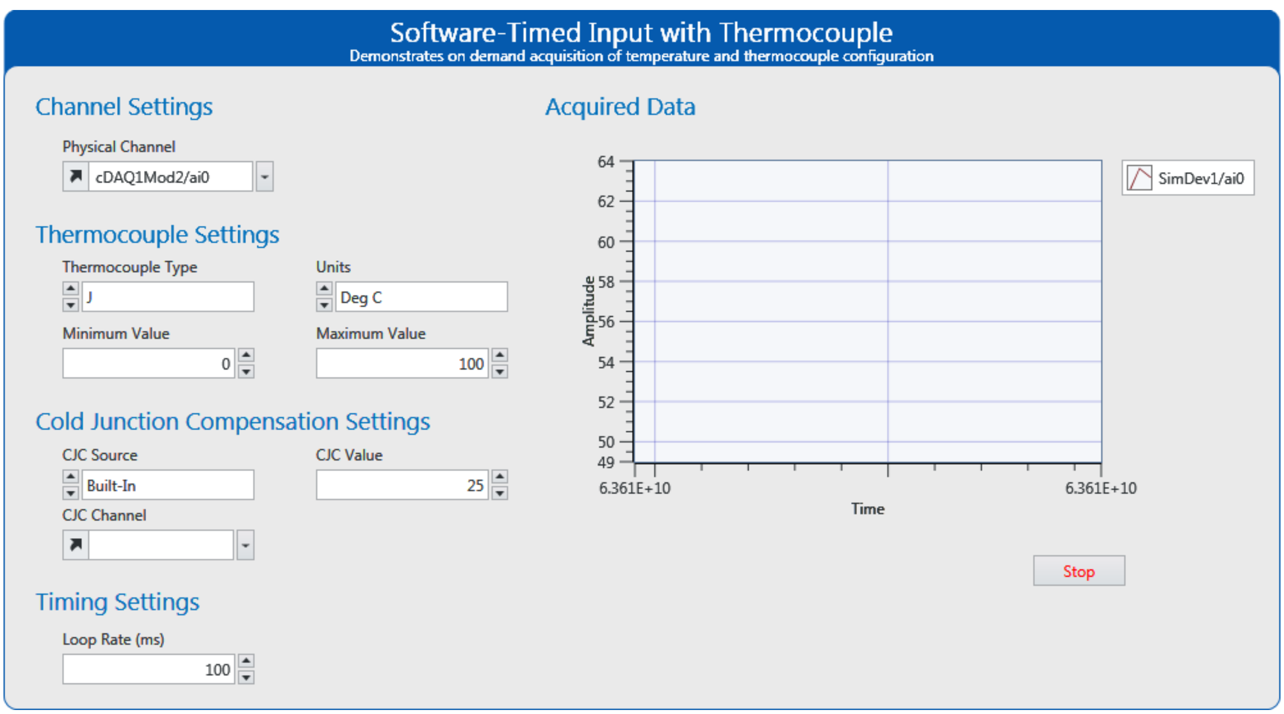
2. On the **Learning** tab, navigate to **Examples»Hardware Input and Output»NI-DAQmx»Programmatic Configuration**.

3. Click **NI-DAQmx Analog Input** and create and project named `Thermocouple Test`.



4. Open `Software-Timed Input with Thermocouple.gvi`.

   Notice that the panel of the VI offers many of the same options that the analog input task configuration pane offered.



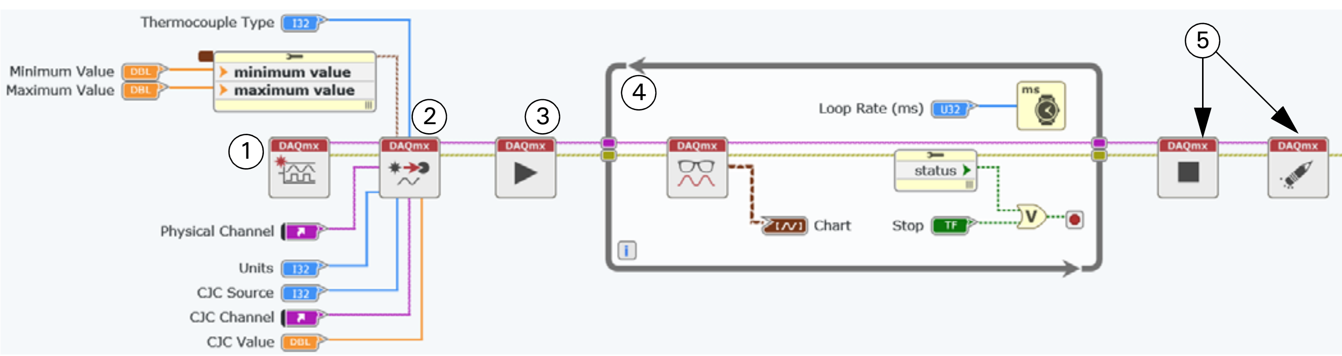
5. Update the panel so that the VI uses the NI 9213 to read a voltage from the thermocouple.

Parameter	Hint	Value
Physical Channel	Which slot of the cDAQ chassis contains the NI 9213 module? What is the answer to step 1c, page B-4?	
Max Temperature, Min Temperature	The thermocouple specification lists the maximum temperature. However, based on the temperature of the room, what max or min values are practical?	
Thermocouple Type	Refer to the *Thermocouple Information* section of Appendix A, *NI CompactDAQ Measurements Demo Box Information*.	
CJC Source, CJC Channel	Refer to the *NI 9213 DataSheet* to determine if the NI 9213 has a built-in cold-junction compensation. If the **CJC Source** is **Channel**, then you must specify the channel.	
CJC Value	If the **CJC source** is **Constant Value** then you must specify the value.	

6. Review the diagram to see how the VI is implemented.

   a. Right-click the **Physical Channel** control and select **Find»On diagram.**

   b. Review the components of the diagram. For a software-timed acquisition application, you see the basic DAQ API structure: Create, Configure, Start, Read/Write, Stop/Clear.



1 **DAQmx Create Task**—This node create a blank task that the DAQmx Create Virtual Channel node fills with details. In a simple application like this, the DAQmx Create Task node is optional.
2 **DAQmx Create Virtual Channel**—Using the Context Help for this VI, you see that only the **Physical Channel** input is required. However, the **Thermocouple Type** is important to specify to get accurate readings. Because this is a simple acquisition, no further configuration is needed.
3 **DAQmx Start Task**—Starting the task means that the device is ready to take a measurement.
4 The While Loop contains the DAQmx Read VI and controls the timing of the acquisition using the Wait (ms) function. This means that the timing of the acquisition is dependent on the OS.
5 **DAQmx Stop Task** VI and **DAQmx Clear Task** VI—This example VI uses both for completeness, but the NI-DAQmx Clear Task VI will stop the task if necessary.

Test

1. Run the VI.
2. Touch the thermocouple with your fingers to make the temperature rise.
3. Stop the VI.
4. Stop and close the VI without saving it.

End of Exercise B-1

C Measuring Sound, Vibration, and Acceleration (IEPE Measurements)

Topics

- Resources

Exercises

Resources

Resources Refer to the following resource for more information about topics in this lesson. For locations with info codes, go to `ni.com/info` and enter the code to access the article.

Article	Location
AC and DC Coupling	Info code: `acdccoupling`
Connecting Accelerometers to a DAQ Device	Info code: `connectaccel`

Exercise C-1 Using NI-DAQmx to Measure Vibration

Goal

Open and run an example program to take a vibration measurement.

Scenario

In this exercise, you use the NI 9234 and the Sound and Vibration Signal Simulator hardware to validate a vibration measurement.

Implementation

Note Make sure that all tasks are stopped before completing this exercise. If you need to reset the device, select the device from the SystemDesigner and click the **Reset** button on the Configuration pane.

1. Refer to in Appendix A, *NI CompactDAQ Measurements Demo Box Information* to answer questions about the wiring of this sensor. (Answers on page C-7.)

   a. Which module pins are wired to the accelerometer in the Sound and Vibration Signal Simulator box? ____________________

   b. What is the minimum and maximum range, in g, that the sensor can read?

