Data Acquisition and Signal Conditioning Exercises

Course Software Version 2009
February 2010 Edition
Part Number 324980B-01

Worldwide Technical Support and Product Information
`ni.com`

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

Worldwide Offices
Australia 1800 300 800, Austria 43 662 457990-0, Belgium 32 (0) 2 757 0020, Brazil 55 11 3262 3599, Canada 800 433 3488,
China 86 21 5050 9800, Czech Republic 420 224 235 774, Denmark 45 45 76 26 00, Finland 358 (0) 9 725 72511,
France 01 57 66 24 24, Germany 49 89 7413130, India 91 80 41190000, Israel 972 3 6393737, Italy 39 02 41309277,
Japan 0120-527196, Korea 82 02 3451 3400, Lebanon 961 (0) 1 33 28 28, Malaysia 1800 887710, Mexico 01 800 010 0793,
Netherlands 31 (0) 348 433 466, New Zealand 0800 553 322, Norway 47 (0) 66 90 76 60, Poland 48 22 328 90 10,
Portugal 351 210 311 210, Russia 7 495 783 6851, Singapore 1800 226 5886, Slovenia 386 3 425 42 00,
South Africa 27 0 11 805 8197, Spain 34 91 640 0085, Sweden 46 (0) 8 587 895 00, Switzerland 41 56 2005151,
Taiwan 886 02 2377 2222, Thailand 662 278 6777, Turkey 90 212 279 3031, United Kingdom 44 (0) 1635 523545

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

Contents

Student Guide

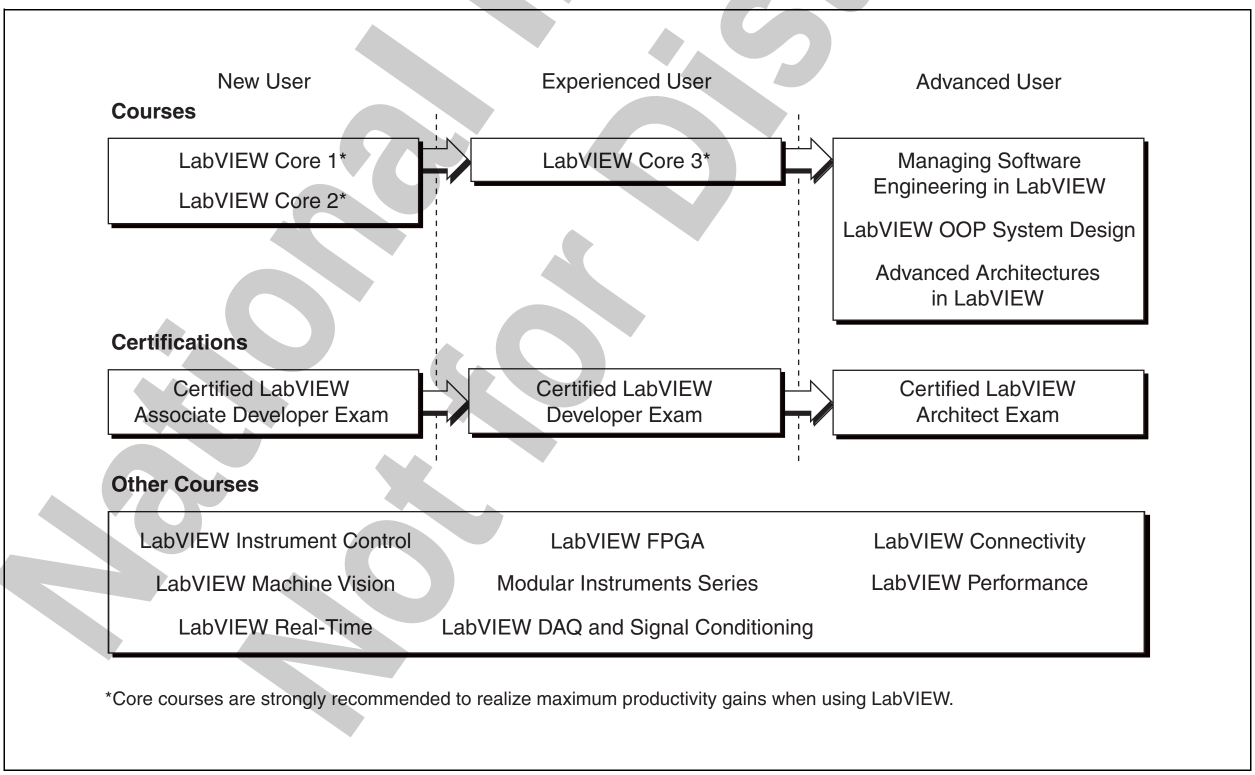
Thank you for purchasing the *Data Acquisition and Signal Conditioning* course kit. This course manual and accompanying software are used in the two-day, hands-on *Data Acquisition and Signal Conditioning* course.

You can apply the full purchase of this course kit toward the corresponding course registration fee if you register within 90 days of purchasing the kit. Visit `ni.com/training` for online course schedules, syllabi, training centers, and class registration.

Note For course manual updates and corrections, refer to `ni.com/info` and enter the info code `daqnsc`.

A. NI Certification

The *Data Acquisition and Signal Conditioning* course is one of many courses offered by National Instruments. The following illustration shows the courses that are part of the LabVIEW training series. If you want to build your proficiency with LabVIEW and prepare for exams to become an NI Certified LabVIEW Developer and NI Certified LabVIEW Architect, refer to `ni.com/training`.



B. Course Description

This course teaches you the fundamentals of data acquisition (DAQ) and signal conditioning and introduces techniques you can implement in real-world applications.

This course also expands on recommended LabVIEW programming styles from the *LabVIEW Core 1* and *LabVIEW Core 2* courses and discusses several programming guidelines you can use in your data acquisition and signal conditioning applications. This course manual assumes that you are familiar with Windows, that you have experience writing algorithms in the form of flowcharts or block diagrams, and that you have taken the *LabVIEW Core 1* course or you have familiarity with all the concepts contained therein. The course and exercises manuals are divided into lessons, described as follows.

In the course manual, each lesson consists of the following:

- An introduction that describes the purpose of the lesson and what you will learn
- A description of the topics in the lesson
- A summary that outlines important concepts and skills taught in the lesson
- A summary quiz that tests and reinforces important concepts and skills taught in the lesson

In the exercise manual, each lesson consists of the following:

- A set of exercises to reinforce topics
- (Optional) Self-study and challenge exercise sections or additional exercises

Several exercises in this manual use a plug-in multifunction I/O data acquisition (DAQ) device connected to an NI BNC-2120. Some exercises also use a cDAQ chassis, NI 9219 universal analog input module, and sensors as described in Lesson 7, *Signal Conditioning*.

C. What You Need to Get Started

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

❑ Windows XP or later installed on your computer.

❑ LabVIEW Full or Professional Development System 2009 or later

❑ NI-DAQmx driver software version 9.0.2 or later

- ❑ Multifunction DAQ device
- ❑ NI BNC-2120, wires, and cable
- ❑ cDAQ chassis, NI 9219 universal analog input module, and sensors
- ❑ USB Cable
- ❑ *Data Acquisition and Signal Conditioning* course CD, containing the following files:

Directory	Description
Exercises	Contains all the VIs and support files needed to complete the exercises in this course
Solutions	Contains completed versions of the VIs you build in the exercises for this course

Note This course assumes you are using the default installation of LabVIEW. If you have changed the palette views from the default settings, some palette paths described in the course may not match your settings. To reset palette views to LabVIEW defaults, select **Tools»Options** from the top pull-down menu and select **Controls/Functions Palettes** under Category. Set **Palette** to **Category (Standard)**, set **Navigation Buttons** to **Label Selected Icons**, set **Loading** to **Load palette in background**, and enable the **Use window titles in Functions palette** check box. Click **OK** to apply the changes and close the dialog box.

Installing the Course Software

To install the software used for this course, insert the course CD and install the required files in the following directory structure:

- All exercise VIs and associated support files are in the <Exercises>\DAQ and Signal Conditioning directory.
- All exercise solutions are in the <Exercises>\DAQ and Signal Conditioning directory.

Tip Folder names in angle brackets, such as <Exercises>, refer to folders on the root directory of your computer.

D. Course Goals

The purpose of this course is to teach you the components of a DAQ system and to teach you how to use that system. By the end of this course, you will understand the five components of a DAQ system:

- Sensors
- Signals
- Signal conditioning
- DAQ hardware
- DAQ software

You also will know how to use LabVIEW with a DAQ device to:

- Acquire analog signals
- Perform signal conditioning on acquired signals
- Generate analog signals
- Perform non-timed digital input and output
- Use counters for event counting, pulse generation, pulse measurement, and frequency measurement

This course does *not* describe any of the following:

- Basic principles of LabVIEW covered in the *LabVIEW Core 1* course
- Every built-in VI, function, or object; refer to the *LabVIEW Help* for more information about LabVIEW features not described in this course
- Developing a complete application for any student in the class; refer to the NI Example Finder, available by selecting **Help»Find Examples**, for example VIs you can use and incorporate into VIs you create

Refer to the *LabVIEW Help* for more information about a particular DAQmx VI.

If you are seeking help for developing your application, please discuss this need with your instructor outside of class time or contact National Instruments Technical Support. For further information on technical support, refer to the National Instruments Web site at `ni.com`.

E. Course Conventions

The following conventions are used in this course manual:

» — The » symbol leads you through nested menu items and dialog box options to a final action. The sequence **File»Page Setup»Options** directs you to pull down the **File** menu, select the **Page Setup** item, and select **Options** from the last dialog box.

This icon denotes a tip, which alerts you to advisory information.

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

bold — Bold text denotes items that you must select or click in the software, such as menu items and dialog box options. Bold text also denotes parameter names, controls and indicators on the front panel, dialog boxes and pages, and sections and components of dialog boxes. Window names and palette names are also denoted in bold.

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

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

2

Data Acquisition Hardware and Software Exercises

Exercise 2-1 Range, Resolution, Code Width, and Accuracy

Objective

To 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. Table 2-1 lists three DAQ devices you can use for the project in this exercise. Typical DAQ devices have either 16- or 18-bit resolution and offer a variety of input ranges.

The DAQ device information in the Table 2-1 represents typical DAQ devices. In this exercise, select the best DAQ device and the optimum input range to maximize accuracy by following these steps:

1. The best approach is to 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.
3. Then, determine if the absolute accuracy of the DAQ device meets the measurement requirements of the project.

Tip Use the code width equation.

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

Table 2-1 lists the three DAQ devices and their corresponding specifications:

Table 2-1. DAQ Device Resolution and Input Range Values

	DAQ Device 1	DAQ Device 2	DAQ Device 3
Resolution	16 Bit	16 Bit	18 Bit
Input Ranges (Volts)	±10, ±5, ±1, ±0.2	±10, ±5, ±2, ±1, ±0.5, ±0.2, ±0.1	±10, ±5, ±2, ±1, ±0.5, ±0.2, ±0.1

Tables 2-2, 2-3, and 2-4 display the values for the analog input (AI) absolute accuracy of each nominal range for each DAQ device.

Table 2-2. DAQ Device 1 AI Absolute Accuracy

Nominal Range		Absolute Accuracy at Full Scale (μV)
Positive Full Scale	**Negative Full Scale**	
10	–10	3,100
5	–5	1,620
1	–1	360
0.2	–0.2	112

Table 2-3. DAQ Device 2 AI Absolute Accuracy

Nominal Range		Absolute Accuracy at Full Scale (μV)
Positive Full Scale	**Negative Full Scale**	
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 2-4. DAQ Device 3 AI Absolute Accuracy

Nominal Range		Absolute Accuracy at Full Scale (µV)
Positive Full Scale	Negative Full Scale	
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

Project 1

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 to 1,372 ºC. With this temperature range, the thermocouple returns a voltage of –6.548 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

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

Solutions

Project 1

1. In this project, the thermocouple outputs a voltage ranging from –6.548 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 2-1

Exercise 2-2 Using Measurement & Automation Explorer

Objective

To become familiar with the Devices and Interfaces section of Measurement and Automation Explorer (MAX) and to explore the test panel functionality.

BNC-2120 Configuration

1. Connect the **Sine/Triangle** BNC connector in the Function Generator section to the BNC connector of channel 1 in the Analog Inputs section.

2. Move the **Sine/Triangle** switch to the `Sine` position.

3. Move the **BNC/Thermocouple** switch to the `BNC` position.

4. Connect the BNC connector of channel 0 in the Analog Outputs section to the BNC connector of channel 2 in the Analog Inputs section.

5. Move the **FS/GS** switches underneath the BNC connectors of channel 0 and channel 2 to the `GS` position.

6. Move the **BNC/Temp. Ref.** switch to the `Temp. Ref.` position.

Implementation

1. Launch MAX by double-clicking the icon on the desktop or selecting **Start»All Programs»National Instruments»Measurement & Automation**.

2. Expand the **Devices and Interfaces** category. MAX searches for installed hardware and lists the devices found.

   The default value for the multifunction DAQ device name is `"Dev1"`. If the device is not listed, select **View»Refresh**.

Note If the device name is not `"Dev1"`, right-click the device and select **Rename** and rename the device `Dev1`.

3. Right-click the DAQ device and select **Self-Test** from the shortcut menu. A dialog box appears that indicates the device has passed the test. Click the **OK** button to close the dialog box.

Note If the DAQ device does not pass the self-test, notify your instructor.

4. Right-click the DAQ device and select **Properties** from the shortcut menu. The **Device Properties** dialog box appears.

   The **RTSI Configuration** tab specifies if a Real Time System Integration (RTSI) cable is attached to the device. The RTSI cables route internal signals from one device to another.

Note The **RTSI Configuration** tab is not applicable to PXI systems.

   The **Accessory** tab allows you to select and configure accessories that are attached to the DAQ device. In the Accessory pull-down menu, select **BNC-2120**.

5. Click **OK** to exit the **Device Properties** dialog box.

6. Right-click the DAQ device and select **Test Panels** from the shortcut menu. The Test Panels dialog box appears. The **Analog Input** tab allows you to read the analog input channels.

7. On the Analog Input tab, set the **Channel Name** to `Dev1/ai0`. Channel 0 is the temperature sensor on the BNC-2120 when the BNC/Temp. Ref. switch is set to the `Temp. Ref.` position.

   ❑ Set the **Mode** to **Continuous** acquisition.

   ❑ Set the **Rate (Hz)** and **Samples To Read** to `100`.

   ❑ Click the **Start** button. A voltage between 0.2 and 0.3 V should display on the graph, as shown in Figure 2-1. The measured value of voltage is displayed in the **Value** field.

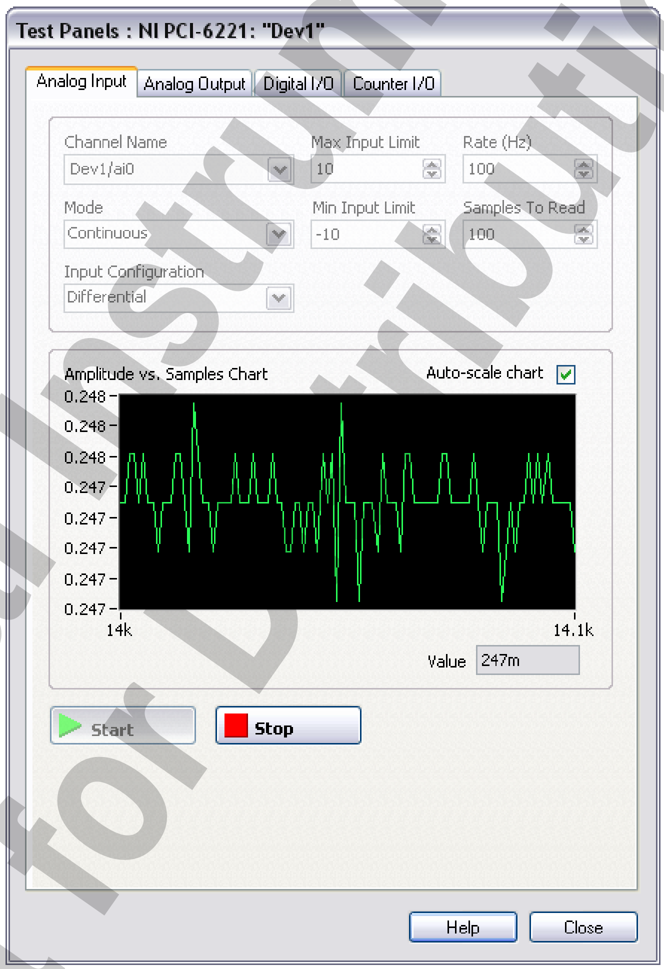


Figure 2-1. Analog Input Test Panel

8. Place your finger on the temperature sensor on the BNC-2120 to increase the voltage. Click the **Stop** button when finished.

Note The test panel is an effective method of troubleshooting because it communicates directly with the NI-DAQmx drivers. If the test panel does not work, the problem is with either the hardware or the driver configuration. If the test panel functions properly and the LabVIEW VI does not produce the expected results, the problem is with the LabVIEW VI.

9. On the Analog Input test panel, configure the following settings:

   ❑ Change the channel to `Dev1/ai1`.

Note Make sure the **Sine/Triangle** BNC connector from the function generator connects to the BNC connector of channel 1 in the Analog Inputs section.

   ❑ Set the **Mode** to **On Demand**.

   The Mode pull-down menu includes the following options:

   - **On Demand**—Acquires a single point of data.
   - **Finite—**Displays only one screen of data.
   - **Continuous**—Continuously displays a screen of data at a time.

   The **Finite** and **Continuous** acquisition modes allow you to adjust the sample rate. The higher the sample rate, the better the graph displays the shape of the waveform.

   ❑ Click the **Start** button. The sine wave might look distorted.

   ❑ Click the **Stop** button.

10. Complete the following steps to improve the accuracy of the graph display.

   ❑ On the BNC-2120, set the Frequency Selection switch to the leftmost position and turn the **Frequency Adjust** dial to `LO`.

   ❑ On the Test Panel, set the **Mode** to **Finite** or **Continuous** and try different values, such as `5000`, `10000`, or `15000` for **Rate (Hz)** until the graph displays a smooth sine wave.

   ❑ Stop the acquisition once you have verified the signal as a smooth sine wave.

11. Click the **Analog Output** tab. On this page you can generate a DC voltage or sine wave on one of the analog output channels of the DAQ device. Complete the following steps to output a DC voltage on analog output channel 0.

    - ❑ Verify that the BNC connector of channel 0 in the Analog Outputs section is connected the BNC connector of channel 2 in the Analog Inputs section.
    - ❑ Set the Mode to **DC Value**.
    - ❑ Enter 5 for the **Output Value** and click **Update**.

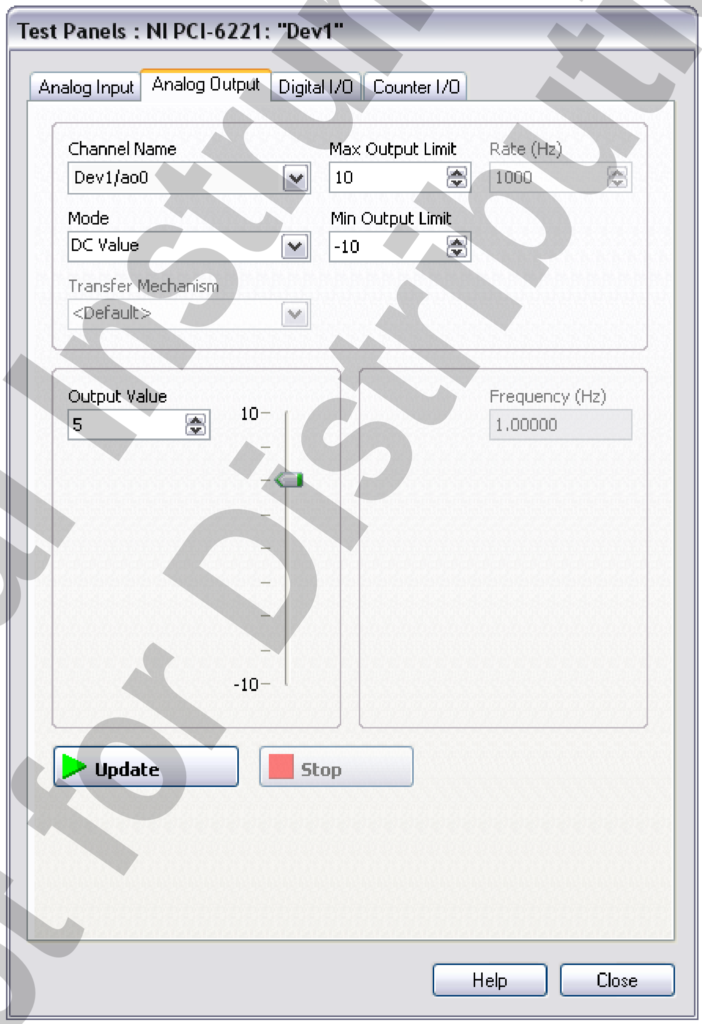


Figure 2-2. Analog Output Test Panel

12. Click the **Analog Input** tab.

- ❑ Set the **Channel Name** to `Dev1/ai2`.
- ❑ Set the **Mode** to **On Demand**.
- ❑ Disable the **Auto-scale chart** checkbox.
- ❑ Click the **Start** button. You should see 5 V displayed on the graph, as shown in Figure 2-3.
- ❑ Click the **Stop** button.

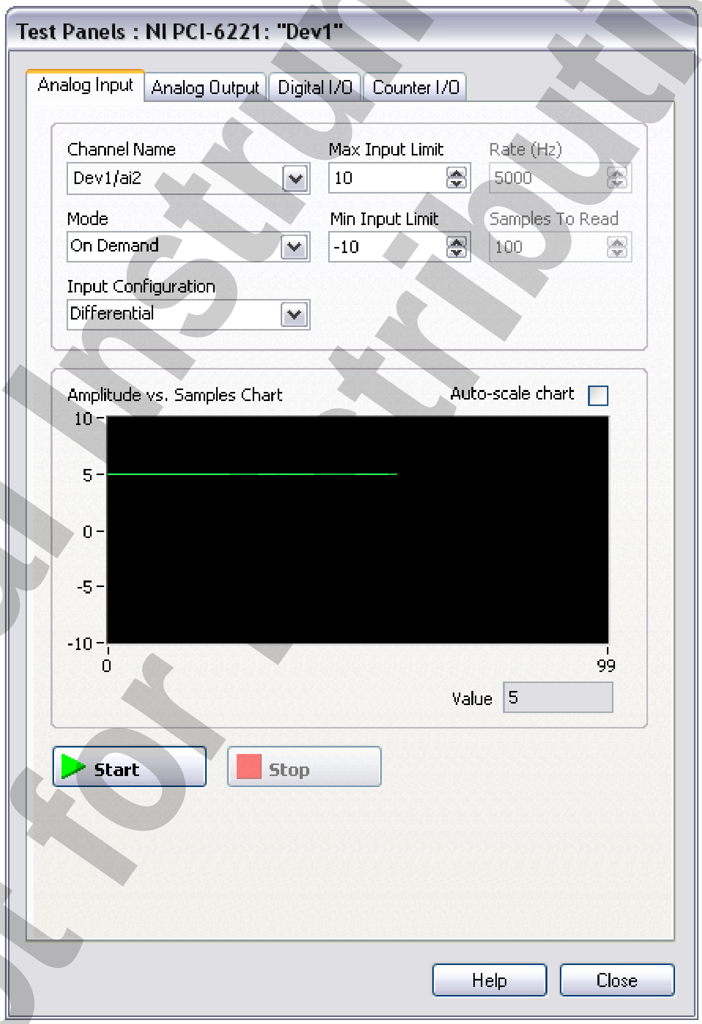


Figure 2-3. Analog Input Test Panel

13. Click the **Digital I/O** tab to access the eight digital lines on the device.

- ❑ Select **port0** as the **Port Name**.
- ❑ In the Select Direction section, click **All Output** to set lines 0 through 7 to be output lines. These lines correspond to the eight LEDs on the BNC-2120.
- ❑ Click **Start** to begin updating the digital values. In the Select State section, toggle these lines by clicking on the toggle switches.
- ❑ Watch the LEDs turn on and off. The LEDs turn on when the switches in the Select State section are set to the **High (1)** position. In other words, the LED turns on when the digital line outputs a logic high.
- ❑ Click **Stop** when finished.

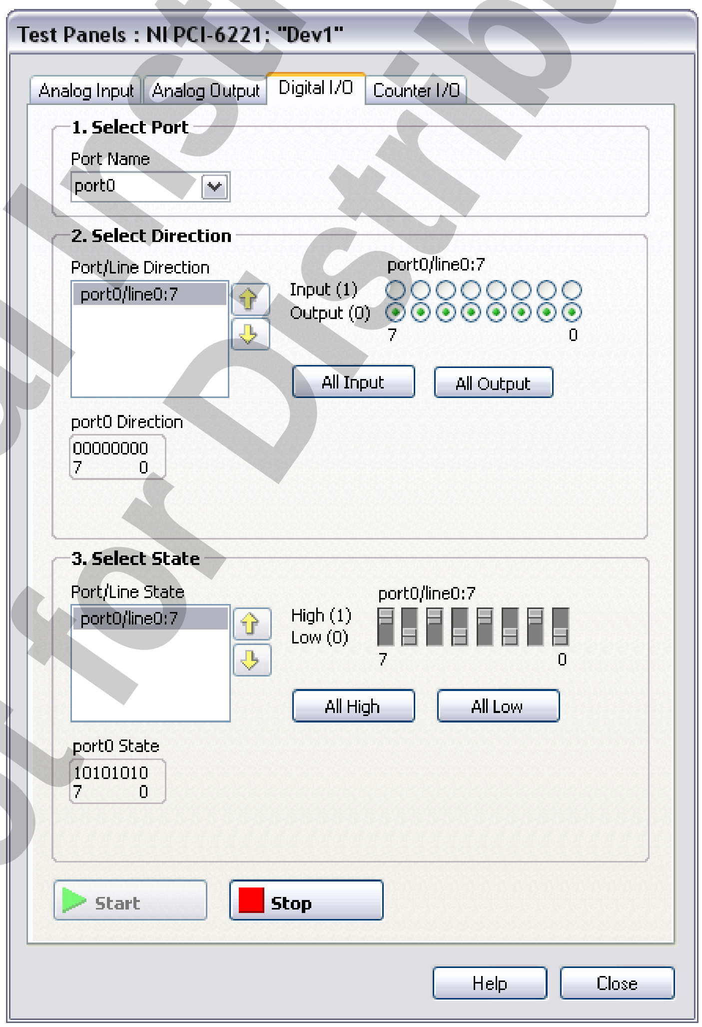


Figure 2-4. Digital I/O Test Panel

14. Click the **Counter I/O** tab to verify counter/timer operation.

    ❑ Set **Mode** to **Edge Counting**. The **Edge Source** defaults to `/Dev1/20MHzTimeBase`. This counts the pulses of a 20 MHz onboard signal.

    ❑ Click the **Start** button. The **Counter Value** should increment rapidly.

    ❑ Click the **Stop** button to stop the counter test.

Figure 2-5. Counter I/O Test Panel

15. Close the test panel and leave MAX open.

End of Exercise 2-2

Exercise 2-3 DAQ Assistant

Objective

To create two NI-DAQmx global virtual channels and create an NI-DAQmx task that uses the two NI-DAQmx global virtual channels and one local channel in MAX. You will also create a custom scale to convert the units of the temperature sensor from voltage to degrees Celsius. You will also use the NI-DAQmx task to generate LabVIEW code.

BNC-2120 Configuration

1. Connect the **Sine/Triangle** BNC connector in the Function Generator section to the BNC connector of channel 1 in the Analog Inputs section.

2. Move the **Sine/Triangle** switch to the `Sine` position.

3. Connect the **TTL Square Wave** BNC connector in the Function Generator section to the BNC connector of channel 2 in the Analog Inputs section.

4. Move the **FS/GS** switches underneath the BNC connectors of channel 1 and channel 2 to the `GS` position.

Part I: Creating DAQmx Global Virtual Channels

Note DAQmx Global Virtual Channels can be useful if you intend to use the same measurement in the same configuration, but with different timing and triggering properties.

1. If MAX is not already open, reopen MAX.

2. To configure channels with the DAQ Assistant, right-click **Data Neighborhood** and select **Create New**.

3. Select **NI-DAQmx Global Virtual Channel** and click the **Next** button.

4. In the dialog box that appears, select **Acquire Signals»Analog Input»Voltage**.

   Although you will take a temperature measurement for the temperature sensor on the BNC-2120, do not select **Temperature** as the sensor type. Use the Temperature type when using a temperature-specific transducer, such as a thermocouple or RTD.

5. The next dialog box that appears prompts you to select the physical channel to use with your new virtual channel. Under **Dev1**, select **ai0** and click **Next**.

6. Name the channel `Temperature Sensor`. Click **Finish**. A similar channel configuration, as shown in Figure 2-6, should appear in MAX.

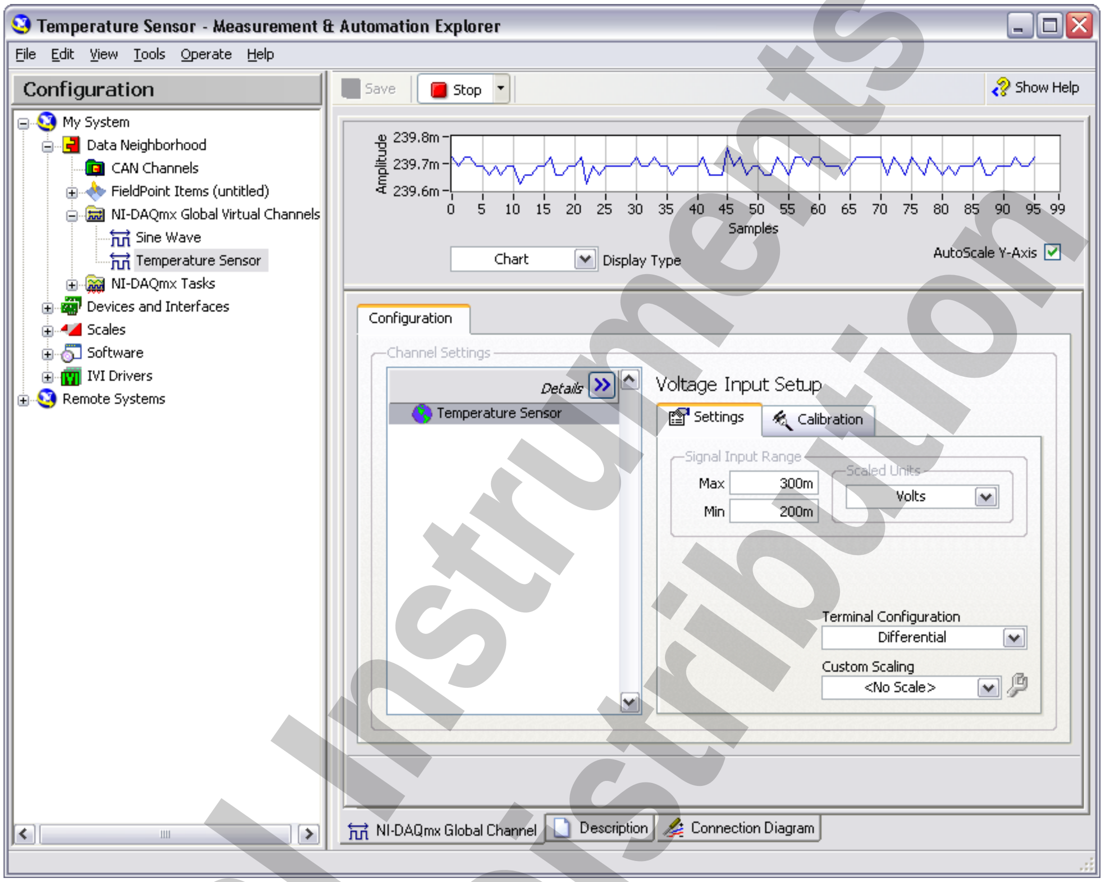


Figure 2-6. Temperature Sensor NI-DAQmx Global Virtual Channel

7. Click the **Run** button to test the channel. Verify that you are receiving data within 200m (0.2 V) to 300m (0.3 V). Set Display Type to **Table** and **Chart** to try different ways of viewing the data. Click the **Stop** button to finish testing the channel.

   After creating an NI-DAQmx channel in MAX, always test the channel before continuing to create other channels.

8. Repeat steps 2 to 6 to create an additional channel with the following settings:

   - Second Channel—Sine Wave
     - Measurement Type: **Acquire Signals»Analog Input»Voltage**
     - Physical Channel: **Dev1»ai1**
     - Name: `Sine Wave`

Part I of this exercise illustrated a few of the conveniences of virtual channels:

- You can use an intuitive name instead of a number to identify a channel.
- You can test the channels to make sure you have properly configured your channels.
- Virtual channels allow easy assignment of units to a given channel.
- When you configure a virtual channel, you can choose from various measurement and sensor types, which automatically apply the required conversion scale.

Part II: Creating a DAQmx Task

In Part I, you created two global channels. When two or more channels have similar timing and triggering needs, group the channels into a logical collection called an NI-DAQmx Task. In Part II, you will create a task and add the two previously created global channels and one local channel to the task.

1. Create a task and add the previously created NI-DAQmx Global Virtual Channels and one local channel to the task.

   ❑ Right-click **Data Neighborhood** and select **Create New**.

   ❑ Select **NI-DAQmx Task** and click **Next**.

   ❑ Select **Acquire Signals»Analog Input»Voltage** as the measurement type.

   ❑ The dialog box prompts you to select the physical channels to include in the task. However, you want to use the NI-DAQmx Global Virtual Channels you created in Part I. To do so, click the **Virtual** tab.

   By default, the **Add Global Channels** option is already selected so that any changes you make later to those channels will be reflected in the created NI-DAQmx Task.

- ❑ Press and hold the <Shift> key and select both channels, as shown in Figure 2-7.

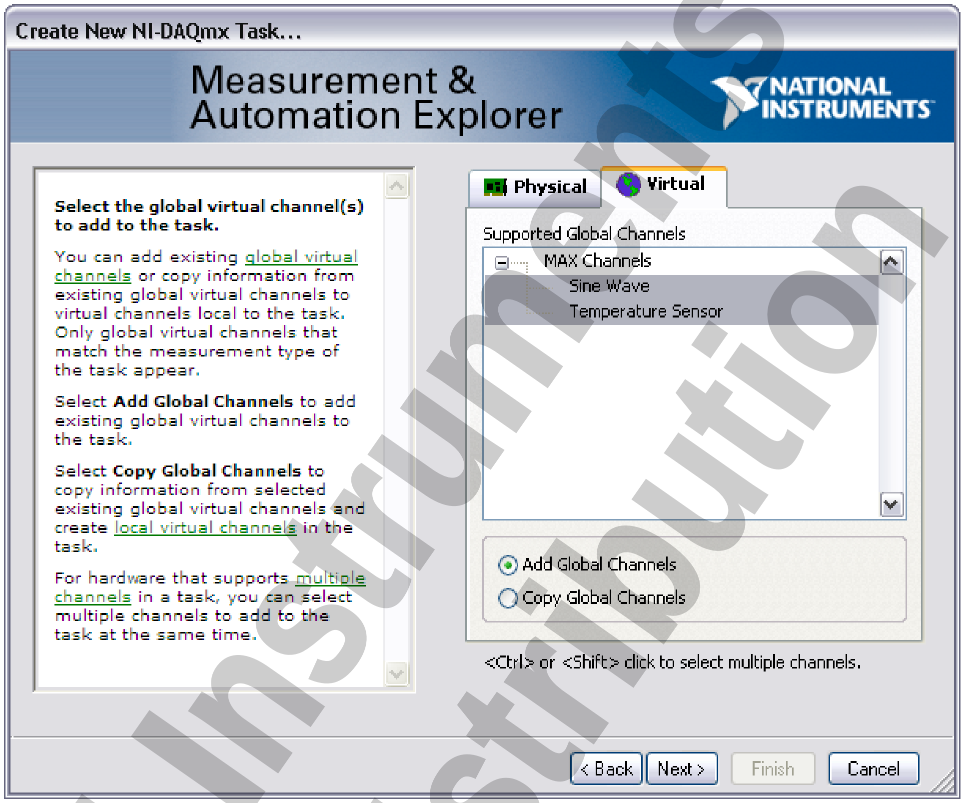


Figure 2-7. Adding Global Virtual Channels to NI-DAQmx Task

- ❑ Click the **Next** button.
- ❑ Name the task `MyVoltageTask` and click **Finish** to create your NI-DAQmx Task.