      ____

      (Refer to the *Accelerometer Specifications* on page A-10.)

   c. What is the sensitivity of the accelerometer? ____________________

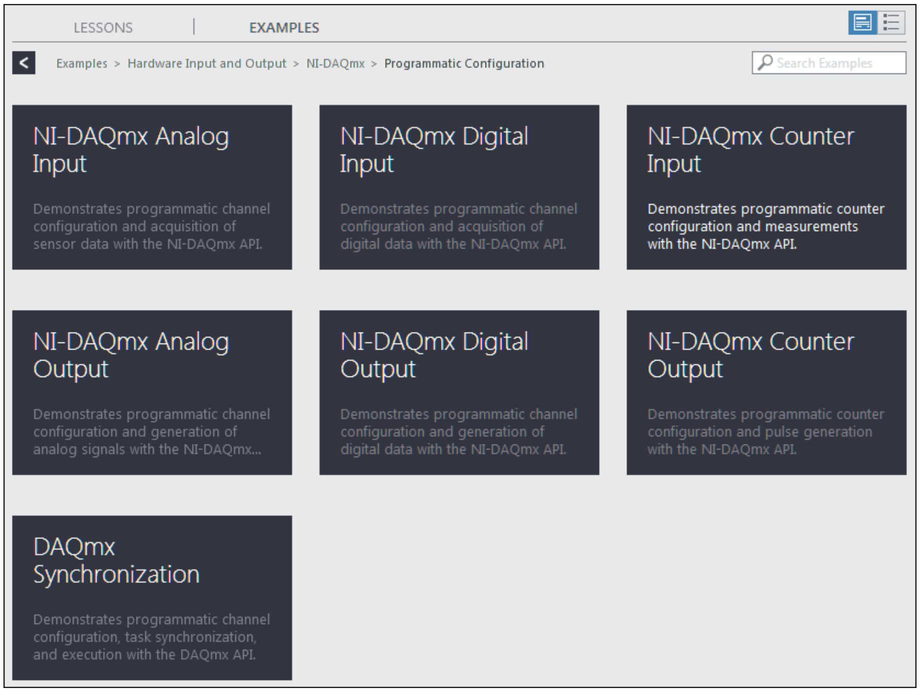
2. Refer to the *NI 9234 Datasheet*, located in `<Exercises>\DAQNXG\_Resources\Hardware Specifications`, to answer questions about the hardware connected to the sensor. (Answers on page C-7.)

   a. Does the device have built-in voltage excitation? ____________________

   b. How much excitation current, in Amps, does the NI 9234 provide? ____________________

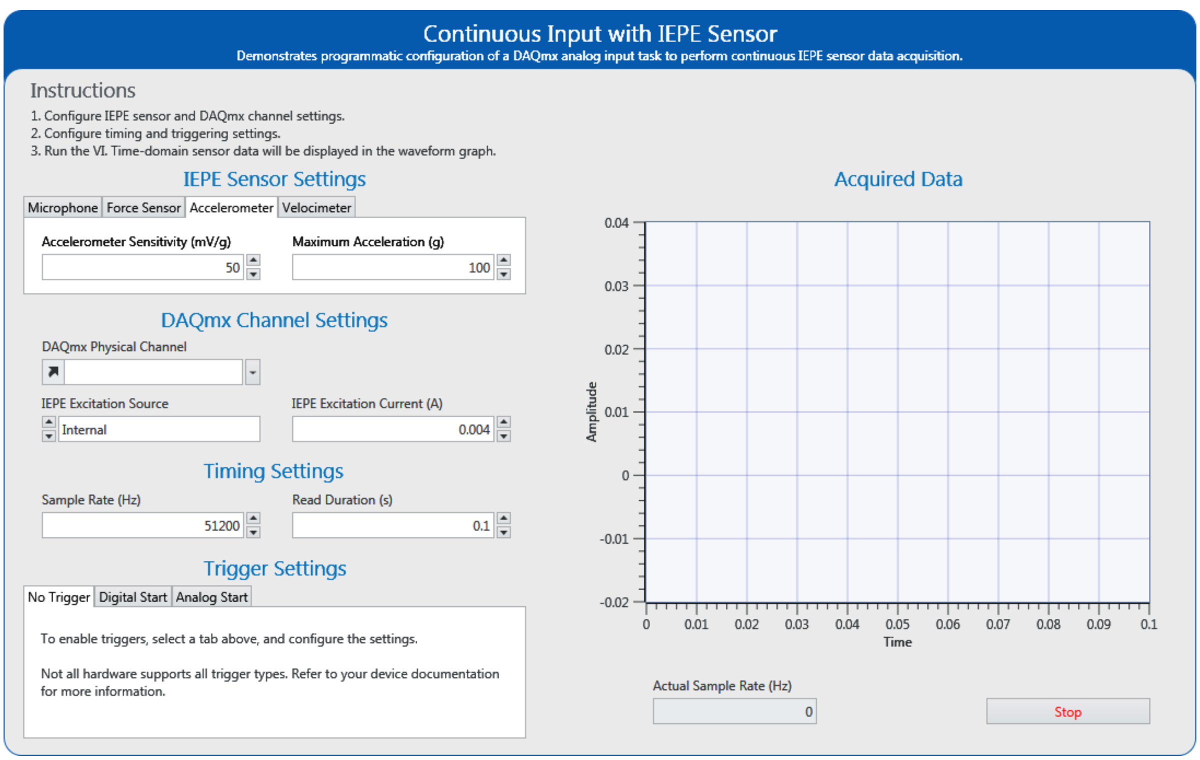
3. On the **Learning** tab, navigate to **Examples»Hardware Input and Output» NI-DAQmx»Programmatic Configuration**.

4. Click **NI-DAQmx Analog Input** and create and project named `Accelerometer Test`.



5. Open `Continuous Input with IEPE Sensor.gvi`.

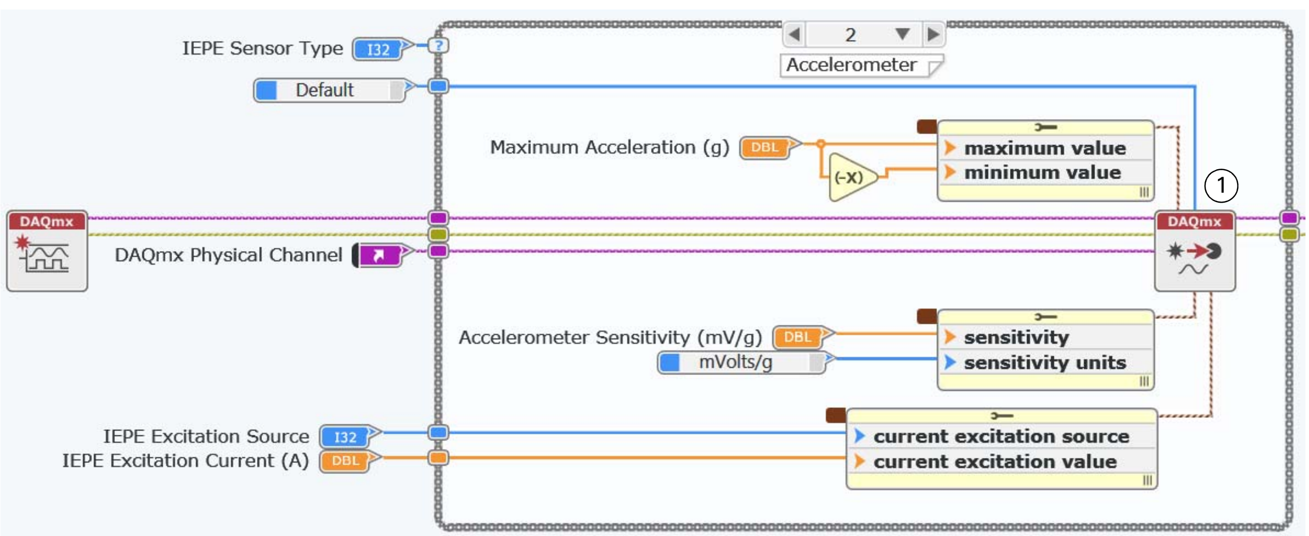
   Notice that the panel of the VI offers many of the same options that the analog input task configuration pane offered.