2. Review the **Configuration** tab of the newly created NI-DAQmx Task. Right-click the **Temperature Sensor** channel and examine the shortcut menu. Notice how you can change the channel to a local channel so that changes made to the channel will only be seen in this one task.

3. Add a physical channel to the task.

❑ Click the **Add Channels** (plus sign) button and select **Voltage**. Select the physical channel **ai2** for your DAQ device. From the Location To Add Channels pull-down menu, select **End of the scan**, as shown in Figure 2-8.

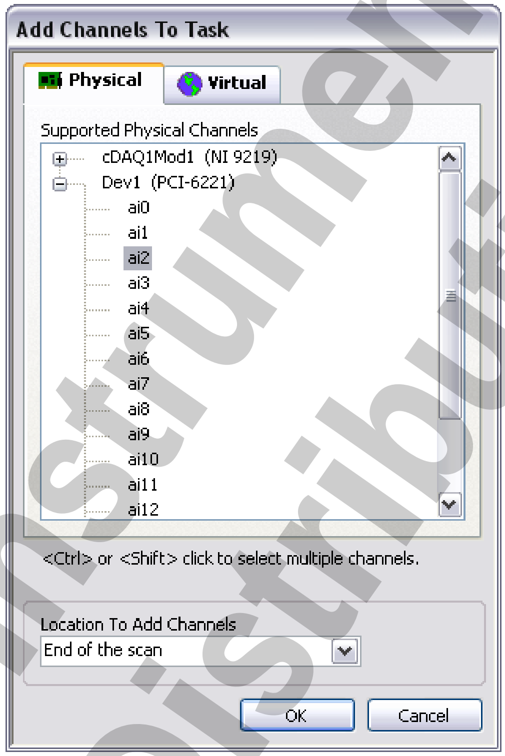


Figure 2-8. Adding Local Physical Channel to NI-DAQmx Task

❑ Click **OK** to create a channel named **Voltage**. To rename the channel, right-click the channel and select **Rename** from the shortcut menu. Rename the channel as `Square Wave` as shown in Figure 2-9.

Figure 2-9. Renaming Local Physical Channel in NI-DAQmx Task

❑ Click **OK** to return to the NI-DAQmx Task. Notice that Square Wave channel does not have a globe icon next to it. This indicates that it is a local channel and can only be used by the `MyVoltageTask` task.

4. Configure the Timing Settings section by verifying that the values are set to:

   - **Acquisition Mode**: **Continuous Samples**
   - **Samples to Read**: `100`
   - **Rate (Hz)**: `1k`

5. Test the DAQmx Task.

   ❑ On the BNC-2120, turn the **Amplitude Adjust** dial to `HI`.

   ❑ At the top of the task click the **Run** button to acquire data. Choose **Graph** for the **Display** type to see a better representation of the data, as shown in Figure 2-10.

   ❑ Click **Stop** to end testing.

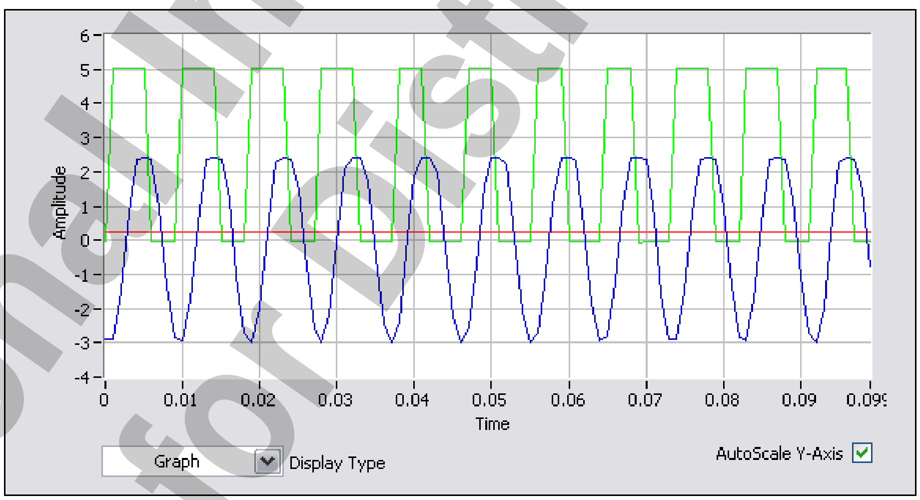


Figure 2-10. Graph Display of NI-DAQmx Task

6. Click **Save** to save the task to MAX.

Part III: Custom Temperature Scale

The voltage values you receive for the temperature sensor on the BNC-2120 are in the range of 0.2 to 0.3 V. This voltage level multiplied by 100 corresponds to the temperature reading in degrees Celsius.

1. Add a custom scale to the Temperature Sensor channel so that the Temperature Sensor channel returns a more readable value to the application.

   ❑ Under Data Neighborhood, select **NI-DAQmx Global Virtual Channels** and select **Temperature Sensor**.

   ❑ In the Settings tab in the Voltage Input Setup section, click the Custom Scaling pull-down menu and select **Create New**.

   ❑ Use the following settings for the custom scale:

   - **Scale Type**: **Linear**
   - **Name**: `Temperature Scale`

   ❑ Click **Finish**. The DAQ Assistant appears.

2. Configure the numerical scaling for your scale in the DAQ Assistant.

   ❑ Since the voltage to degrees Celsius correspondence is one to 100, enter `100` for the **Slope** value. Because there is no offset, leave the **Y-Intercept** value at the default, `0`.

   ❑ Set **Pre-Scaled** to **Volts** in the Units section. In the **Scaled** field, enter `Deg C`.

   ❑ Click the **OK** button.

3. Configure the Voltage Input Setup section.

   ❑ For the Temperature Sensor channel, change the maximum and minimum input values to `40` and `0`, respectively. For measuring the air temperature or the temperature of your finger, 0 to 40 degrees Celsius is an appropriate range.

4. Click the **Run** button for the Temperature Sensor NI-DAQmx Global Virtual Channel to acquire data. Notice that the data is now in the range of 20 to 30 degrees Celsius.

5. Click **Stop** to stop the measurement.

6. Click the **Save** button to save the NI-DAQmx global virtual channel settings.

7. Go back to **MyVoltageTask** and **Run** the task. Notice how the Graph display now has the updated scaling for the Temperature Sensor channel.

8. Click **Stop** to stop the task.

9. Save the NI-DAQmx Task.

10. Exit MAX.

Part IV: Generate code by using the DAQmx Task in LabVIEW

1. Use the task you create in MAX to generate LabVIEW code.

    ❑ Open a blank VI.

    ❑ Place a DAQmx Task Name control on the front panel.

    ❑ Click the pull-down menu and select **MyVoltageTask**, which you just created in MAX.

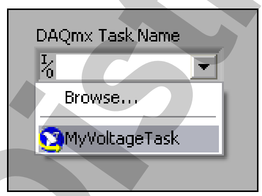


Figure 2-11. DAQmx Task Name Control

    ❑ Right-click the DAQmx Task Name control and select **Generate Code»Convert to Express VI**.

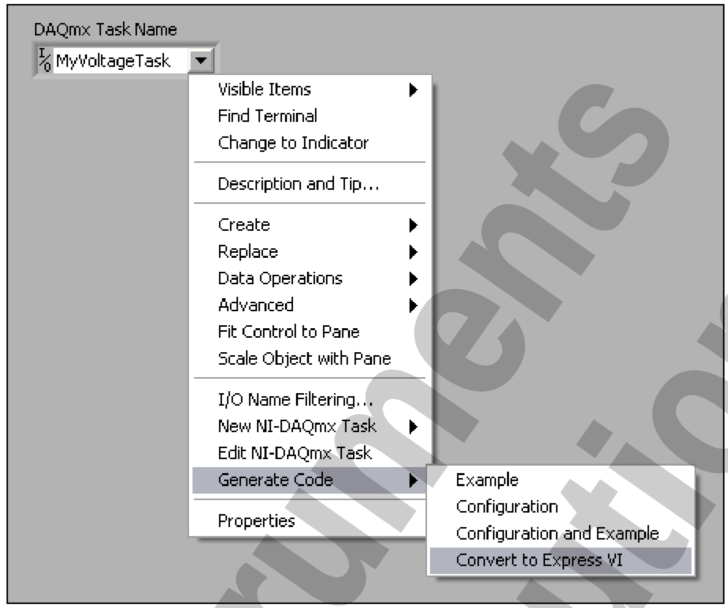


Figure 2-12. DAQmx Task Name Control Shortcut Menu

2. Confirm the settings you previously entered in the DAQ Assistant.

   ❑ Click **OK** and LabVIEW will generate the appropriate Express VI.

   ❑ If prompted to place a loop around the code, click **Yes**.

3. Create a graph indicator.

   ❑ Right-click the data output of the DAQ Assistant Express VI and select **Create»Graph Indicator**.

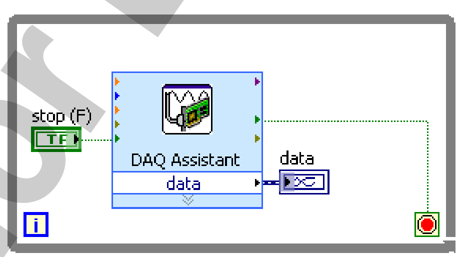


Figure 2-13. Generated Block Diagram

4. Run the VI.

   ❑ Click **Run** on the front panel.

   ❑ Observe the results.

5. Close the VI.

- ❑ Click **Stop** when desired.
- ❑ Close the VI. Do not save changes.

End of Exercise 2-3

Notes

Notes

3

Analog Input Exercises

Exercise 3-1 Differential, RSE, and NRSE

Objective

To learn how to choose a grounding mode for a measurement system and how to properly connect signals to that measurement system.

Part I

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. Which measurement system would you use when connecting three outputs of the instrument to a DAQ device in the computer? (circle one)

Differential RSE NRSE

Figure 3-1 shows the instrument and a 68-pin layout for a PCI-6221. 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

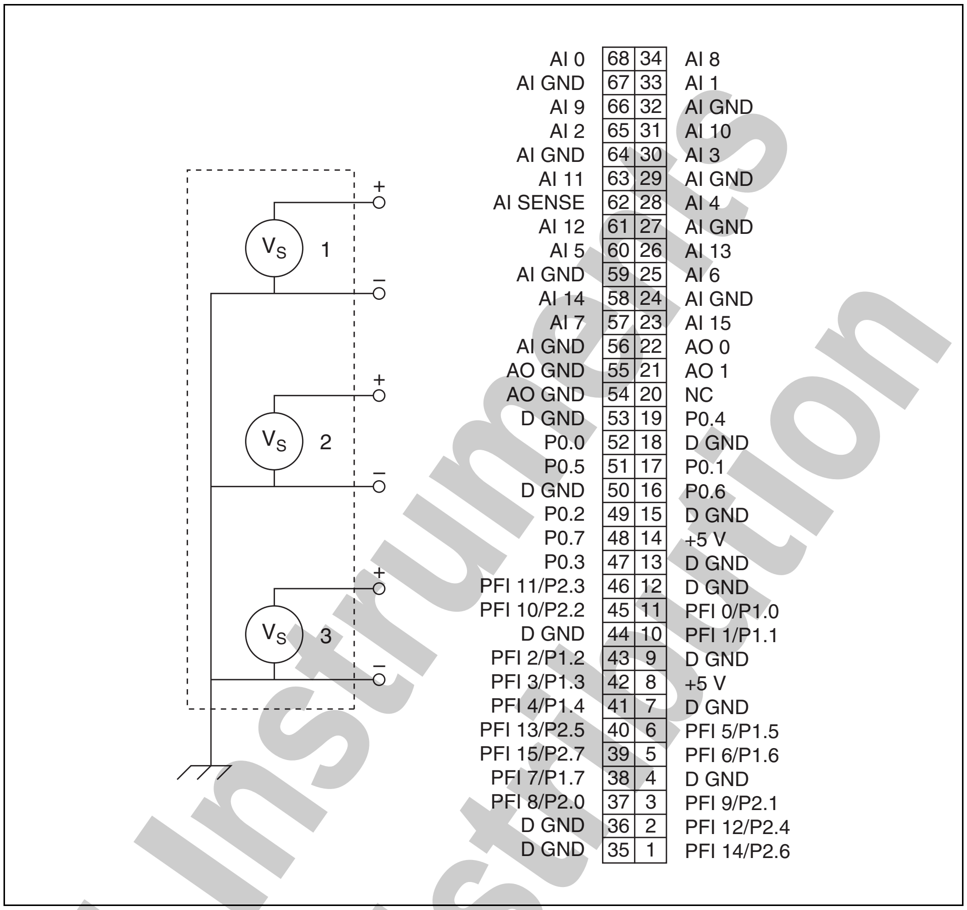


Figure 3-1. PCI-6221 68-pin Layout

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.

Part II

Assume you have three batteries. Which measurement system would you use when connecting the outputs of the batteries to a DAQ device in the computer? (circle one)

Differential RSE NRSE

Figure 3-2 shows the batteries and a 50-pin layout for a PCI-MIO-16E-4. 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

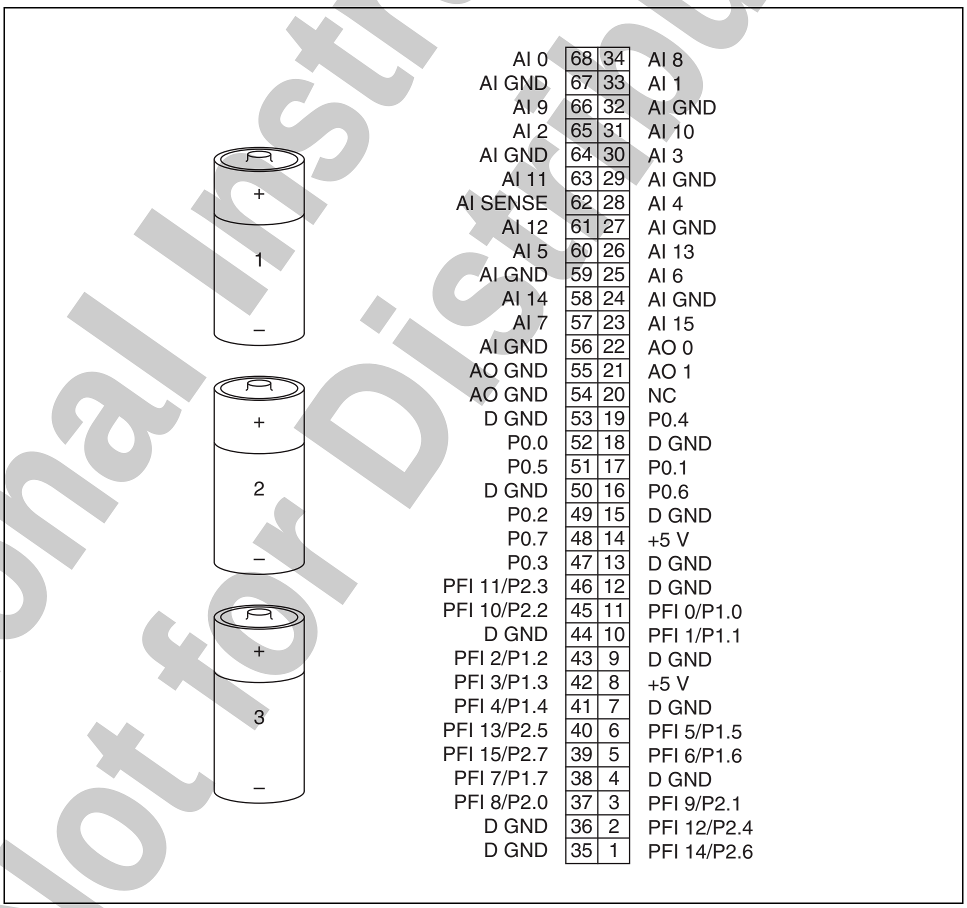


Figure 3-2. PCI-MIO-16E-4 50-pin Layout

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 negative terminal 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 3-1

Exercise 3-2 Sampling Rate and Aliasing

Objective

To demonstrate aliasing and the effects of sampling rate on an input signal.

BNC-2120 Configuration

1. Connect the BNC connector of channel 0 in the Analog Outputs section to the BNC connector of channel 1 in the Analog Inputs section.
2. Verify that the **BNC/Thermocouple** switch is in the `BNC` position.

Implementation

1. Use the Sampling Rate Example VI to demonstrate aliasing and the effects of sampling rate on an input signal.

   ❑ Open the Sampling Rate Example VI located in the `<Exercises>\DAQ and Signal Conditioning\Aliasing` directory. The front panel in Figure 3-3 appears.

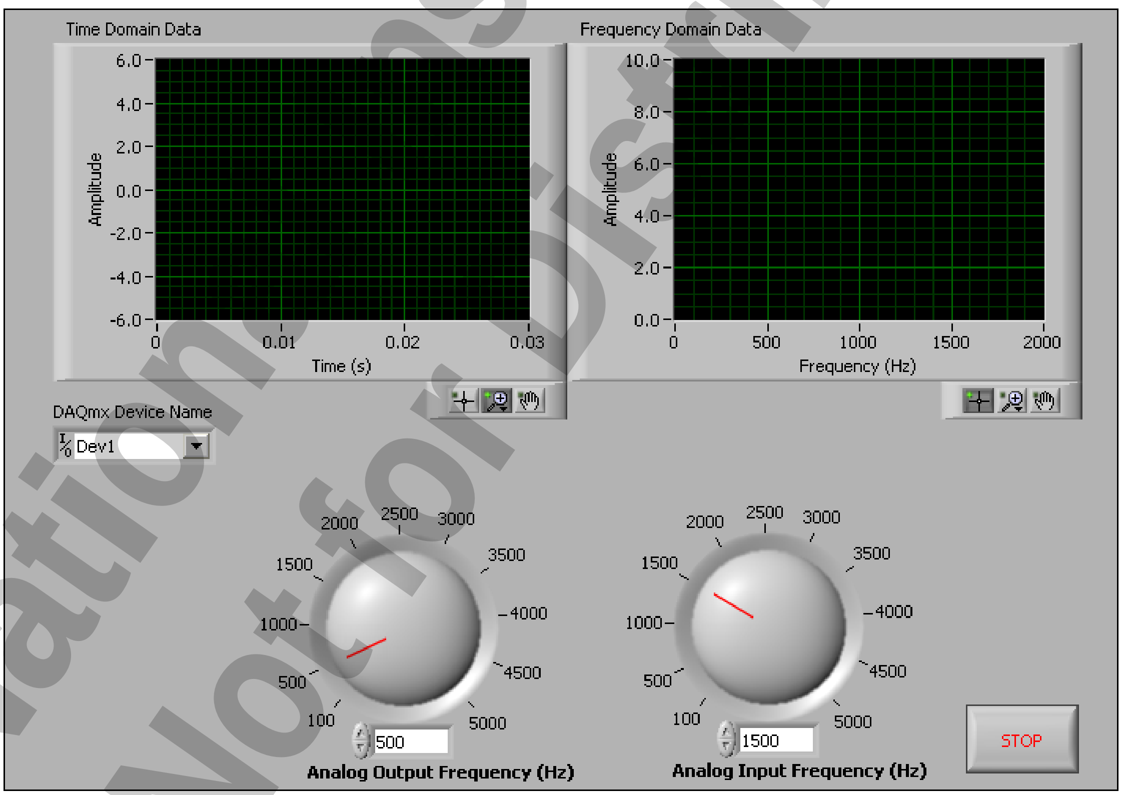


Figure 3-3. Sampling Rate Example VI Front Panel

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

❑ Set the front panel controls with the following values:

- **Analog Output Frequency (Hz)**: `500`
- **Analog Input Frequency (Hz)**: `1500`

❑ 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 sampling rate of 1,500 Hz, the Nyquist frequency is 750 Hz. This implies that the sampling 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.

❑ Stop the VI. Click the Zoom button on the time-domain data graph and zoom in on the x-axis, as shown in Figure 3-4.

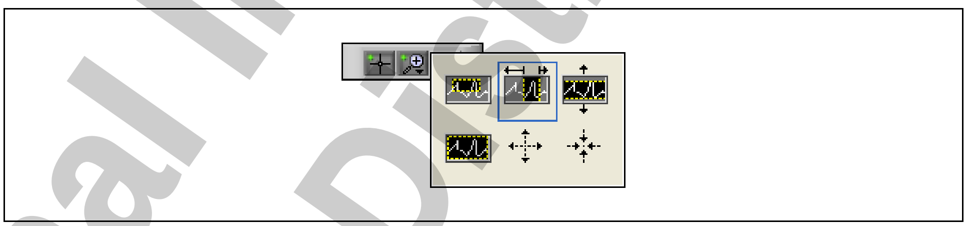


Figure 3-4. Graph Palette - Zoom Button

The data looks like a triangle 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.

- ❑ Run the VI. Increase the Analog Input Frequency to `5,000` Hz. Stop the VI and click the Zoom button on the time-domain data graph and zoom in on the x-axis. The shape of the time domain signal looks like a smooth sine wave. Increasing the sampling rate ten times faster than the signal you are trying to acquire more accurately represents the shape of the signal. In general, try to acquire a signal five to ten times faster than the highest frequency in the signal you are trying to capture.

- ❑ Decrease the Analog Input Frequency to `1,000` Hz. Run the VI. You are now sampling at $2f_n$. When you sample at $2f_n$, the time-domain signal looks like a triangle wave. You are accurately representing the frequency of the signal you are acquiring, but not the shape. If you increase the Analog Input Frequency just above 1,000 Hz, the frequency of the signal is also represented in the frequency-domain graph. As you slowly increase the Analog Input Frequency knob, you can see the frequency of the acquired signal. This implies that you must sample greater than $2f_n$ to accurately represent the frequency domain of an acquired signal.

- ❑ Decrease the Analog Input Frequency to `750` Hz. f_n is equal to 375 Hz, which is lower than the acquired signal frequency. Although the time-domain data waveform appears sinusoidal in nature, the signal is aliased, which is indicated by the incorrect frequency displayed on the frequency domain plot. The alias frequency you see 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 graph.

The frequency domain 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.

- ❑ Stop and close the VI. Do not save changes.

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

End of Exercise 3-2

Exercise 3-3 Voltmeter VI

Objective

To acquire an analog signal using the DAQmx API to implement a continuous software timed measurement.

BNC-2120 Configuration

1. Verify that the **BNC/Temp. Ref.** switch is in the `Temp. Ref.` position. This connects analog input channel 0 of the DAQ device to the temperature sensor on the BNC-2120.

Implementation

1. Open a blank VI.
2. Save the VI as `<Exercises>\DAQ and Signal Conditioning\Analog Input\Voltmeter.vi`.
3. Create the block diagram to perform a continuous software-timed acquisition, as shown in Figure 3-5, using the following items:

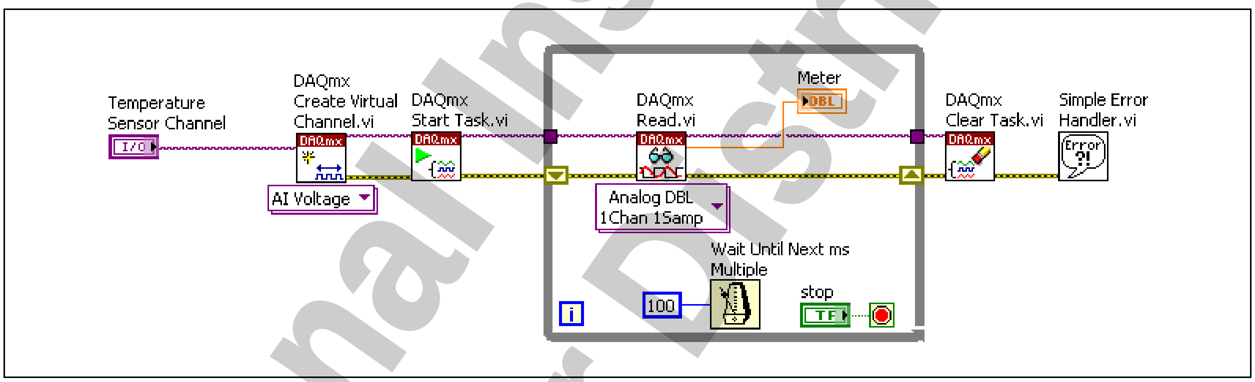
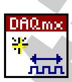


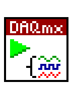
Figure 3-5. Voltmeter VI Block Diagram



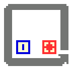
❑ DAQmx Create Virtual Channel VI—This VI creates a virtual channel of the type you specify in the instance pull-down menu.

– Select the **Analog Input»Voltage** instance from the pull-down menu.

– Right-click the physical channels input and select **Create» Control**. Rename the control as `Temperature Sensor Channel`.

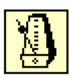


❑ DAQmx Start Task VI—This VI starts the measurement task.



- ❑ While Loop—The While Loop repeats the subdiagram inside it until the conditional terminal receives a particular Boolean value.
  - – Wire the error out terminal of the DAQmx Start Task VI to the left border of the While Loop.
  - – Right-click the tunnel and select **Replace with Shift Register**.
  - – Create the stop control from the Loop Condition terminal of the While Loop.

- ❑ DAQmx Read VI—This VI performs the measurement reading you specify in the instance pull-down menu.
  - – Select the **Analog»Single Channel»Single Sample»DBL** instance. This instance returns a single analog sample of double-precision, floating-point numeric data from a single channel.
  - – Right-click the data output and select **Create»Indicator**. Rename the indicator as `Meter`.



- ❑ Wait Until Next ms Multiple function—This function causes the loop to execute every 100 ms.
  - – Right-click the millisecond multiple input and select **Create» Constant**. Set the constant to `100`.

- ❑ DAQmx Clear Task VI—This VI clears the task. Before clearing, this VI stops the task, if necessary, and releases any resources the task reserved.

- ❑ Simple Error Handler VI—In the event of an error, this VI displays a dialog box with information about the error and where it occurred.

4. Modify the front panel.
   - ❑ Display the front panel and right-click the Meter indicator and select **Replace»Numeric»Meter**.
     - – Configure the Meter indicator to display a range of `0` to `0.4`.

❑ Arrange the front panel as shown in Figure 3-6.

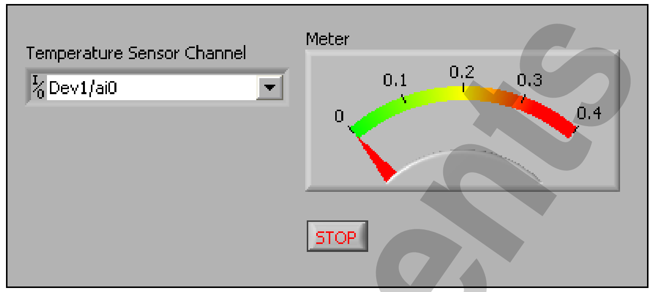


Figure 3-6. Voltmeter VI Front Panel

5. Save the VI.

Test

1. View the voltage output.

   ❑ Set the **Temperature Sensor Channel** to `Dev1/ai0`.

   ❑ Run the VI. The meter displays the voltage of the temperature sensor outputs.

   ❑ Place your finger on the temperature sensor and notice that the voltage increases.

2. Stop the VI.

3. Save and close the VI when finished.

End of Exercise 3-3

Exercise 3-4 Finite Acquisition with Analysis

Objective

To acquire an array of data using finite buffered configuration and to analyze this data for maximum and minimum values.

BNC-2120 Configuration

1. Connect the **Sine/Triangle** BNC connector in the Function Generator section to the BNC connector of channel 1 in the Analog Inputs section.
2. Verify that the **Sine/Triangle** switch is in the `Sine` position.
3. Verify that the **BNC/Thermocouple** switch is in the `BNC` position.
4. Verify that the **FS/GS** switch underneath the BNC connector of channel 1 is in the `GS` position.

Finite Buffered Acquisition

1. Open a blank VI.
2. Save the VI as `<Exercises>\DAQ and Signal Conditioning\Analog Input\Finite Buffered Acquisition.vi`.
3. Create the block diagram to perform a buffered acquisition of a finite number of samples, as shown in Figure 3-7, using the following items:

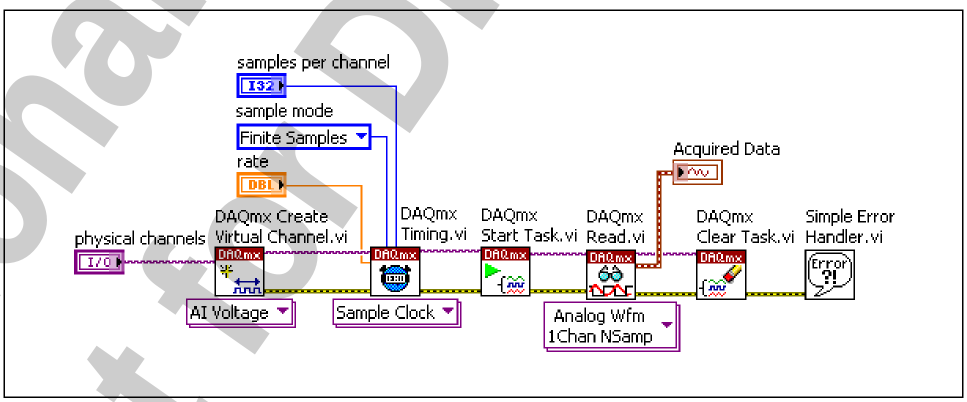
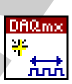


Figure 3-7. Finite Buffered Acquisition VI Block Diagram



❑ DAQmx Create Virtual Channel VI

– Select the **Analog Input»Voltage** instance from the pull-down menu.

– Right-click the physical channels terminal and select **Create» Control**.

❑ DAQmx Timing VI—This VI configures the number of samples to acquire or generate and creates a buffer when needed.

– Select the **Sample Clock** instance from the pull-down menu.

– Right-click the samples per channel and rate inputs and select **Create»Control**.

– Right-click sample mode and select **Create»Constant**. Set the constant to **Finite Samples**.

❑ DAQmx Start Task VI

❑ DAQmx Read VI

– Select the **Analog»Single Channel»Multiple Samples» Waveform** instance from the pull-down menu.

– Right-click the data output and select **Create»Indicator**. Rename the indicator as `Acquired Data`.

❑ DAQmx Clear Task VI

❑ Simple Error Handler VI

4. Arrange the items on the front panel similar to Figure 3-8. Do not alter the x- and y-axis scales.

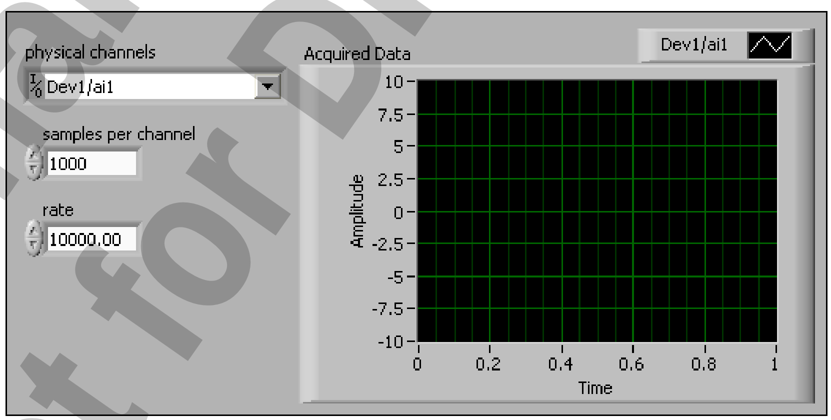


Figure 3-8. Finite Buffered Acquisition VI Front Panel

❑ Right-click the Acquired Data indicator and select **Replace» Graph»Waveform Graph**.

5. Save the VI.

Test

1. Output a sine wave with a frequency of 100 Hz from the function generator on the BNC-2120.

   ❑ On the BNC-2120, set the **Frequency Selection** switch in the Function Generator section to the leftmost position.

   ❑ Turn the **Frequency Adjust** dial on the BNC-2120 to `LO`.

2. Set the front panel controls of the Finite Buffered Acquisition VI to the following values:

   - **Physical Channels**: `Dev1/ai1`
   - **Samples Per Channel**: `1000`
   - **Rate**: `10000`

3. Run the VI.

Note If your data does not appear, troubleshoot the VI. If that doesn't reveal the problem, then try to build a task in MAX. If you still have trouble, ask for additional assistance.

Waveform Analysis

1. Save an additional copy of this VI.

   ❑ Select **File»Save As**.

   ❑ Select the **Copy»Substitute copy for original** option and click **Continue**.

   ❑ Save the VI as `<Exercises>\DAQ and Signal Conditioning\Analog Input\Finite Buffered Acquisition with Analysis.vi`.

2. Add waveform analysis to the finite acquisition by analyzing the data after it has been acquired. Modify the block diagram, as shown in Figure 3-9.

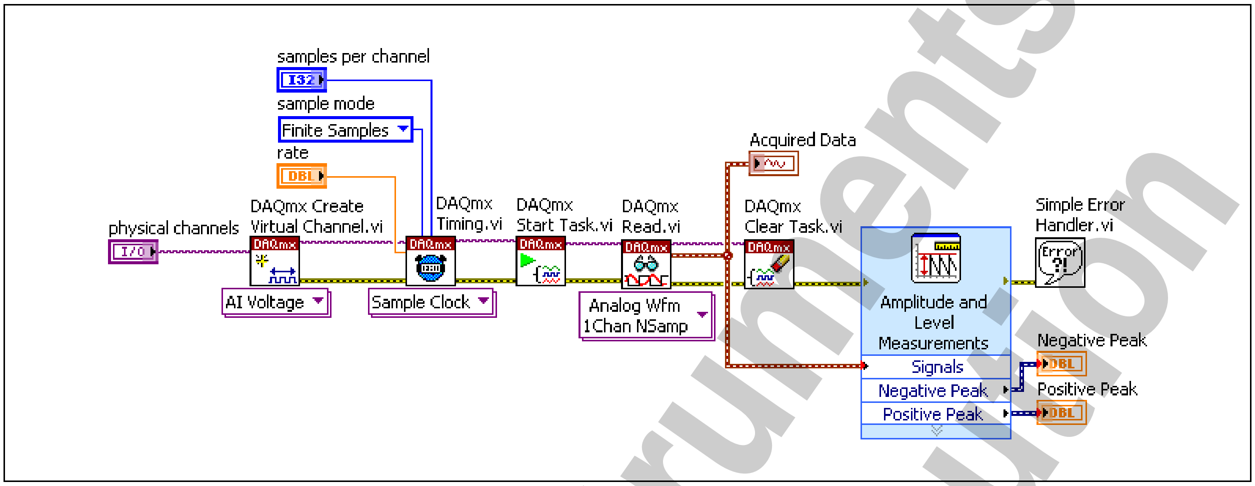
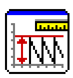


Figure 3-9. Finite Buffered Acquisition with Analysis VI Block Diagram



- ❑ Amplitude and Level Measurements Express VI—In this exercise, the Amplitude and Level Measurements Express VI determines the maximum and minimum values in a signal.
  - – In the Configure Amplitude and Level Measurements dialog box, place checkmarks in the **Maximum peak** and **Minimum peak** checkboxes.
  - – Click **OK** to apply the changes and close the dialog box.
  - – Create numeric indicators for the Negative Peak and Positive Peak terminals.
- ❑ Move the Simple Error Handler VI so that the error clusters run through all the VIs and enforce dataflow.

3. Save the VI.

Test

1. From the front panel, run the VI and observe the resulting values.
2. Adjust the amplitude of the sine wave by turning the **Amplitude Adjust** dial on the BNC-2120, and observe how it affects the results of the VI the next time you run it.
3. Experiment with other amplitude and level measurements.
   - ❑ Double-click the Amplitude and Level Measurements Express VI and select different measurements to output. Click **OK**.
   - ❑ Create indicators for these values.
   - ❑ Run the VI to experiment with different amplitude and level measurements.
4. Save and close the VI.

End of Exercise 3-4

Exercise 3-5 Continuous Buffered Acquisition and Logging

Objective

To continuously acquire data from a DAQ device and log the data to a file.

BNC-2120 Configuration

1. Connect the **Sine/Triangle** BNC connector in the Function Generator section to the BNC connector of channel 1 in the Analog Inputs section.
2. Verify that the **Sine/Triangle** switch is in the `Sine` position.
3. Verify that the **BNC/Thermocouple** switch is in the `BNC` position.
4. Verify that the **FS/GS** switch underneath the BNC connector of channel 1 is in the `GS` position.