6. Update the panel so that the VI uses the following values.

Parameter	Hint	Value
Physical Channel	Which slot of the cDAQ chassis contains the NI 9234 module? What is the answer to step 1a, page C-4?	
Terminal Configuration	Which configuration works with either source reference?	
IEPE Excitation Source	Refer to the *NI 9234 Datasheet* or your answers to questions in step 2, page C-4.	
IEPE Current Value (A)	Refer to the *NI 9234 Datasheet*.	
Max Acceleration and **Min Acceleration**	Refer to the accelerometer specs or your answers to questions in step 1, page C-4.	
Sensitivity	Refer to the accelerometer specs or your answers to questions in step 1, page C-4.	

7. Review the diagram to see how the VI is implemented.



1 **DAQmx Create Virtual Channel**—Using the Context Help for this node, you see that only the **physical channel** input is required.

Test

1. Run the VI.

2. Tap the side of the demo box and see how the signal changes.

Tip You might want to turn off the Y-axis autoscale to see the signal better.

3. Stop the VI.
4. Change the **Physical Channel** input to a different channel on the NI 9234.
5. Run the VI and note any changes.
6. Stop and close the VI without saving it.

Answers from page **C-4**.

1. Refer to in Appendix A, *NI CompactDAQ Measurements Demo Box Information* to answer questions about the wiring of this sensor.
   a. Which device pins are wired to the accelerometer in the Sound and Vibration Signal Simulator box? **AI0 and AI1**
   b. What is the minimum and maximum range, in g, that the sensor can read? **-5g to +5 g**
   c. What is the sensitivity of the accelerometer? **174 mV/g**
2. Refer to the *NI 9234 Datasheet*, located in `<Exercises>\DAQNXG\_Resources\Hardware Specifications`, to answer questions about the hardware connected to the sensor.
   a. Does the device have built-in voltage excitation? **Yes**
   b. How much excitation current, in Amps, does the NI 9234 provide? **0.0021 Amps (typical)**

End of Exercise C-1

D Measuring Strain, Force, or Pressure (Bridge-Based Measurements)

Topics

Exercises

Resources

Resources Refer to the following resource for more information about topics in this lesson. For locations with info codes, go to `ni.com/info` and enter the code to access the article.

Article	Location
Measuring Strain with Strain Gages	Info code: `measurestrain`

Article	Location
Load, Pressure, and Torque Measurements: How-To Guide	Info code: `pressurehowto`

Article	Location
Connecting Strain Gages to a DAQ Device	Info code: `connectstrain`

Article	Location
Using Connection Diagrams for NI-DAQmx Tasks	Info code: `connectiontasks`

Exercise D-1 Using NI-DAQmx to Measure Strain

Goal

Open and run an example program to take a software-timed, strain measurement with the NI 9236 and a strain gage.

Scenario

In this exercise, you use the specifications of a strain gage and the NI 9236 C Series module to verify a strain reading. Instead of building a VI from scratch, you take advantage of an NI-DAQmx shipping example to take a software-timed measurement.

Implementation

Note Make sure that all tasks are stopped before completing this exercise. If you need to reset the device, select the device from the SystemDesigner and click the **Reset** button on the Configuration pane.

1. Refer to Appendix A, *NI CompactDAQ Measurements Demo Box Information* to answer questions about the connection of this sensor. (Answers on page D-9.)

   a. Which device pins are wired to the strain gage?____________________________ (Refer to the *Wiring Diagram for the Strain Gage and the NI 9236* section in Appendix A.)

   b. Which bridge configuration type is the strain gage installed in the demo box?

   ____

   c. What is the gage resistance, in ohms?______________________________________

   d. What is the gage factor?____

2. Refer to the *NI 9236 Datasheet*, located in `<Exercises>\DAQNXG\_Resources\Hardware Specifications`, to answer questions about the hardware connected to the sensor. (Answers on page D-9.

   a. Does the device have built-in voltage excitation?____________________________

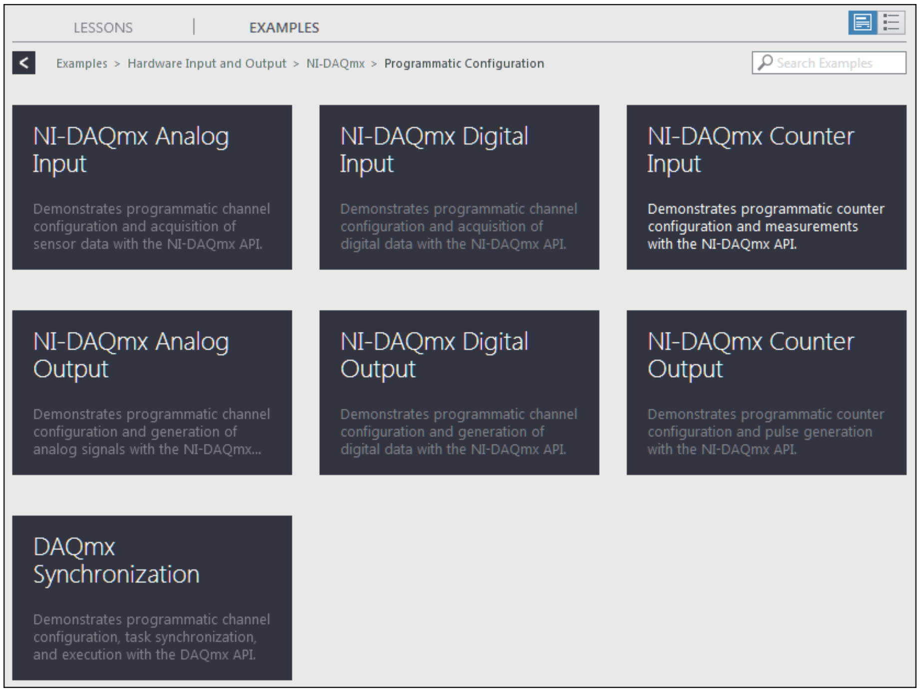
   b. How much bridge excitation, in volts, does the NI 9236 provide?_____________

   c. What is the minimum and maximum values, in volts, that the device can read?

   ____

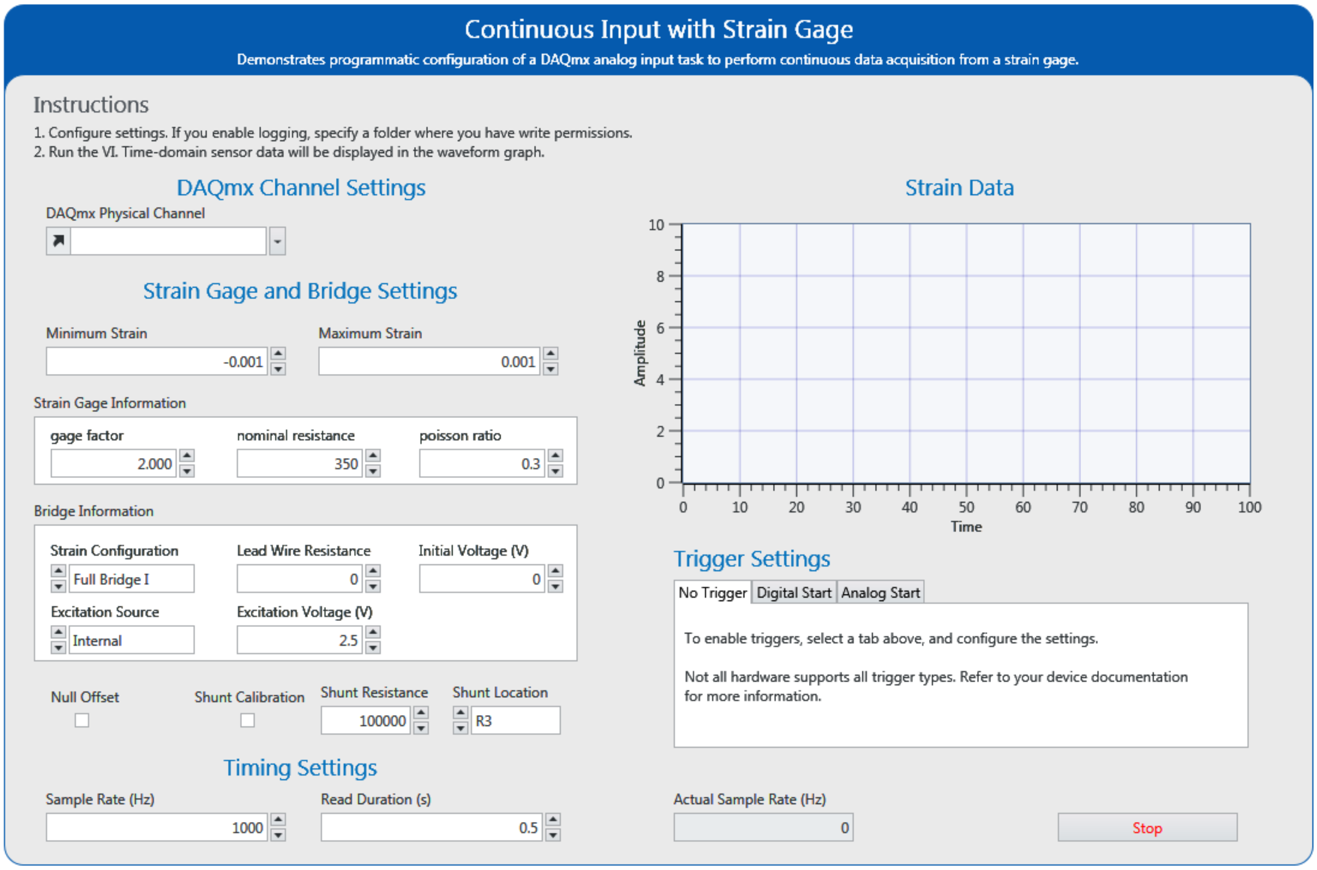
3. On the **Learning** tab, navigate to **Examples»Hardware Input and Output» NI-DAQmx»Programmatic Configuration**.

4. Click **NI-DAQmx Analog Input** and create and project named `Strain Test`.



5. Open `Continuous Input with Strain Gage.gvi`.

   Notice that the panel of the VI offers many of the same options that the analog input task configuration pane offered.

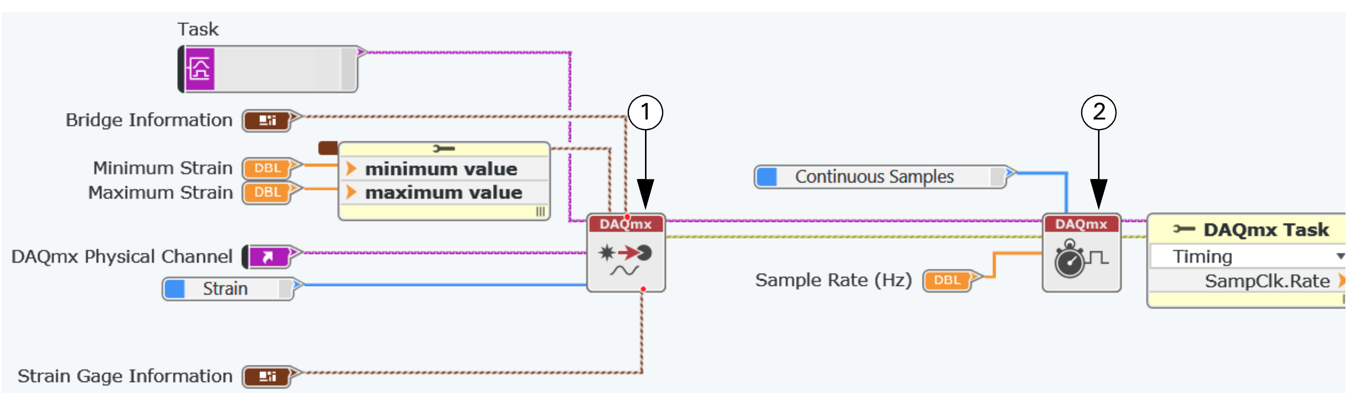


6. Update the panel so that the VI uses the NI 9236 to read a voltage from the strain gage.

Parameter	Hint	Value
DAQmx Physical Channel	Which slot of the cDAQ chassis contains the NI 9213 module? What is the answer to step 1a, page D-4?	
Minimum Strain and **Maximum Strain**	What are the answers to step 2, page D-4?	
Gage Factor	The gage factor for metallic strain gages is usually around 2. The sensor documentation may have more information.	
Nominal Gage Resistance	What is the answer to step 1c, page D-4?	
Poisson Ratio	The Poisson ratio for Constantan alloys ranges between .34 and .35.	.34
Strain Configuration	What is the answer to step 1b, page D-4?	
Lead Wire Resistance and **Initial Bridge Voltage**	For now, assume that both of these are nominal.	0
Excitation Source and **Excitation Voltage**	What are the answers to step 2, page D-4?	

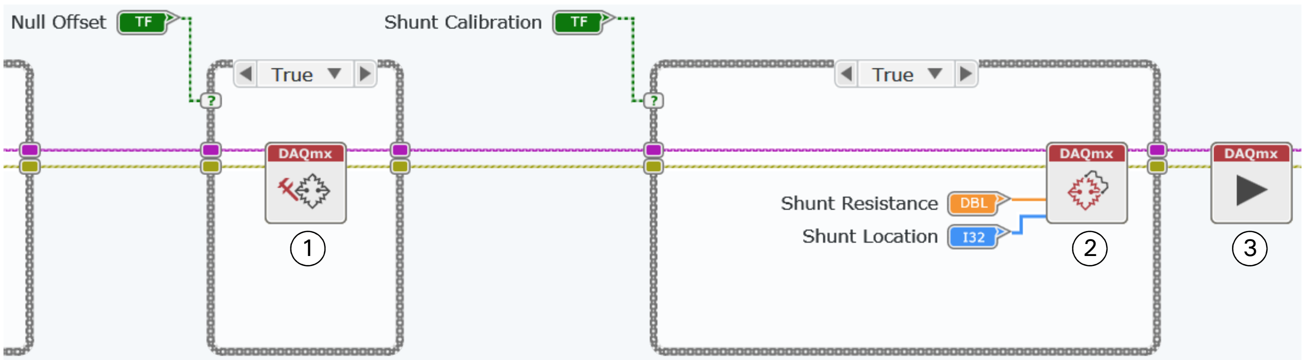
7. Review the components of the diagram.

   a. For a hardware-timed continuous acquisition application you begin on the left of the diagram with a basic DAQ API structure: create a channel and configure timing.



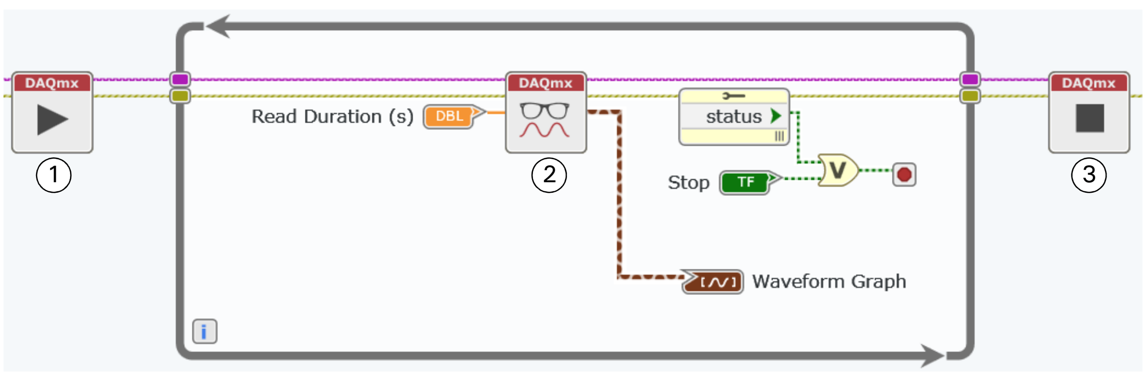
1	**DAQmx Create Virtual Channel**—Using the Context Help for this node, you see that only the **Physical Channel** input is required. However, **Bridge Information** and **Strain Gage Information** is important to specify to get accurate readings.
2	**DAQmx Timing node**—This node is necessary when you want to specify hardware-timed signal acquisition.

b. For strain gage applications, you might also need to configure additional signal conditioning such as offset nulling or shunt calibration.