Continuous Buffered Acquisition

1. Open a blank VI.
2. Save the VI as `<Exercises>\DAQ and Signal Conditioning\Analog Input\Continuous Buffered Acquisition.vi`.
3. Create the block diagram to perform a continuous buffered acquisition, as shown in Figure 3-10, using the following items:

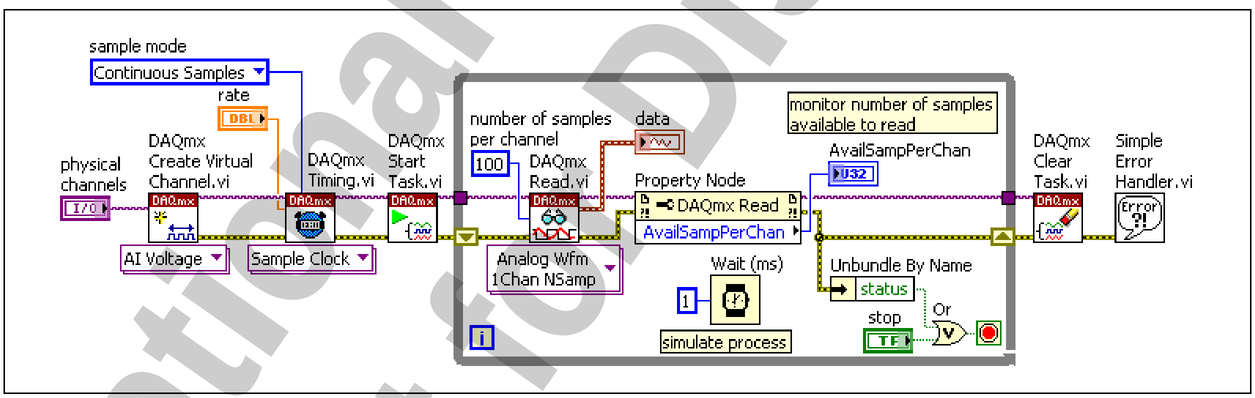
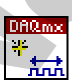


Figure 3-10. Continuous Buffered Acquisition VI Block Diagram



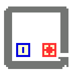
❑ DAQmx Create Virtual Channel VI

– Select the **Analog Input»Voltage** instance from the pull-down menu.

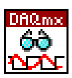
– Right-click the physical channels terminal and select **Create»Control**.

- ❑ DAQmx Timing VI
  - – Select the **Sample Clock** instance from the pull-down menu.
  - – Right-click the rate input and select **Create»Control**.
  - – Right-click sample mode and select **Create»Constant**. Set the constant to **Continuous Samples**.

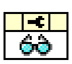
- ❑ DAQmx Start Task VI



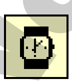
- ❑ While Loop
  - – Wire the error out terminal of the DAQmx Start Task VI to the left border of the While Loop.
  - – Right-click the tunnel and select **Replace with Shift Register**.
  - – Create the stop control from the Loop Condition terminal of the While Loop.



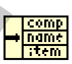
- ❑ DAQmx Read VI
  - – Select the **Analog»Single Channel»Multiple Samples» Waveform** instance from the pull-down menu.
  - – Right-click the number of samples per channel input and select **Create»Constant**. Set the constant at `100`.
  - – Right-click the data output and select **Create»Indicator**.



- ❑ DAQmx Read Property Node—This Property Node configures additional settings for reading a channel.
  - – Set the DAQmx Read Property Node to **Status»Available Samples Per Channel**.
  - – Right-click the Available Samples Per Channel output and select **Create»Indicator**.



- ❑ Insert a Wait (ms) function with a constant of `1` to represent a process in the loop.



- ❑ Unbundle by Name function

- ❑ Or function

- ❑ DAQmx Clear Task VI

- ❑ Simple Error Handler VI

4. Arrange the front panel as shown in Figure 3-11.

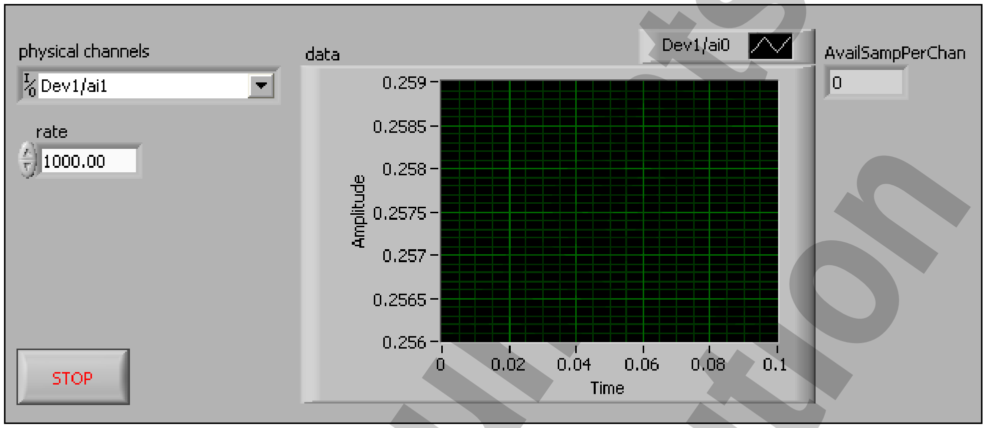


Figure 3-11. Continuous Buffered Acquisition VI Front Panel

- ❑ Replace the data indicator with a Waveform Graph.

5. Save the VI.

Test

1. Set the front panel controls of the Continuous Buffered Acquisition VI with the following values:
   - **Physical Channels**: `Dev1/ai1`
   - **Rate**: `100000`
2. Run the VI.
3. Monitor the AvailSampPerChan indicator. If you acquire data more rapidly than you read it, the buffer will fill and eventually overflow.

   With a sampling rate of 100,000 Hz and a simulated process of 1 ms, the buffer will probably overflow and stop the VI with an error.
4. Decrease the sampling rate to `1000` Hz and run the VI again. Notice the effect on the AvailSampPerChan indicator.
5. Stop the VI.

Streaming to Disk

1. Modify the VI to stream the acquired data to disk.

   ❑ Create a copy of the VI by selecting **File»Save As»Copy» Substitute copy for original»Continue**.

   ❑ Save the VI as `Continuous Buffered Acquisition with Logging.vi` in the `<Exercises>\DAQ and Signal Conditioning\Analog Input` directory.

   ❑ Modify the block diagram as shown in Figure 3-12 using the following items:

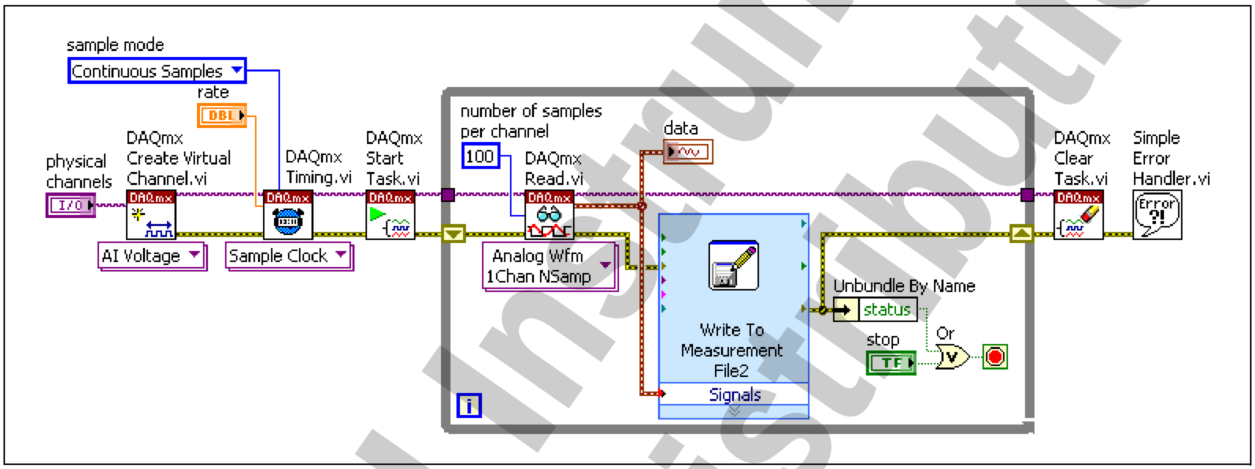


Figure 3-12. Continuous Buffered Acquisition with Logging VI Block Diagram

   ❑ Delete the DAQmx Read Property node and the Wait (ms) function.

   – Reconnect the DAQmx Task reference between the DAQmx Read and DAQmx Clear Task VIs.

   – Delete the AvailSampPerChan indicator.

   ❑ Expand the While Loop.

   ❑ Write to Measurement File Express VI—This Express VI writes LabVIEW measurement data to a file.

- ❑ In the Configure Write To Measurement File dialog box, select the settings shown in Figure 3-13.

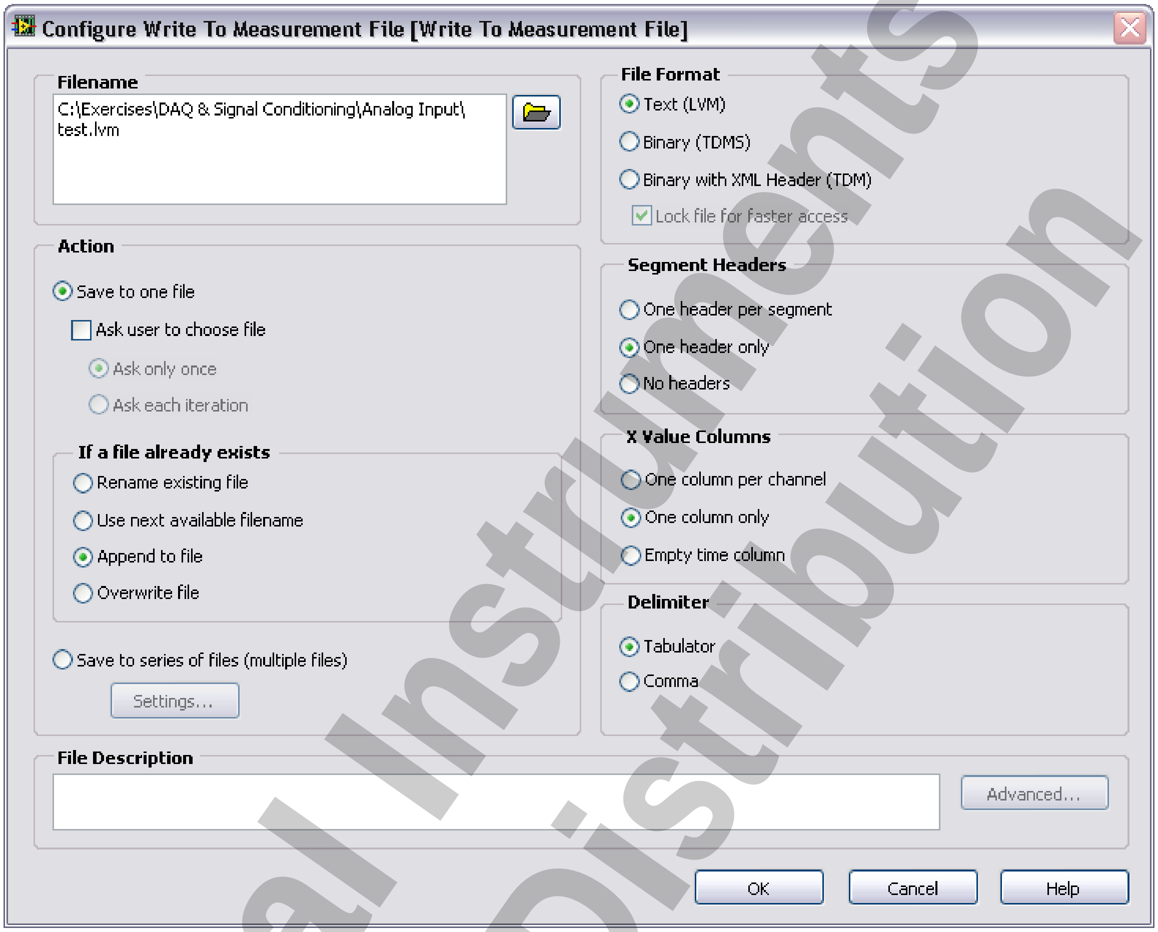


Figure 3-13. Configure Write To Measurement File Dialog

- ❑ Click **OK**.
- ❑ Reconnect the error clusters.
- ❑ Wire the data wire to the Signals input, as shown in Figure 3-12.

2. Run the VI for a few seconds, and click the **Stop** button.

   The VI should have created the `text.lvm` file in the `<Exercises>\DAQ and Signal Conditioning\Analog Input` directory and logged the acquired data to this file.

3. Save the VI.

Test

1. Open a blank VI.
2. Save the VI as `Read Data File.vi` in the `<Exercises>\DAQ and Signal Conditioning\Analog Input` directory.
3. Create the block diagram to build a VI that reads the LabVIEW Measurement File, as shown in Figure 3-14, using the following items:

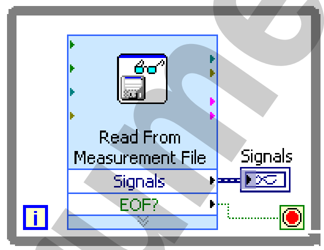
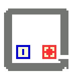


Figure 3-14. Read Data File VI Block Diagram



❑ While Loop

❑ Read from Measurement File VI—This Express VI reads LabVIEW measurement data from a file.

– In the **Configure Read From Measurement File** dialog box, select the settings shown in Figure 3-15.

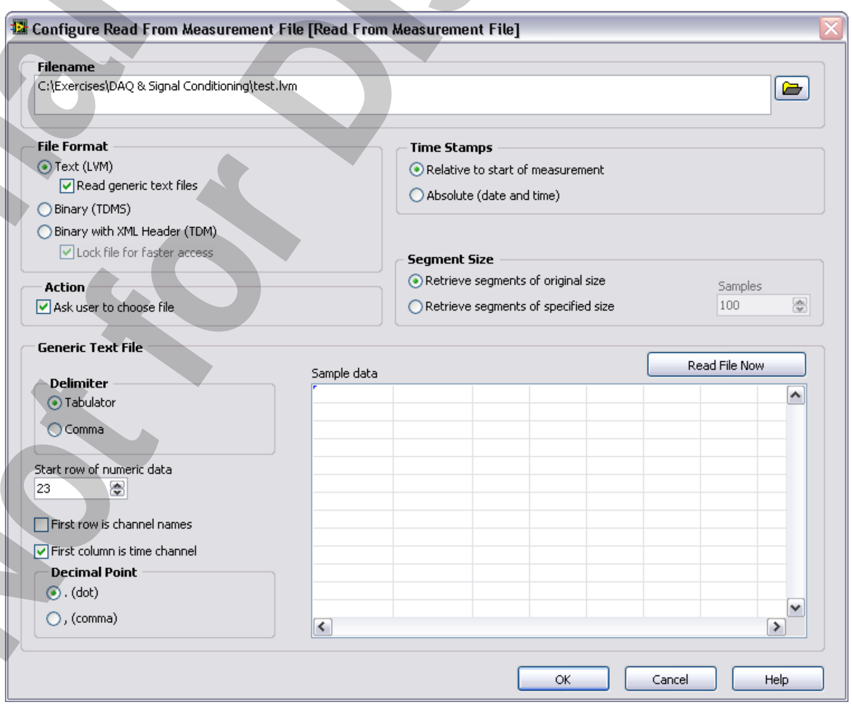


Figure 3-15. Configure Read From Measurement File Dialog

- Click **OK**.
- Right-click the **Signals** output and select **Create»Graph Indicator** from the shortcut menu.
- Resize the Read From Measurement File Express VI to display another element. Set the element to **EOF?**.
- Wire the EOF? element to the loop condition terminal of the While Loop.

4. Replace the waveform graph with a waveform chart.

   ❑ Go to the front panel.

   ❑ Right-click the Signals graph indicator and select **Replace» Modern»Graph»Waveform Chart**.

5. Run the VI. Select the test.lvm file in the <Exercises>\DAQ and Signal Conditioning\Analog Input directory. The signal that you acquired and logged in the Continuous Acquisition with File Logging VI now appears in the waveform chart.

6. Save the VI.

Challenge: Explore TDMS File Logging

Technical Data Management Streaming (TDMS) is a binary file format that allows for high-speed data logging. When you enable TDMS data logging, NI-DAQmx can stream data directly from the device buffer to the hard disk. NI-DAQmx improves performance and reduces the disk footprint by writing raw data to the TDMS file, including the scaling information separately for use when reading back the TDMS file. You can also read data while logging to disk.

1. Open the NI Example Finder by selecting **Help»Find Examples**.
2. Open TDMS Streaming - Cont Log and Read Data.vi in the **Hardware Input and Output»DAQmx»Analog Measurements» Voltage** folder.
3. Explore the block diagram. In particular, pay attention to the DAQmx Configure Logging (TDMS) VI. Right-click this VI and select **Help** for more information.
4. Click the pull-down menu of the DAQmx Read VI and select the **Analog»Single Channel»Multiple Samples»Waveform** instance.

5. On the front panel, set the Physical Channel control to `Dev1/ai1`.

6. Set the TDMS File Path control to `<Exercises>\DAQ and Signal Conditioning\Analog Input\test.tdms`.

7. Observe the Logging Options section and verify that View File When Done? is set to True.

8. Run the VI. After a few seconds, click the **Stop** button.

9. The TDMS File Viewer window appears. Explore the Properties, Values (table), and Analog Values (graph) tabs. Expand the File contents section to view the grouping of your data. Click **Quit** when done.

Note There is a free Microsoft Excel add-in for loading TDMS files into Excel available on `ni.com`. TDMS file logging is covered in more depth in the *LabVIEW Core 2* course. For more information about TDMS files, refer to the *LabVIEW Help* and `ni.com`.

10. Close the VI. Do not save changes.

End of Exercise 3-5

Exercise 3-6 Triggered Continuous Buffered Acquisition

Objective

To use a digital trigger to start a continuous data acquisition operation.

BNC-2120 Configuration

1. In the Timing I/O section, wire the **PULSES** or **CLK** screw terminal to the **PFI 1** screw terminal.

Note Some models of the BNC-2120 have a **PULSES** terminal and some models have a **CLK** terminal instead. They are functionally the same.