1 **DAQmx Perform Bridge Offset Nulling Calibration**—Performs a bridge offset nulling calibration on the channels in the task.
2 **DAQmx Perform Shunt Calibration Ex**—Performs a shunt calibration on the channels in the task.
3 **DAQmx Start Task**—Signals the completion of task configuration and the readiness to start the measurement.

c. On the right of the diagram you see the basic DAQ API structure: Start, Read/Write, Stop/Clear.



1 **DAQmx Start Task**—Starting the task means that the device is ready to take a measurement.
2 The While Loop contains the DAQmx Read node. The **duration** input specifies the amount of time in seconds to read samples. For example, if the **duration** is 1 second, the loop will execute once per second.
3 **DAQmx Stop Task** and **DAQmx Clear Task** node—This example VI uses both for completeness, but the DAQmx Clear Task node will stop the task if necessary.

Test

1. Run the VI.
2. What is the value generated by the gage before you press the strain bar?________
3. Press the strain bar and make sure that the VI registers a change on the Strain Data graph.
4. What is the maximum value you can generate? What is the smallest change you can generate by pressing on the bar?________________________________
5. Stop and close the VI without saving it.

Answers from page D-4.

1. Refer to Appendix A, *NI CompactDAQ Measurements Demo Box Information* to answer questions about the connection of this sensor.

   a. Which device pins are wired to the strain gage?

Pin Number	Channel Name
Pin 1	**EXC0**
Pin 2	**AI0**
Pin 3	**RC0**

   b. Which bridge configuration type is the strain gage installed in the demo box? **Quarter Bridge I**

   c. What is the gage resistance, in ohms? **350**

   d. What is the gage factor? **2.1**

2. Refer to the *NI 9236 Datasheet*, located in `<Exercises>\DAQNXG\_Resources\Hardware Specifications`, to answer questions about the hardware connected to the sensor.

   a. Does the device have built-in voltage excitation? **Yes**

   b. How much bridge excitation, in volts, does the NI 9236 provide? **3.3 V**

   c. What is the minimum and maximum values, in volts, that the device can read? **-.0294 V and .0294 V**

End of Exercise D-1

E Measuring Position with Encoders (Counter Input)

Topics

+ Resources

Exercises

Exercise E-1 Exploring an Encoder Application

Resources

Resources Refer to the following resource for more information about topics in this lesson. For locations with info codes, go to `ni.com/info` and enter the code to access the article.

Article	Location
Linear and Rotary Encoders	Info code: `encoders`

Exercise E-1 Exploring an Encoder Application

Goal

Open and explore an example program that uses counter to acquire an encoder signal.

Scenario

In this exercise, imagine that you are building an application that uses a quadrature encoder to measure the position of a dial. You answer questions about how to set up an encoder measurement and open a NI-DAQmx shipping example to review how to program the measurement.

Implementation

1. Separate the following information into two categories depending on where the information is specified. (Answers on page E-10.)

    - Counter used in measurement
    - Decoding type used
    - Z Index availability
    - Number of pulses per revolution
    - Z index used in measurement

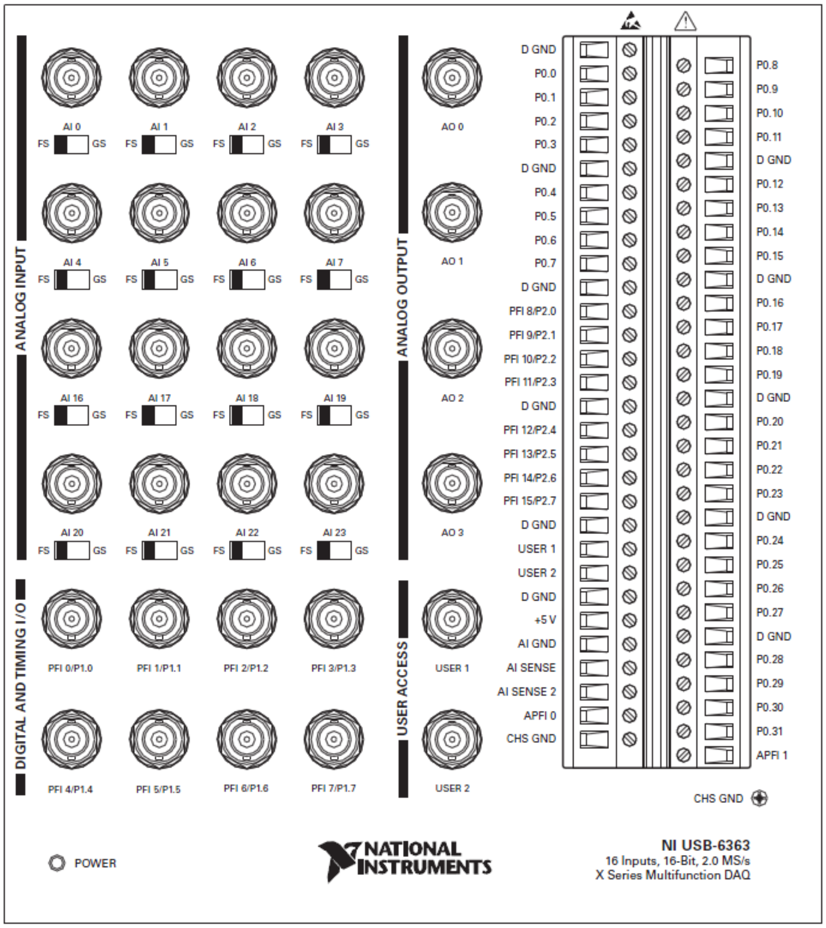
Specified in Encoder Documentation	Specified by NI-DAQmx Task
	Counter used in measurement

2. You are using an old encoder and can no longer find the specifications for it. How might you find the pulses per revolution?

3. Below is the pinout for a USB-6363, an X-Series DAQ device. Where you would connect your encoder's A, B, Z, +5VDC, and COM wires if you were connecting to the default terminals of Counter 0. (Hint: The last two have multiple answers).

Default NI-DAQmx Counter Terminals

Counter/Timer Signal	Terminal Name	Counter/Timer Signal	Terminal Name
CTR 0 SRC	PFI 8	CTR 2 SRC	PFI 0
CTR 0 GATE	PFI 9	CTR 2 GATE	PFI 1
CTR 0 AUX	PFI 10	CTR 2 AUX	PFI 2
CTR 0 OUT	PFI 12	CTR 2 OUT	PFI 14
CTR 0 A	PFI 8	CTR 2 A	PFI 0
CTR 0 Z	PFI 9	CTR 2 Z	PFI 1
CTR 0 B	PFI 10	CTR 2 B	PFI 2
CTR 1 SRC	PFI 3	CTR 3 SRC	PFI 5
CTR 1 GATE	PFI 4	CTR 3 GATE	PFI 6
CTR 1 AUX	PFI 11	CTR 3 AUX	PFI 7
CTR 1 OUT	PFI 13	CTR 3 OUT	PFI 15
CTR 1 A	PFI 3	CTR 3 A	PFI 5
CTR 1 Z	PFI 4	CTR 3 Z	PFI 6
CTR 1 B	PFI 11	CTR 3 B	PFI 7
		FREQ OUT	PFI 14



4.You will be working in an environment with some noise and vibration. Which decoding type should you use?