2. Connect the **Sine/Triangle** BNC connector in the Function Generator section to the BNC connector of channel 1 in the Analog Inputs section.

3. Verify that the **Sine/Triangle** switch is in the `Sine` position.

4. Verify that the **BNC/Thermocouple** switch is in the `BNC` position.

5. Verify that the **FS/GS** switch underneath the BNC connector of channel 1 is in the `GS` position.

Implementation

1. Update the Continuous Buffered Acquisition VI to include digital triggering.

   ❑ Open `<Exercises>\DAQ and Signal Conditioning\Analog Input\Continuous Buffered Acquisition with Logging.vi`.

   ❑ Select **File»Save As»Copy»Substitute copy for original** and save the VI as `Triggered Continuous Buffered Acquisition with Logging.vi` in the `<Exercises>\DAQ and Signal Conditioning\Analog Input` directory.

❑ Insert triggering as shown in Figure 3-16 by right-clicking the task name wire between the DAQmx Timing VI and the DAQmx Start VI and selecting **Insert»DAQmx - Data Acquisition Palette»DAQmx Trigger VI**.

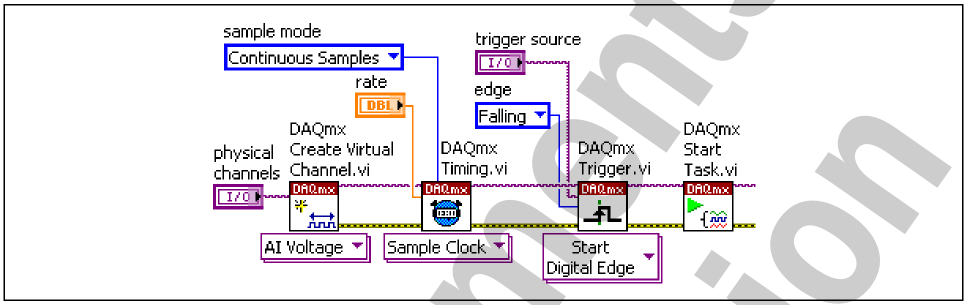


Figure 3-16. Triggered Continuous Buffered Acquisition VI Block Diagram

❑ Click the pull-down menu of the DAQmx Trigger VI and select the **Start»Digital Edge** instance.

❑ Delete the error cluster wire residing under the DAQmx Trigger VI and wire the error cluster through the DAQmx Trigger VI.

2. Configure the DAQmx Trigger VI to allow the user to select the trigger source.

   ❑ Right-click the source terminal of the DAQmx Trigger VI and select **Create»Control**. Rename the control as `trigger source`.

3. Set the task to trigger on the falling edge, so that the task triggers when the Quadrature Encoder knob is turned.

   ❑ Right-click the edge terminal of the DAQmx Trigger VI and select **Create»Constant**. Set the edge constant to **Falling**.

4. Save the VI.

Test

1. Set the front panel controls of the Triggered Continuous Buffered Acquisition with Logging VI with the following values:
   - **Physical Channels**: `Dev1/ai1`
   - **Rate**: `1000`
   - **Trigger Source Control**: `/Dev1/PFI1`
2. Run the VI.
   - ❑ Notice how there are no samples in the graph. The VI is waiting to detect a falling edge on the `/Dev1/PFI1` channel.
   - ❑ Turn the Quadrature Encoder knob to begin acquisition.

Note When the Quadrature Encoder knob is not moving, the BNC-2120 outputs 5 V on the **PULSES** or **CLK** terminal. Each time you turn the Quadrature Encoder knob, the BNC-2120 outputs four falling edges on the **PULSES** or **CLK** terminal.

3. Stop and close the VI when finished.

End of Exercise 3-6

Notes

Notes

4

Analog Output Exercises

Exercise 4-1 Continuous Single-Point Generation

Objective

To build a VI that generates a variable voltage on an analog output channel.

Scenario

In this exercise, you create the Variable Servo Fan VI to control the speed of a fan with a variable voltage. To implement this capability, you must be able to continuously update the voltage generated by the analog output channel on the DAQ device.

Note The speed of a servomotor is proportional to the voltage with which it is driven. As the input voltage increases, the speed of the motor increases.

BNC-2120 Configuration

1. Connect the BNC connector of channel 0 in the Analog Outputs section to the BNC connector of channel 2 in the Analog Inputs section.

2. Verify that the **FS/GS** switch underneath the BNC connector of channel 2 is in the `GS` position.

Implementation

1. Open a blank VI.

2. Save the VI as `Variable Servo Fan.vi` in the `<Exercises>\DAQ and Signal Conditioning\Analog Output` directory.

3. Create the block diagram to perform a software-timed analog generation, as shown in Figure 4-1, using the following items:

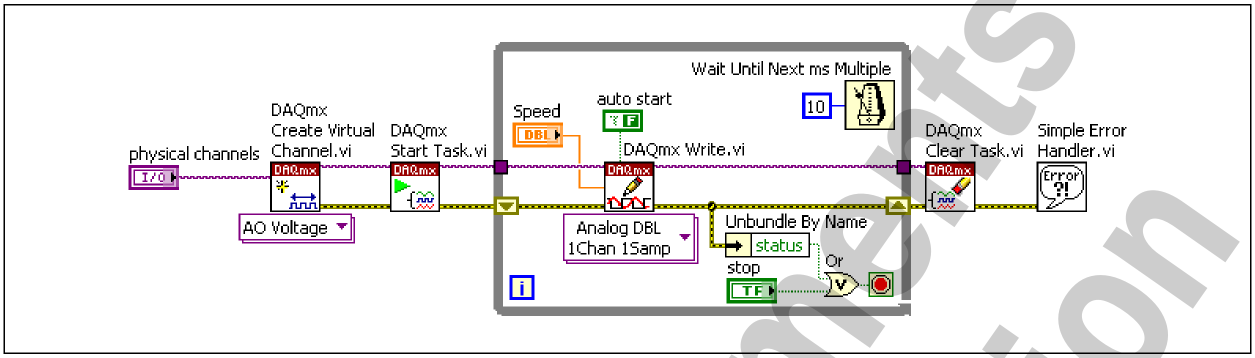
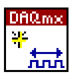
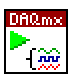


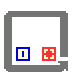
Figure 4-1. Variable Servo Fan VI Block Diagram



- ❑ DAQmx Create Virtual Channel VI
  - – Select the **Analog Output»Voltage** instance from the pull-down menu.
  - – Right-click the physical channels input and select **Create»Control**.

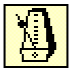


- ❑ DAQmx Start Task VI

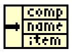


- ❑ While Loop
  - – Wire the error out terminal of the DAQmx Start Task VI to the left border of the While Loop.
  - – Right-click the tunnel and select **Replace with Shift Register**.
  - – Create the stop control from the Loop Condition terminal of the While Loop.

- ❑ DAQmx Write VI—This VI writes samples to the task or virtual channels you specify.
  - – Select the **Analog»Single Channel»Single Sample»DBL** instance from the pull-down menu.
  - – Right-click the data input and select **Create»Control**. Rename the control as `Speed`.
  - – Right-click auto start and select **Create»Constant**. Set the constant to **false**. Because this VI uses the Start Task VI to start the task, you must change the auto start constant on the DAQmx Write VI to false.



- ❑ Wait Until Next ms Multiple function
  - – Right-click the millisecond multiple input and select **Create»Constant**. Enter `10` as the value for the constant.



- ❑ Unbundle by Name function

- ❑ Or function

- ❑ DAQmx Clear Task VI

- ❑ Simple Error Handler VI

4. Modify the front panel as shown in Figure 4-2.

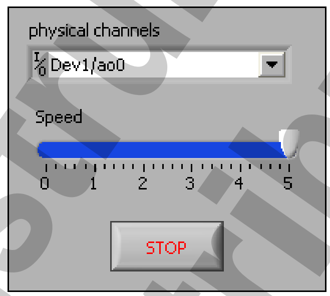


Figure 4-2. Variable Servo Fan VI Front Panel

- ❑ Replace the Speed numeric control with a horizontal pointer slide by right-clicking the numeric control and selecting **Replace»Numeric»Horizontal Pointer Slide**.

5. Save the VI.

Test

1. Monitor analog input channel 2 of the DAQ device.

   ❑ Open the NI Example Finder by selecting **Help»Find Examples**.

   ❑ Navigate to the **Hardware Input and Output»DAQmx»Analog Measurements»Voltage** folder and open `Cont Acq&Graph Voltage-Int Clk.vi`.

   ❑ Set the front panel controls of the Cont Acq&Graph Voltage-Int Clk VI with the following values:

   - **Physical Channels**: `Dev1/ai2`
   - **Sample Rate (Hz)**: `1000`
   - **Samples to Read**: `250`

   ❑ Run the Cont Acq&Graph Voltage-Int Clk VI. The VI is now acquiring and displaying the voltage measured by analog input channel 2.

2. Test the Variable Servo Fan VI.

   ❑ On the Variable Servo Fan VI, set the Physical Channels control to `Dev1/ao0`.

   ❑ Run the Variable Servo Fan VI. Adjust the Speed control and observe the voltage that is displayed on the waveform chart of the Cont Acq&Graph Voltage-Int Clk VI.

3. Stop and close the VIs when finished.

End of Exercise 4-1

Exercise 4-2 Finite Buffered Generation

Objective

To build a VI that generates a waveform on an analog output channel.

BNC-2120 Configuration

1. Connect the BNC connector of channel 0 in the Analog Outputs section to the BNC connector of channel 2 in the Analog Inputs section.

2. Verify that the **FS/GS** switch underneath the BNC connector of channel 2 is in the `GS` position.

Implementation

1. Open a blank VI.

2. Save the VI as `Finite Buffered Generation.vi` in the `<Exercises>\DAQ and Signal Conditioning\Analog Output` folder.

3. Create the block diagram to generate a finite number of samples, as shown in Figure 4-3, using the following items:

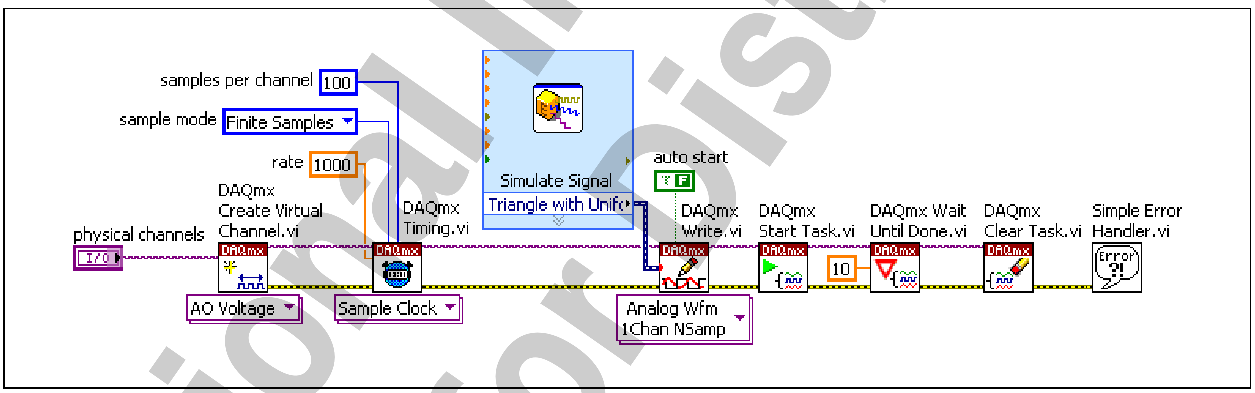


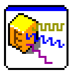
Figure 4-3. Finite Buffered Generation VI Block Diagram

❑ DAQmx Create Virtual Channel VI

– Select the **Analog Output»Voltage** instance from the pull-down menu.

– Right-click the physical channels input and select **Create» Control**.

- ❑ DAQmx Timing VI
  - – Right-click the rate input and select **Create»Constant**. Set the constant to `1000`.
  - – Right-click sample mode input and select **Create»Constant**. Set the constant to **Finite Samples**.
  - – Right-click the samples per channel input and select **Create»Constant**. Set the constant to `100`.



- ❑ Simulate Signal Express VI—This Express VI displays a dialog box with which you can configure the parameters of the signal to simulate.
  - – In the dialog box that appears, set the options as shown in Figure 4-4.

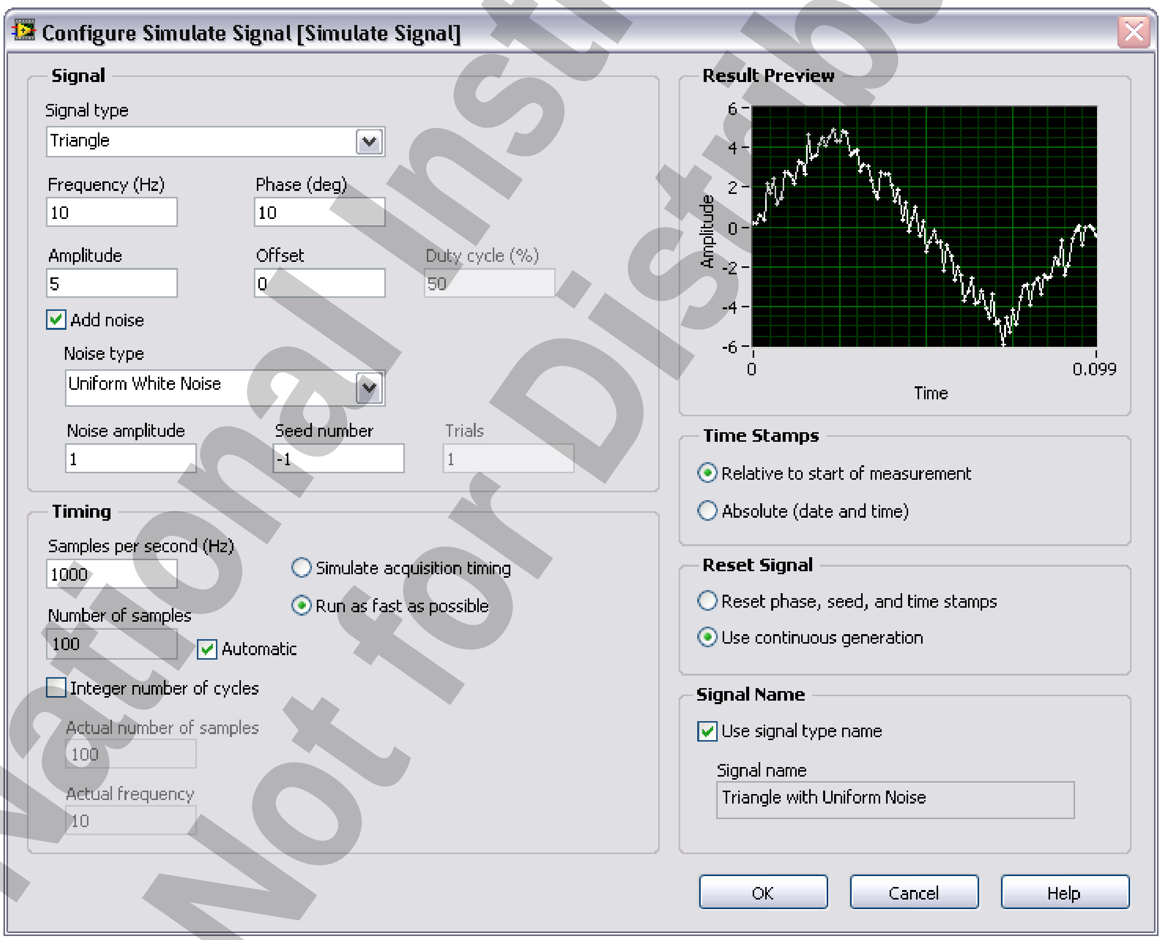


Figure 4-4. Simulate Signal Express VI Configuration Dialog

- – Click **OK**.

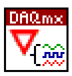
❑ DAQmx Write VI—This VI writes data generated by the Simulate Signal Express VI into the buffer for a finite, buffered analog output operation.

– Select the **Analog»Single Channel»Multiple Samples»Waveform** instance from the pull-down menu.

– Right-click the auto start input and select **Create»Constant** from the shortcut menu. Set the constant to **false**.

– Wire the signal from the Simulate Signal Express VI to the data input of the DAQmx Write VI.

❑ DAQmx Start Task VI



❑ DAQmx Wait Until Done VI—This VI waits until the analog output task completes before returning.

– Right-click the timeout (sec) input and select **Create»Constant**. Set the constant to `10`.

❑ DAQmx Clear Task VI

❑ Simple Error Handler VI

4. Configure the front panel, as shown in Figure 4-5.

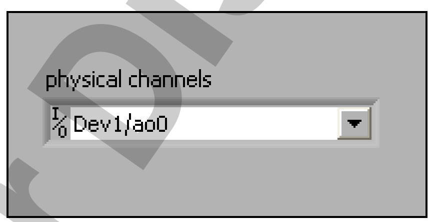


Figure 4-5. Finite Buffered Generation VI Front Panel

5. Save the VI.

Test

1. Monitor analog input channel 2 of the DAQ device.

   ❑ If the Cont Acq&Graph Voltage-Int Clk VI is not open, launch the NI Example Finder and navigate to the **Hardware Input and Output»DAQmx»Analog Measurements»Voltage** folder and open `Cont Acq&Graph Voltage-Int Clk.vi`.

   ❑ Set the front panel controls of the Cont Acq&Graph Voltage-Int Clk VI with the following values:

   - **Physical Channels**: `Dev1/ai2`
   - **Sample Rate (Hz)**: `1000`
   - **Samples to Read**: `250`

   ❑ Run the Cont Acq&Graph Voltage-Int Clk VI. The VI is now acquiring and displaying the voltage measured by analog input channel 2.

2. Test the Finite Buffered Generation VI.

   ❑ On the Finite Buffered Generation VI, set the Physical Channels control to `Dev1/ao0`.

   ❑ While viewing the graph on the Cont Acq&Graph Voltage-Int Clk VI, run the Finite Buffered Generation VI.

   You should see a single triangle wave with noise displayed on the waveform chart of the Cont Acq&Graph Voltage-Int Clk VI each time you run the Finite Buffered Generation VI.

3. Stop the VIs, but leave them open for Exercise 4-3.

End of Exercise 4-2

Exercise 4-3 Triggered Continuous Buffered Generation

Objective

To build a VI to trigger a continuous buffered generation on an analog output channel.