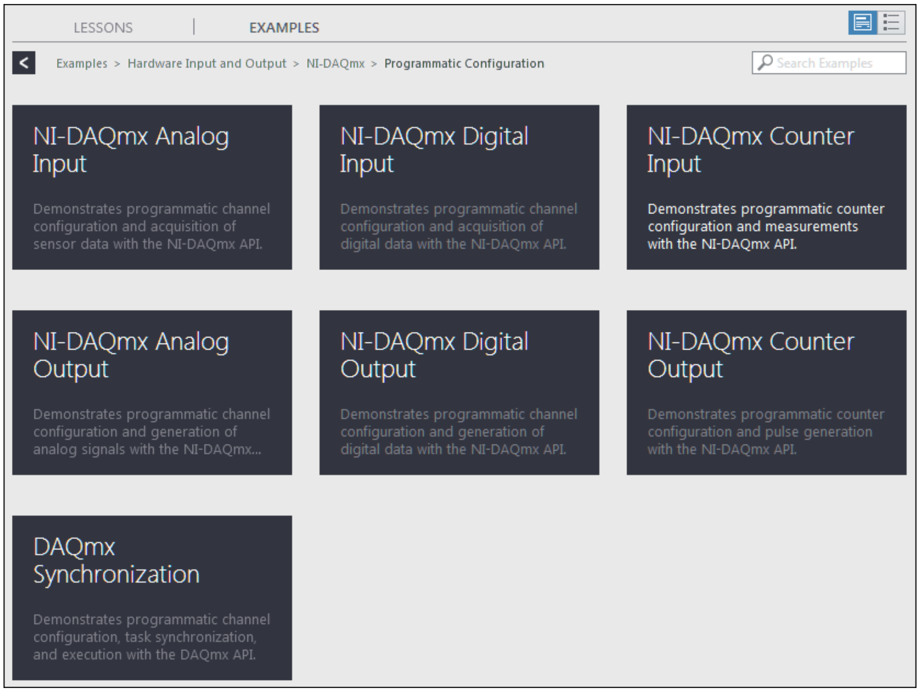
____

____

____

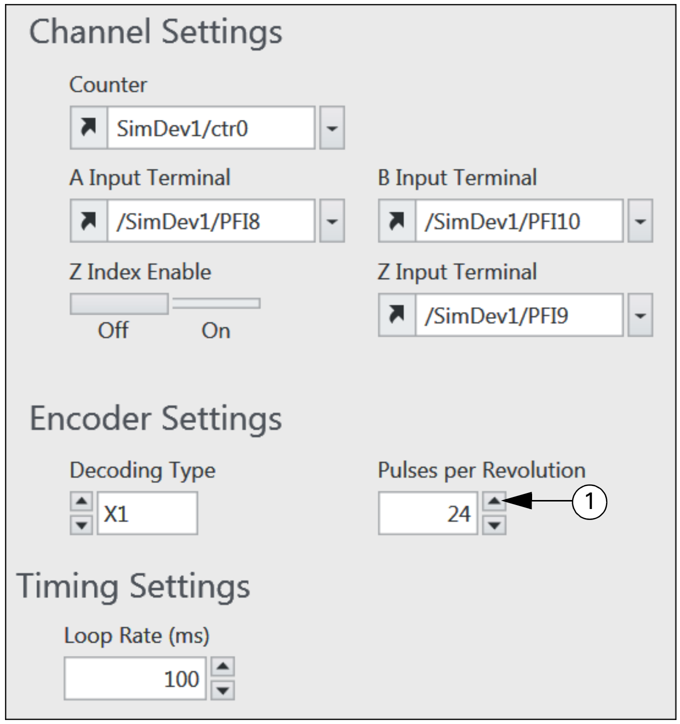
4. On the **Learning** tab, navigate to **Examples»Hardware Input and Output» NI-DAQmx»Programmatic Configuration**.

5. Click **NI-DAQmx Counter Input** and create and project named `Encoder Practice`.



6. Open `Read Angular Encoder On Demand.gvi`.

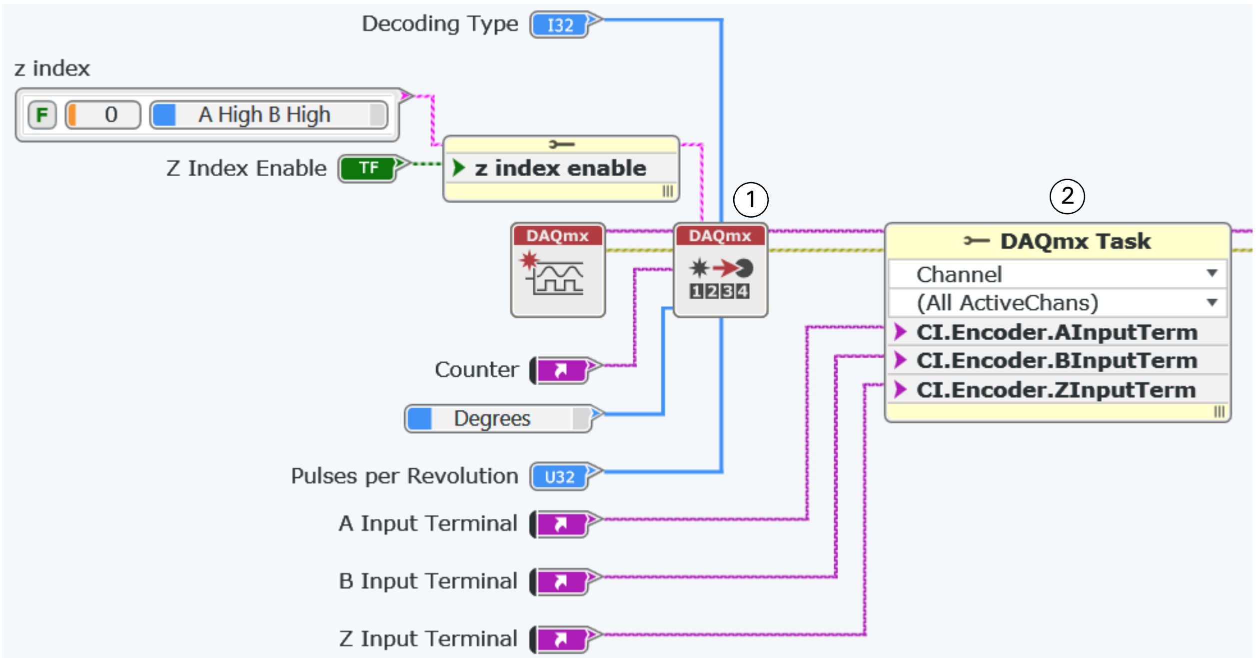
   Notice that the panel of the VI asks for the channel A and channel B input terminals. The VI also gives you the option to enable Z indexing.



1	**Pulses Per Revolution**—Specifies the number of pulses the encoder generates per revolution. This value is the number of pulses on either channel A or channel B. It is not the total number of pulses on both channel A and channel B.

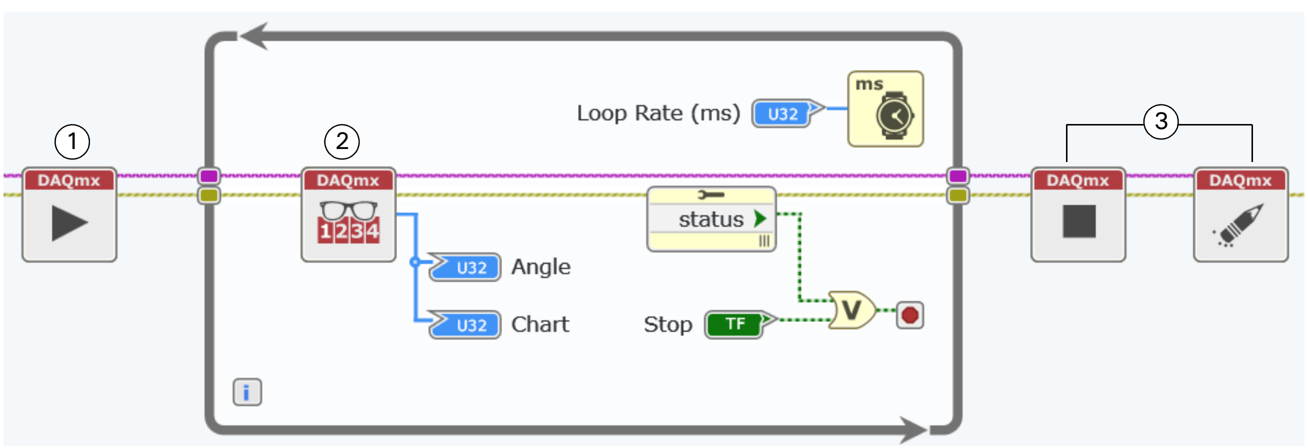
7. Review the diagram to see how the VI is implemented.

   a. For a software-timed acquisition application, you begin on the left of the diagram with a basic DAQ API structure: create a virtual channel and configure timing.



1	**DAQmx Create Virtual Channel**—Using the Context Help for this node, you see that only the **Physical Channel** input is required.
2	**DAQmx Channel Property Node**—This property node is necessary to specify the A, B, and Z channels unless you are using the default channels for the counter you specify. Refer to the *NI-DAQmx Help* for more information about default counter channels.

   b. On the right of the diagram you see the basic DAQ API structure: Start, Read/Write, Stop/Clear.