Scenario

In this exercise, you modify the Finite Buffered Generation VI from Exercise 4-2 to trigger a continuous waveform generation using the quadrature encoder on the BNC-2120.

BNC-2120 Configuration

1. In the Timing I/O section, wire the **PULSES** or **CLK** terminal to the **PFI 1** screw terminal.

Note Some models of the BNC-2120 have a **PULSES** terminal and some models have a **CLK** terminal instead. They are functionally the same.

2. Connect the BNC connector of channel 0 in the Analog Outputs section to the BNC connector of channel 2 in the Analog Inputs section.

3. Verify that the **FS/GS** switch underneath the BNC connector of channel 2 is in the `GS` position.

Implementation

1. Update the Finite Buffered Generation VI to trigger waveform generation using a quadrature encoder.

   ❑ Open the Finite Buffered Generation VI you created in Exercise 4-2.

   ❑ Select **File»Save As»Copy»Substitute Copy for Original** and save the VI as `Triggered Continuous Buffered Generation.vi` in the `<Exercises>\DAQ and Signal Conditioning\Analog Output` directory.

- ❑ Modify the block diagram, as shown in Figure 4-6, using the following items:

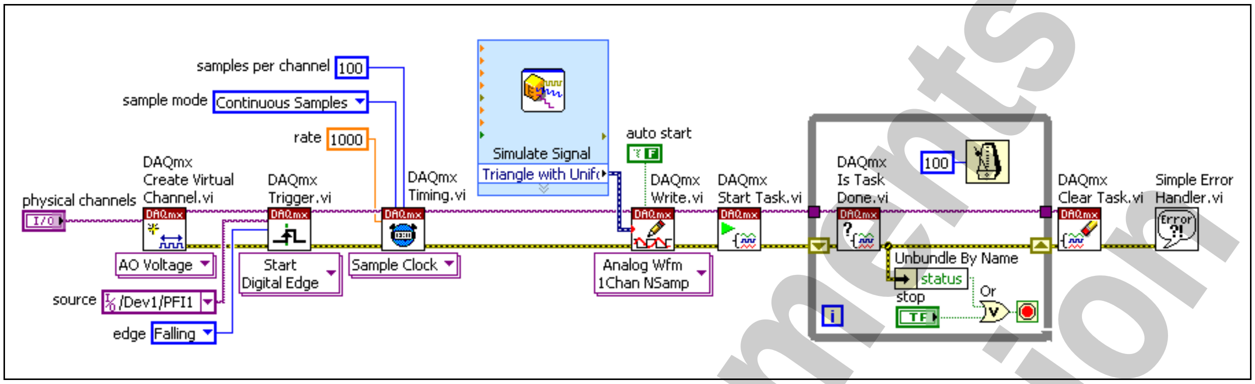
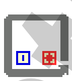
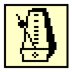


Figure 4-6. Triggered Continuous Buffered Generation VI Block Diagram

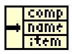
- ❑ Place the DAQmx Trigger VI on the block diagram between the DAQmx Create Virtual Channel VI and the DAQmx Timing VI. The DAQmx Trigger VI configures the trigger conditions for the analog output task.
  - – Select the **Start»Digital Edge** instance from the pull-down menu.
  - – Right-click the source input and select **Create»Constant**. Set the constant to `/Dev1/PFI1`.
  - – Right-click the edge input and select **Create»Constant**. Set the constant to **Falling**.
- ❑ On the DAQmx Timing VI, change the sample mode constant to **Continuous Samples**.
- ❑ Delete the DAQmx Wait Until Done VI.



- ❑ Add a While Loop.
  - – Wire the error out terminal of the DAQmx Start Task VI to the left border of the While Loop.
  - – Right-click the tunnel and select **Replace with Shift Register**.
  - – Create the stop control from the Loop Condition terminal of the While Loop.

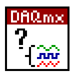


- ❑ Wait Until Next ms Multiple function
  - – Right-click the millisecond multiple input and select **Create»Constant**. Set the constant to `100`.



- ❑ Unbundle by Name function

- ❑ Or function



- ❑ Place the DAQmx Is Task Done VI on the block diagram. This VI ensures that the specified operation is complete before you stop the task.
- ❑ Save the VI.

Test

1. Monitor analog input channel 2 of the DAQ device.
   - ❑ If the Cont Acq&Graph Voltage-Int Clk VI is not open, launch the NI Example Finder and navigate to the **Hardware Input and Output»DAQmx»Analog Measurements»Voltage** folder and open `Cont Acq&Graph Voltage-Int Clk.vi`.
   - ❑ Set the front panel controls of the Cont Acq&Graph Voltage-Int Clk VI with the following values:
     - **Physical Channels**: `Dev1/ai2`
     - **Sample Rate (Hz)**: `1000`
     - **Samples to Read**: `250`
   - ❑ Run the Cont Acq&Graph Voltage-Int Clk VI. The VI is now acquiring and displaying the voltage measured by analog input channel 2.
2. Test the Triggered Continuous Buffered Generation VI.
   - ❑ On the Triggered Continuous Buffered Generation VI, set the Physical Channels control to `Dev1/ao0`.
   - ❑ Run the Triggered Continuous Buffered Generation VI.
   - ❑ Return to the front panel of the Cont Acq&Graph Voltage-Int Clk VI.

❑ Turn the Quadrature Encoder knob on the BNC-2120. Remember that when you turn the Quadrature Encoder knob, it produces a falling edge on the PULSES or CLK terminal which is connected to the PFI1 line. The falling edge received on the PFI1 line triggers the waveform generation on the Triggered Continuous Buffered Generation VI.

You should see the triangle wave with noise displayed continuously on the waveform chart of the Cont Acq&Graph Voltage-Int Clk VI, as shown in Figure 4-7.

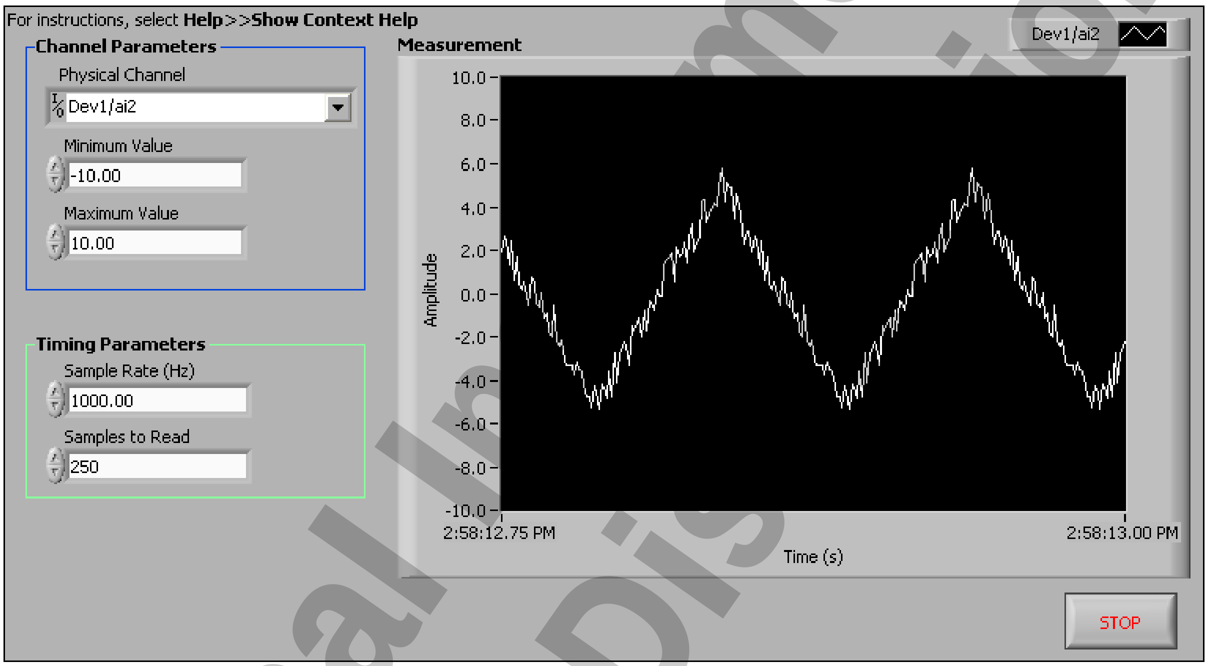


Figure 4-7. Cont Acq&Graph Voltage-Int Clk VI Front Panel

3. Stop and close the VIs when finished.

End of Exercise 4-3

Notes

Notes

5

Digital I/O Exercises

Exercise 5-1 Digital Read

Objective

To acquire digital data using the DAQ device, and display the digital data on the front panel of a LabVIEW VI.

BNC-2120 Configuration

1. Use a wire to connect the **UP/DN** screw terminal in the Timing I/O section to the screw terminal for line **7** in the Digital I/O section.

Implementation

1. Open the `Digital Read.vi` in the `<Exercises>\DAQ and Signal Conditioning\Digital` directory.

2. Modify the block diagram to acquire and display digital data, as shown in Figure 5-1, using the following items:

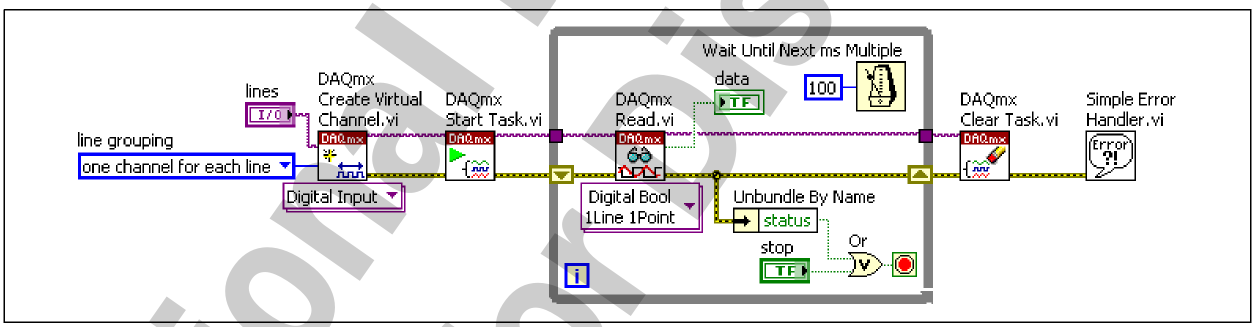
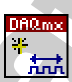


Figure 5-1. Digital Read VI Block Diagram



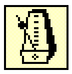
❑ DAQmx Create Virtual Channel VI

– Select the **Digital Input** instance from the pull-down menu.

– Right-click the lines input and select **Create»Control**.

– Right-click the line grouping input and select **Create»Constant**. Set the constant to **one channel for each line**.

- ❑ DAQmx Read VI
  - – Select the **Digital»Single Channel»Single Sample» Boolean (1 line)** instance from the pull-down menu.
  - – Right-click the data output and select **Create»Indicator**.



- ❑ Wait Until Next ms Multiple function
  - – Right-click the millisecond multiple input and select **Create» Constant**. Enter `100` as the value for the constant.

3. Arrange the items on the front panel, as shown in Figure 5-2.

Figure 5-2. Digital Read VI Front Panel

4. Save the VI.

Test

1. Set the front panel control of the Digital Read VI with the following value:
   - **Lines**: `Dev1/port0/line7`
2. Test the VI.
   - ❑ Run the VI.
   - ❑ Rotate the Quadrature Encoder knob clockwise. The data LED on the front panel should be lit.
   - ❑ Rotate the Quadrature Encoder knob counterclockwise. The data LED on the front panel should turn off.

Note The **UP/DN** terminal on the BNC-2120 outputs a high or a low signal indicating the rotation direction of the Quadrature Encoder knob. When you rotate the Quadrature Encoder knob clockwise, the **UP/DN** terminal outputs a high signal. When you rotate the Quadrature Encoder knob counterclockwise, the **UP/DN** terminal outputs a low signal.

3. Stop and close the VI when finished.

End of Exercise 5-1

Exercise 5-2 Digital Write

Objective

To output digital data from the DAQ device and display updates on the LEDs of the BNC-2120.

BNC-2120 Configuration

1. Verify that all the screw terminals in the Digital I/O section of the BNC-2120 are not attached to any wires.

Note The Digital I/O screw terminals are internally connected to port 0, lines 0 through 7 on the DAQ device. When a digital I/O line is either pulled or driven high, the corresponding digital I/O LED is lit. When a digital I/O line is either pulled or driven low, the corresponding digital I/O LED is not lit.

Implementation

1. Open the Digital Write.vi in the <Exercises>\DAQ and Signal Conditioning\Digital directory.

2. Modify the block diagram to generate digital data, as shown in Figure 5-3, using the following items:

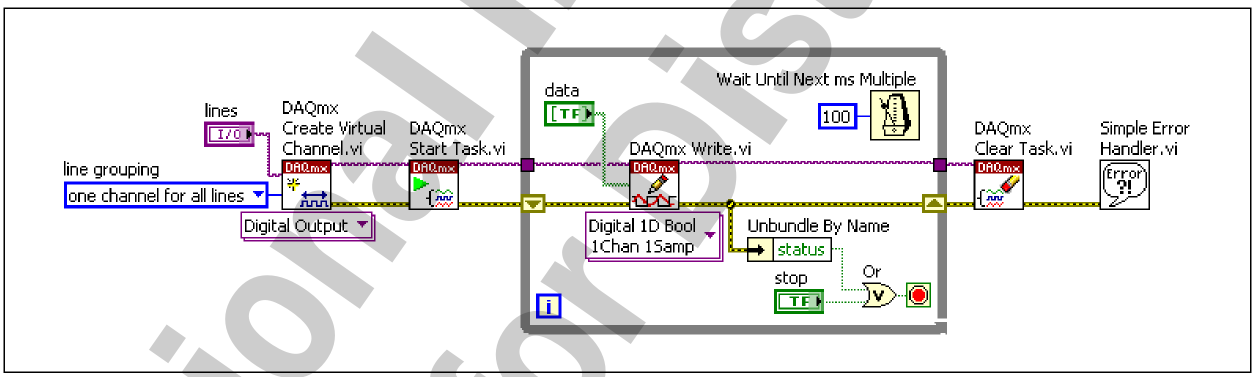


Figure 5-3. Digital Write VI Block Diagram

❑ DAQmx Create Virtual Channel VI

– Select the **Digital Output** instance from the pull-down menu.

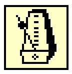
– Right-click the lines input and select **Create»Control**.

– Right-click the line grouping input and select **Create»Constant**. Set the constant to **one channel for all lines**.

❑ DAQmx Write VI

– Select the **Digital»Single Channel»Single Sample» 1D Boolean (N lines)** instance from the pull-down menu.

– Right-click the data input and select **Create»Control**.



❑ Wait Until Next ms Multiple function

– Right-click the millisecond multiple input and select **Create» Constant**. Enter `100` as the value for the constant.

3. Arrange the items on the front panel, as shown in Figure 5-2. You will set the values of the controls in the Test section of this exercise.

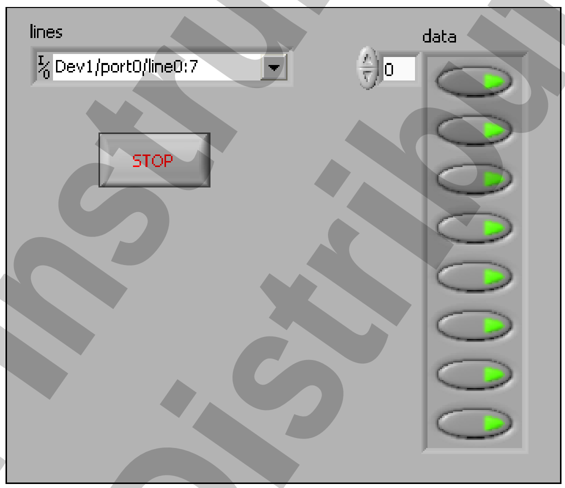


Figure 5-4. Digital Write VI Front Panel

❑ Resize the data Boolean array control to show eight elements by clicking and dragging the border of the data Boolean array control.

4. Save the VI.

Test

1. Set the front panel controls of the Digital Write VI as follows:

❑ Click the pull-down menu of the lines control and select **Browse**.

- ❑ Hold the <Shift> key and select lines 0 through 7, as shown in Figure 5-5.

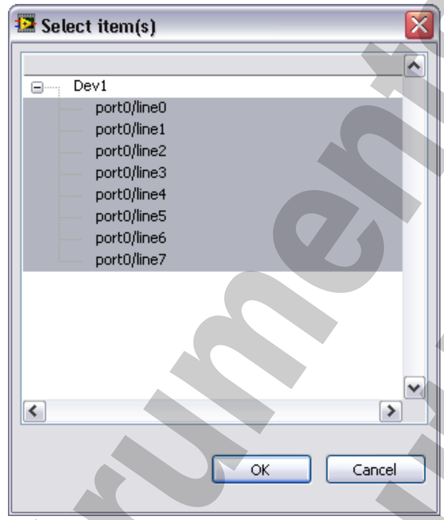


Figure 5-5. DAQmx Digital Channel Browse Dialog

- ❑ Click **OK**.

- ❑ On the front panel, click each Boolean button so that they are all lit.

Note The number of digital lines selected must match the number of Boolean elements sent by the Boolean array. Therefore, you must initialize eight elements in the data Boolean array to match the eight digital lines that you have selected.

2. Test the VI.

   - ❑ Run the VI. All of the digital I/O LEDs on the BNC-2120 should be lit.

   - ❑ Experiment by clicking the buttons in the Boolean array control, and observe the results on the corresponding LEDs.

     Is there anything peculiar about the order of the Boolean array control on the front panel versus the order of the actual LEDs on the BNC-2120? How could you change the control to better reflect the actual orientation of the LEDs?

3. Stop and close the VI when finished.

Challenge

Modify the VI so that the Boolean array control will better reflect the actual orientation of the LEDs on the BNC-2120.

End of Exercise 5-2

Exercise 5-3 Correlated Digital Output

Objective

To output digital data from the DAQ device using the Analog Output Sample Clock as the sample clock and view the digital data on the LEDs of the BNC-2120.

BNC-2120 Configuration

1. Verify that all the screw terminals in the Digital I/O section of the BNC-2120 are not attached to any wires.

Note The Digital I/O screw terminals are internally connected to port 0, lines 0 through 7 on the DAQ device. When a digital I/O line is either pulled or driven high, the corresponding digital I/O LED is lit. When a digital I/O line is either pulled or driven low, the corresponding digital I/O LED is not lit.