1	**DAQmx Start Task**—Starting the task means that the device is ready to take a measurement.
2	The While Loop contains a Counter Input instance of the DAQmx Read node. The **Loop Rate (ms)** controls the timing of the acquisition. This means that the timing of the acquisition is dependent on the OS.
3	**DAQmx Stop Task** and **DAQmx Clear Task** node—This example VI uses both for completeness, but the DAQmx Clear Task node will stop the task if necessary.

Answers from page_E-4.

1. Separate the following information into two categories, depending on where this information is specified.

- Counter used in measurement
- Decoding type used
- Z Index availability
- Number of pulses per revolution
- Z index used in measurement

Specified in Encoder Documentation	Specified by NI-DAQmx Task
	Counter used in measurement
Z index availability	**Decoding type used**
Number of pulses per revolution —You find this information in the encoder documentation	**Number of pulses per revolution** —You specify this in the DAQmx Create Virtual Channel node and the value must match the encoder specification.
	Z index used in measurement

2. You are using an old encoder and can no longer find the specifications for it. How might you find the pulses per revolution?

   Use Counter task to count how many edges are output in one revolution.

3. Below is the pinout for a USB-6363, an X-Series DAQ device. Label where you would connect your encoder's A, B, Z, +5VDC, and COM wires if you were connecting to the default terminals of Counter 0.

   (Hint: the last two have multiple answers).

   A connects to PFI 8

   B connects to PFI 10

   Z connects to PFI 9

   Connect Com to any digital ground.

   Connect the +VDC input to any digital line.

4.You will be working in an environment with some noise and vibration. Which decoding type should you use?

   X1 encoding, as it is the least sensitive to changes and noise.

End of Exercise E-1

F Measuring Edges, Frequency, Pulse Width, and Duty Cycle

Topics

+ Resources

Exercises

Resources

Resources Refer to the following resource for more information about topics in this lesson. For locations with info codes, go to `ni.com/info` and enter the code to access the article.

Article	Location
Frequency Measurements: How-to Guide	Info code: `frequencyhowto`

Article	Location
High Frequency Two-Counter Measurement Method	Info code: `hifreq2count`
Frequency Measurements: How-to Guide	Info code: `frequencyhowto`

Article	Location
Large-Range Two-Counter Measurement Method	Info code: `large2count`
Frequency Measurements: How-to Guide	Info code: `frequencyhowto`

Exercise F-1 Simple Edge Counting

Goal

Open and explore an example program to count edges on a digital signal.

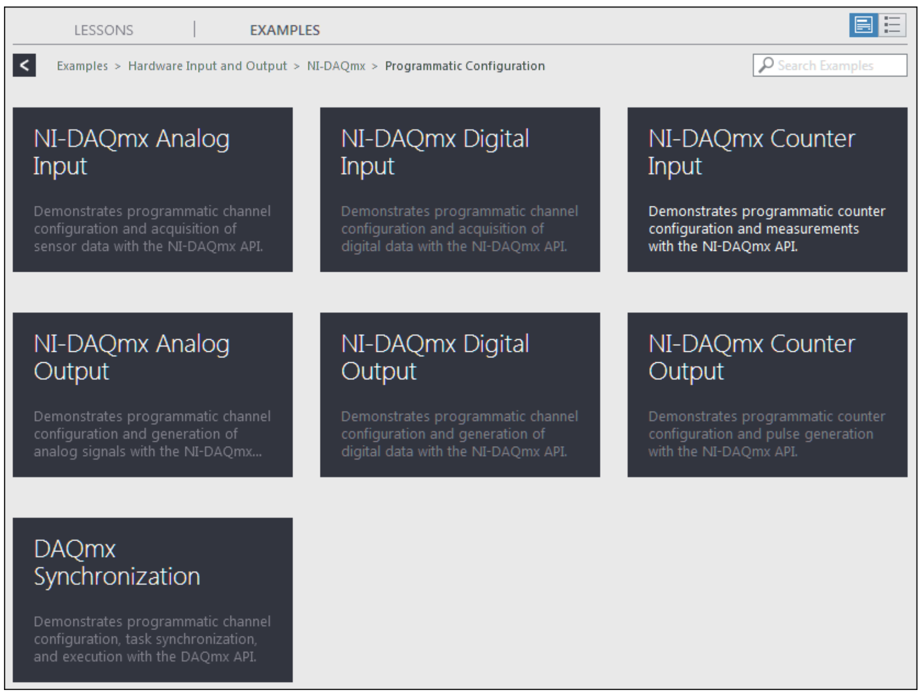
Scenario

In this exercise, you answer questions about how to set up an counter measurement and open a NI-DAQmx shipping example to implement it.

Implementation

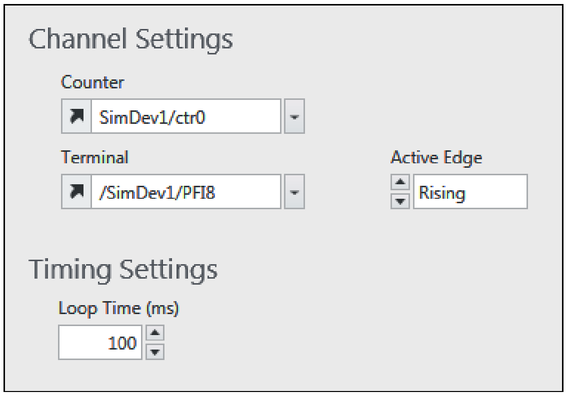
1. On the **Learning** tab, navigate to **Examples»Hardware Input and Output» NI-DAQmx»Programmatic Configuration**.

2. Click **NI-DAQmx Counter Input** and create and project named `Counter Test`.



3. Open `On Demand Digital Edge Count.gvi`.

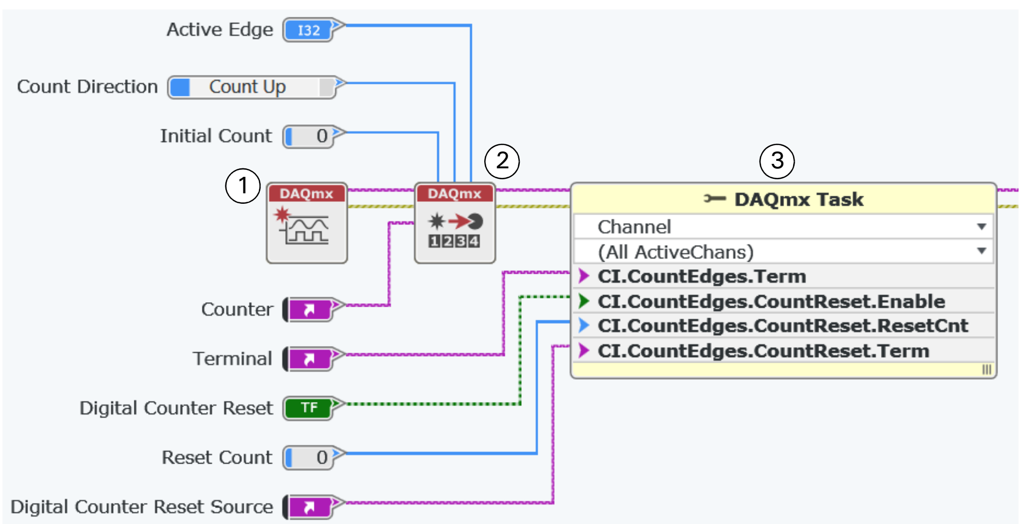
   Notice that the panel of the VI only requires the counter, counter terminal you are measuring, and the active edge specification.



4. Review the diagram to see how the VI is implemented.

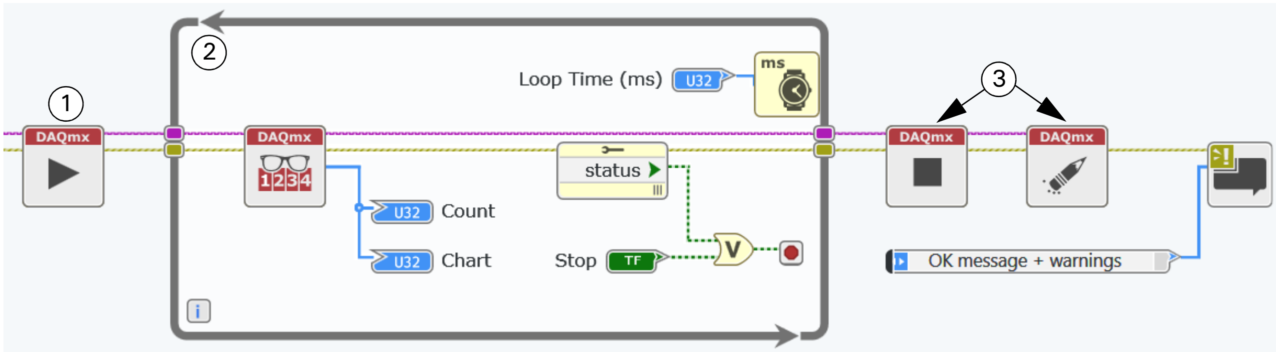
   a. Right-click the **Counter** control and select **Find»On diagram** from the shortcut menu.

   b. Review the components of the diagram. For a software-timed acquisition application, you begin on the left of the diagram with a basic DAQ API structure: create a virtual channel and configure the channel.



1	**DAQmx Create Task**—Creates a task without channel information unless you specify something in the **global virtual channels** input. Because the DAQmx Create Virtual Channel node creates a task if necessary, using the DAQmx Create Task node is optional.
2	**DAQmx Create Virtual Channel**—Using the Context Help for this VI, you see that only the **Counter** input is required.
3	**Property Node**—This property node is necessary to specify the terminal that contains the digital signal that you want to measure.

c. On the right of the diagram you see the basic DAQ API structure: Start, Read/Write, Stop/Clear.



1 **DAQmx Start Task**—Starting the task means that the device is ready to take a measurement.
2 The While Loop contains a Counter Input instance of the DAQmx Read node. The **Loop Time (ms)** input controls the timing of the acquisition, which means the timing of the acquisition is dependent on the OS.
3 **DAQmx Stop Task** node and **DAQmx Clear Task** node—This example VI uses both for completeness, but the DAQmx Clear Task node will stop the task if necessary.

5. Answer the following questions about the built-in counter on the cDAQ-9178. (Answers on page F-7.)

   a. What is the resolution of the built-in counter for the cDAQ-9178?________

   ____

   (Hint: Refer to the *NI cDAQ-9178 User Manual*. A copy of the manual is located in `<Exercises>\DAQNXG\_Resources\Hardware Specifications`.)

   b. What is the highest number that the **Count** indicator will display?_________

   ____

   (Hint: Calculate the terminal count for the cDAQ-9178.)

6. Update the panel so that the VI uses the counter on the cDAQ-9178 to count every time you push down the Trigger button on the demo box. (Answers on page F-7.)

Parameter	Hint	Value
Counter	What are the names of the counter lines for the cDAQ-9178 chassis? Refer to the User Manual or the ni.com, *C Series Physical Channels* topic.	
Input Terminal	Refer to Table A-1, Connections in the NI CompactDAQ Measurements Demo Box, to see how the Trigger button on the demo box is wired.	
Active Edge	Does the digital line go high when you push the Trigger button or when you release the Trigger button?	

Test

1. Run the VI.
2. Press the Trigger button on the NI CompactDAQ Measurements Demo Box repeatedly to watch the **Count** indicator update.
3. Stop the VI.
4. Stop and close the VI without saving it.

Answers from page_F-6.

5. Answer the following questions about the built-in counter on the cDAQ-9178.

   a. What is the resolution of the built-in counter for the cDAQ-9178?

   The resolution of the built-in counter for the cDAQ-9178 chassis is 32 bits.

   b. What is the highest number that will be output by the count indicator?_____

   Terminal count = 2^(counter resolution) – 1
   So the highest count displayed would be 2^(32) – 1, which equals 4294967295.

Answers from page_F-6.

7. Update the panel so that the VI uses the counter on the cDAQ-9178 to count rising edges.

Parameter	Hint	Value
Counter	What are the names of the counter lines for the cDAQ-9178 chassis? Refer to the User Manual or the *NI-DAQmx Help*, **C Series Physical Channels** topic.	**Any of the following will work:** **cDAQ1/_ctr0** **cDAQ1/_ctr1** **cDAQ1/_ctr2** **cDAQ1/_ctr3**
Input Terminal	Refer to Table A-1, Connections in the NI CompactDAQ Measurements Demo Box, to see how the Trigger button on the demo box is wired.	**PFI 0**
Active Edge	Does the digital line go high when you push the Trigger button or when you release the Trigger button?	**Rising**

End of Exercise F-1

G Additional Information and Resources

National Instruments provides global services and support as part of our commitment to your success. Take advantage of product services in addition to training and certification programs that meet your needs during each phase of the application life cycle; from planning and development through deployment and ongoing maintenance.

A. NI Services

To get started, register your product at ni.com/myproducts.

As a registered NI product user, you are entitled to the following benefits:

- Access to applicable product services.
- Easier product management with an online account.
- Receive critical part notifications, software updates, and service expirations.

Log in to your National Instruments ni.com User Profile to get personalized access to your services.

B. Services and Resources

- **Maintenance and Hardware Services**—NI helps you identify your systems' accuracy and reliability requirements and provides warranty, sparing, and calibration services to help you maintain accuracy and minimize downtime over the life of your system. Visit ni.com/services for more information.
  - **Warranty and Repair**—All NI hardware features a one-year standard warranty that is extendable up to five years. NI offers repair services performed in a timely manner by highly trained factory technicians using only original parts at a National Instruments service center.
  - **Calibration**—Through regular calibration, you can quantify and improve the measurement performance of an instrument. NI provides state-of-the-art calibration services. If your product supports calibration, you can obtain the calibration certificate for your product at ni.com/calibration.
- **System Integration**—If you have time constraints, limited in-house technical resources, or other project challenges, National Instruments Alliance Partner members can help. To learn more, call your local NI office or visit ni.com/alliance.
- **Training and Certification**—The NI training and certification program is the most effective way to increase application development proficiency and productivity. Visit ni.com/training for more information.
  - The Skills Guide assists you in identifying the proficiency requirements of your current application and gives you options for obtaining those skills consistent with your time and budget constraints and personal learning preferences. Visit ni.com/skills-guide to see these custom paths.
  - NI offers courses in several languages and formats including instructor-led classes at facilities worldwide, courses on-site at your facility, and online courses to serve your individual needs.
- **Technical Support**—Support at ni.com/support includes the following resources:
  - **Self-Help Technical Resources**—Visit ni.com/support for software drivers and updates, a searchable KnowledgeBase, product manuals, step-by-step troubleshooting wizards, thousands of example programs, tutorials, application notes, instrument drivers, and so on. Registered users also receive access to the NI Discussion Forums at ni.com/forums. NI Applications Engineers make sure every question submitted online receives an answer.

- **Software Support Service Membership**—The Standard Service Program (SSP) is a renewable one-year subscription included with almost every NI software product, including NI Developer Suite. This program entitles members to direct access to NI Applications Engineers through phone and email for one-to-one technical support, as well as exclusive access to online training modules at `ni.com/self-paced-training`. NI also offers flexible extended contract options that guarantee your SSP benefits are available without interruption for as long as you need them. Visit `ni.com/ssp` for more information.
- **Declaration of Conformity (DoC)**—A DoC is our claim of compliance with the Council of the European Communities using the manufacturer's declaration of conformity. This system affords the user protection for electromagnetic compatibility (EMC) and product safety. You can obtain the DoC for your product by visiting `ni.com/certification`.

For information about other technical support options in your area, visit `ni.com/services`, or contact your local office at `ni.com/contact`.

You also can visit the Worldwide Offices section of `ni.com/niglobal` to access the branch office websites, which provide up-to-date contact information, support phone numbers, email addresses, and current events.

C. Other National Instruments Training Courses

National Instruments offers several training courses for LabVIEW users. These courses continue the training you received here and expand it to other areas. Visit `ni.com/training` to purchase course materials or sign up for instructor-led, hands-on courses at locations around the world.

D. National Instruments Certification

Earning an NI certification acknowledges your expertise in working with NI products and technologies. The measurement and automation industry, your employer, clients, and peers recognize your NI certification credential as a symbol of the skills and knowledge you have gained through experience. Visit `ni.com/training` for more information about the NI certification program.