Implementation

1. Open the `Correlated Digital Output.vi` in the `<Exercises>\DAQ and Signal Conditioning\Digital` directory.

2. Modify the block diagram to generate digital data, as shown in Figure 5-6, using the following items:

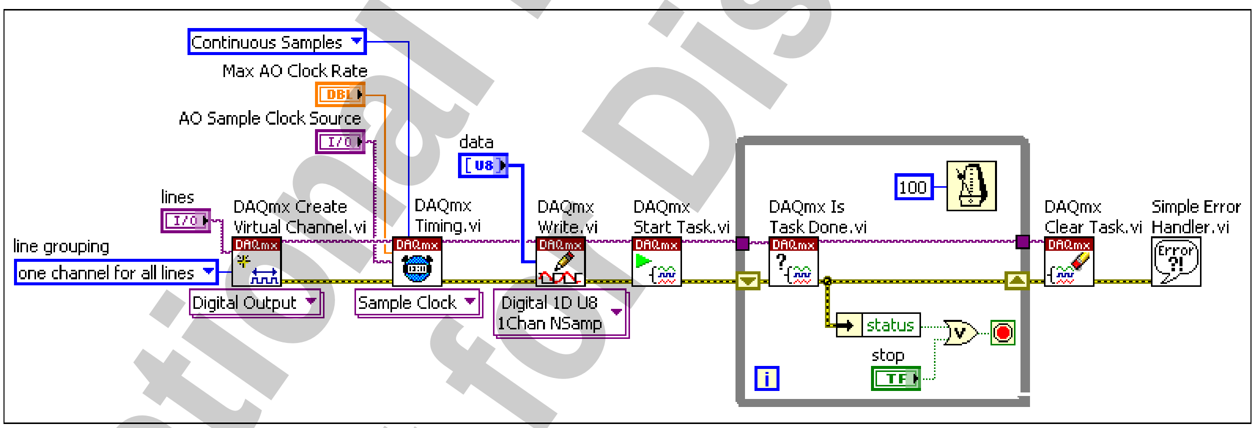


Figure 5-6. Correlated Digital Output VI Block Diagram

❑ DAQmx Create Virtual Channel VI

– Select the **Digital Output** instance from the pull-down menu.

– Right-click the lines input and select **Create»Control**.

– Right-click the line grouping input and select **Create»Constant**. Set the constant to **one channel for all lines**.

❑ DAQmx Timing VI

– Right-click sample mode input and select **Create»Constant**. Set the constant to **Continuous Samples**.

– Right-click the rate input and select **Create»Control**. Rename the control as `Max AO Clock Rate`.

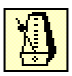
– Right-click the source input and select **Create»Control**. Rename the control as `AO Sample Clock Source`.

❑ DAQmx Write VI

– Select the **Digital»Single Channel»Multiple Samples»1D U8** instance from the pull-down menu.

– Right-click the data input and select **Create»Control**.

❑ DAQmx Is Task Done VI—This VI queries the status of a task and indicates if it completed execution.



❑ Wait Until Next ms Multiple function

– Right-click the millisecond multiple input and select **Create» Constant** from the shortcut menu. Enter `100` as the value for the constant.

3. Configure and arrange the front panel similar to Figure 5-7.

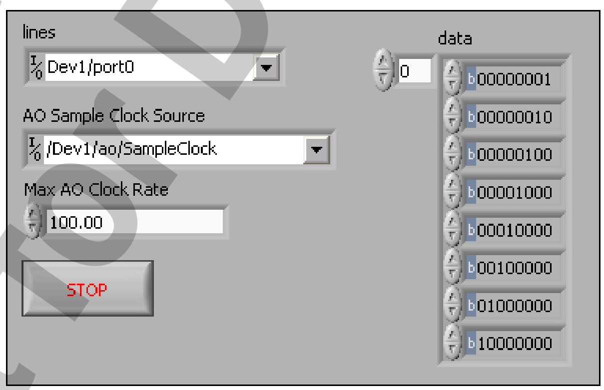


Figure 5-7. Correlated Digital Output VI Front Panel

❑ Right-click the lines control and select **I/O Name Filtering**. Set Port/Line Filtering to **Ports Only** and click **OK**.

❑ Resize the numeric array control to show eight elements.

- ❑ Display the radix by right-clicking an element in the numeric array control and selecting **Visible Items»Radix**.
- ❑ Display the numeric array in binary format by right-clicking an element in the numeric array and selecting **Display Format**. The dialog box in Figure 5-8 appears.

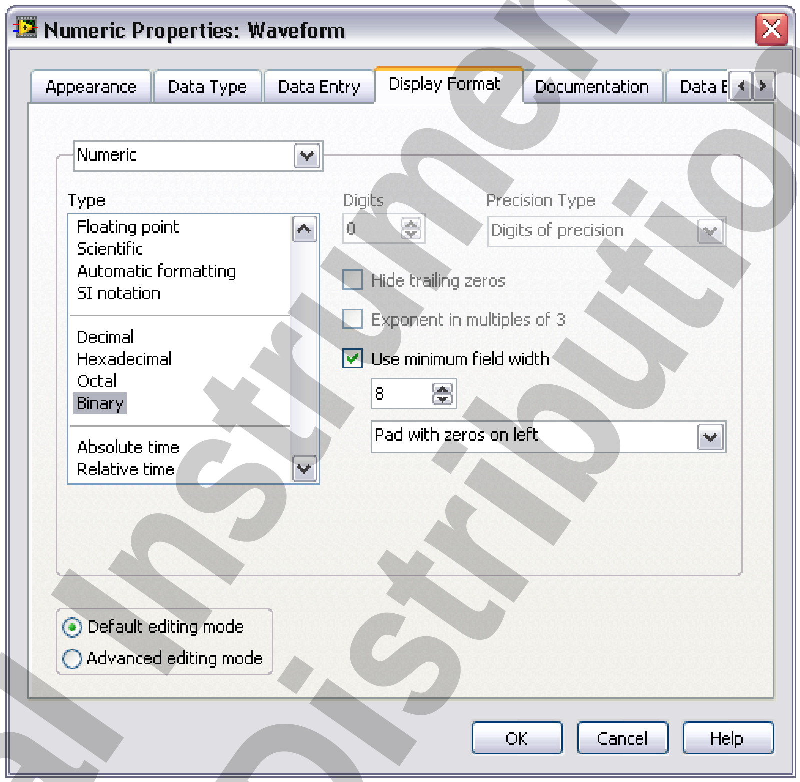


Figure 5-8. Numeric Display Format Dialog

   – Select **Default editing mode**. This option is located in the bottom left corner of the dialog box.
   – Select **Binary**.
   – Check **Use minimum field width**.
   – Set minimum field width to `8`.
   – Select **Pad with zeros on left**.
   – Click **OK**.

- ❑ Resize an element in the numeric array control to show all eight digits of the binary number, if necessary.

4. Save the VI.

Test

1. Set the values of the controls on the front panel.

   - **Lines**: `Dev1/port0`
   - **AO Sample Clock Source**: `/Dev1/ao/SampleClock`

Note M Series DAQ devices do not have a dedicated sample clock for digital I/O tasks; therefore, to create a clocked digital I/O signal, the task requires a clock from another source. In this case you will use the analog output circuitry's clock.

   - **Max AO Clock Rate**: `100`
   - Set the eight elements in the data numeric array to `00000001`, `00000010`, `00000100`, `00001000`, `00010000`, `00100000`, `01000000`, and `10000000`, as shown in Figure 5-7.

2. Test the VI.

   ❑ Run the VI.

   ❑ Observe the LEDs on the BNC-2120.

   Are the LEDs changing?

   The digital output task in this VI must receive a clock signal from the specified clock source before it can output the next digital sample to the LEDs.

3. Generate clock signals on the specified clock source to test the VI.

   ❑ Open `<Exercises>\DAQ and Signal Conditioning\Digital\Generate AO Sample Clock.vi`.

   ❑ Set the front panel controls of the Generate AO Sample Clock VI with the following values:

   - **Physical Channels**: `Dev1/ao0`
   - **Output Clock Rate (Hz)**: `2`

4. Run the VI.

   The Generate AO Sample Clock VI continuously generates voltage samples on the selected analog output channel. The Output Clock Rate (Hz) control determines the frequency of the analog output sample clock.

   ❑ Observe the LEDs changing.

   ❑ Change the **Output Clock Rate (Hz)** to affect the frequency of the digital task.

      Because the digital output is using the analog output sample clock as its clock source, the digital output is synchronized with the analog output generation.

5. Stop and close all VIs when finished.

End of Exercise 5-3

Notes

6

Counters Exercises

Exercise 6-1 Simple Edge Counting

Objective

To create a VI to count the number of edges produced by rotating the knob of the Quadrature Encoder.

Scenario

In this exercise, you will build a VI that records the number of times an edge occurs.

Default NI-DAQmx Counter Terminals

In this section, you will view the default NI-DAQmx counter terminals for your Dev1 DAQ device.

1. Open MAX.
2. Right-click the "Dev1" DAQ device under **My System»Devices and Interfaces** and select **Device Pinouts**.
3. View the Default NI-DAQmx Counter Terminals table in the NI-DAQmx Device Terminals Help window.
4. Notice that each counter/timer signal is associated with default PFI line. You will wire your counter signals in this lesson using the PFI lines exposed by the BNC-2120.
5. Leave the NI-DAQmx Device Terminals Help open for the rest of this lesson as a reference.

BNC-2120 Configuration

1. Use a wire to connect the **PULSES** or **CLK** screw terminal to the default Counter 0 Source (PFI 8) screw terminal in the Timing I/O section.

Note Some models of the BNC-2120 have a **PULSES** terminal and some models have a **CLK** terminal instead. They are functionally the same.

Implementation

1. Open the Simple Edge Counting VI in the `<Exercises>\DAQ and Signal Conditioning\Counter` directory.

2. Modify the block diagram of the Simple Edge Counting VI to record the number of times an edge occurs, as shown in Figure 6-1, using the following items:

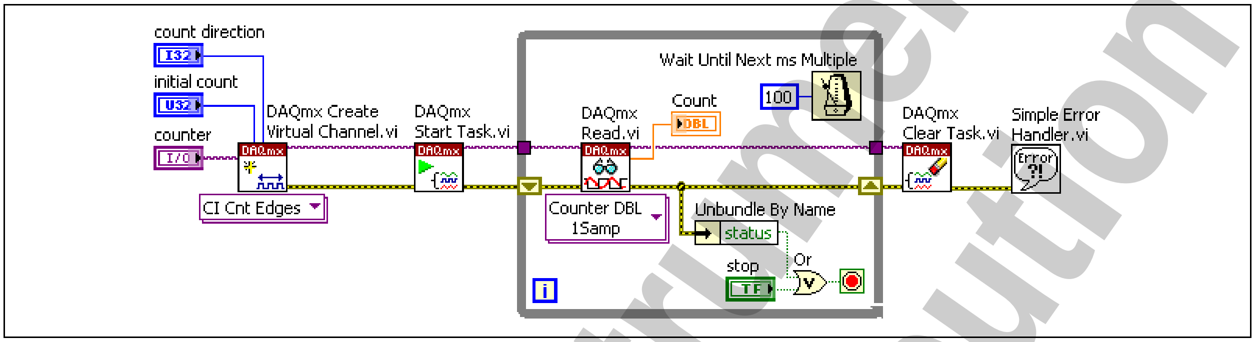


Figure 6-1. Simple Edge Counting VI Block Diagram

❑ DAQmx Create Virtual Channel VI

– Select the **Counter Input»Count Edges** instance from the pull-down menu.

– Right-click the counter, initial count, and count direction inputs and select **Create»Control**.

❑ DAQmx Read VI

– Select the **Counter»Single Sample»DBL** instance from the pull-down menu to read a single double-precision, floating-point value from the counter.

– Right-click the data output and select **Create»Indicator**. Rename the indicator as `Count`.

❑ Wait Until Next ms Multiple function

– Right-click the millisecond multiple input and select **Create»Constant**. Enter `100` as the value for the constant.

3. Arrange the front panel controls as shown in Figure 6-2.

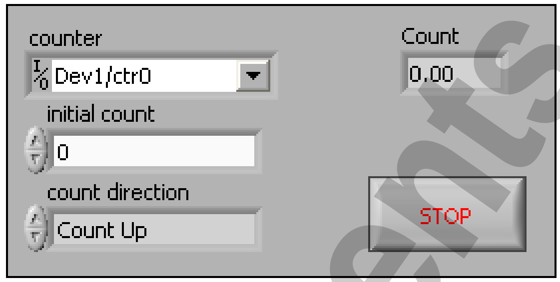


Figure 6-2. Simple Edge Counting VI Front Panel

4. Set the front panel controls with the following values:
   - **Counter**: `Dev1/ctr0`
   - **Initial Count**: `0`
   - **Count Direction**: `Count Up`
5. Save the VI.

Test

1. Test the VI.
   - ❑ Run the VI.
   - ❑ Rotate the Quadrature Encoder knob on the BNC-2120. The **PULSES** or **CLK** terminal outputs four pulses per one mechanical click of Quadrature Encoder to the source of counter 0.

     The Count indicator value should increase for every pulse read by the source of counter 0.
2. Stop the VI, but leave it open for Exercise 6-2.

End of Exercise 6-1

Exercise 6-2 Advanced Edge Counting

Objective

To use a pause trigger and finite buffered methods to perform edge counting.

Part A: Pause Trigger (Gated) Counting

In pause trigger, or gated, counting, you use an additional line to control the counter's gate. The gate pauses the increment or decrement of the counter when the gate is either in a high or low state (depending on the settings you choose).

BNC-2120 Configuration

1. Use a wire to connect the screw terminal for line 7 in the Digital I/O section to the **PFI 7** screw terminal in the Timing I/O section.

2. Use a wire to connect the **PULSES** or **CLK** screw terminal to the default Counter 0 Source (PFI 8) screw terminal in the Timing I/O section.

Note Some models of the BNC-2120 have a **PULSES** terminal and some models have a **CLK** terminal instead. They are functionally the same.

Implementation

1. If necessary, open the Simple Edge Counting VI located in the `<Exercises>\DAQ and Signal Conditioning\Counter` directory.

2. Select **File»Save As»Copy»Substitute copy for original** and save the VI as `Simple Edge Counting - Gated.vi` in the `<Exercises>\DAQ and Signal Conditioning\Counter` directory.

3. Modify the block diagram to add gating to edge counting, as shown in Figure 6-3, using the following item:

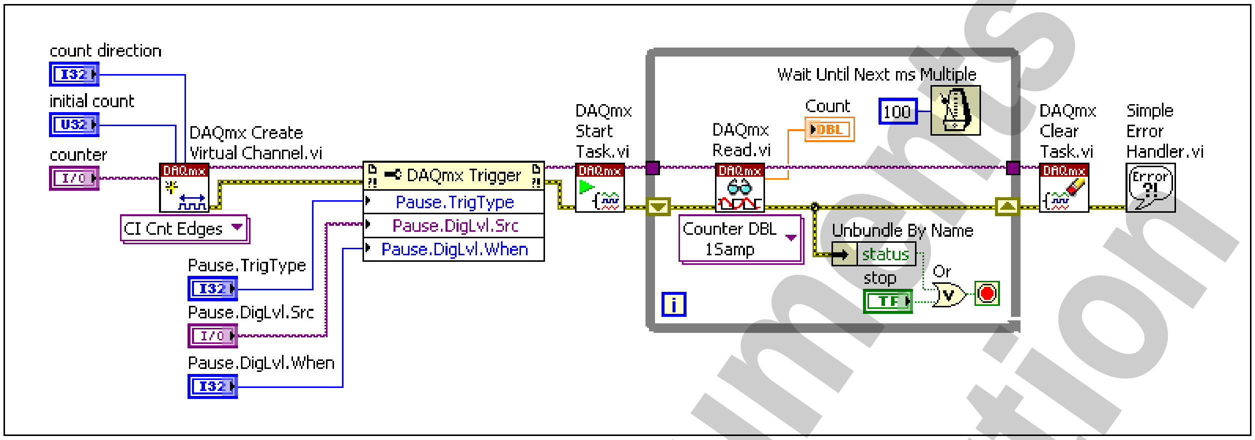
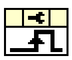


Figure 6-3. Simple Edge Counting - Gated VI Block Diagram



❑ DAQmx Trigger Property Node—Performs additional configuration of the task trigger.

– Resize the Property Node to display three elements.

– Right-click each element and select the following properties:

Select Property»More»Pause»Trigger Type
Select Property»More»Pause»Digital Level»Source
Select Property»More»Pause»Digital Level»Pause When

– Right-click any element and select **Change All to Write**.

– For each property, right-click the input and select **Create» Control**.

4. Arrange the front panel controls as shown in Figure 6-4.

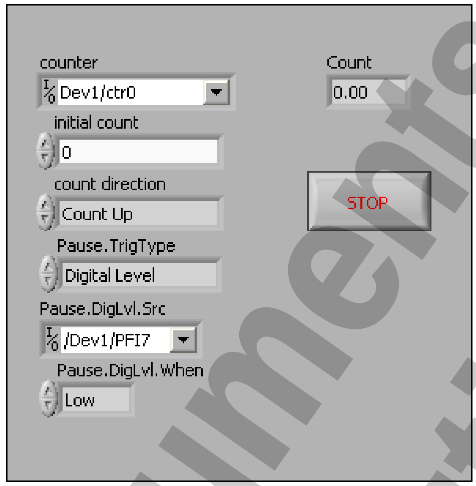


Figure 6-4. Simple Edge Counting - Gated VI Front Panel

5. Set the front panel controls with the following values:

   - **Counter**: `Dev1/ctr0`
   - **Initial Count**: `0`
   - **Count Direction**: `Count Up`
   - **Trigger Type**: `Digital Level`
   - **Source**: `/Dev1/PFI7`

     PFI 7 should be wired to the screw terminal for line 7. You will output a digital high or low signal from the screw terminal for line 7 in the Test section.

   - **Pause When**: `Low`

6. Save the VI.

Test

1. Send a digital low signal to the Pulse Trigger line.

   ❑ Open the Digital Write VI located in the `<Exercises>\DAQ and Signal Conditioning\Digital` directory.

   ❑ On the front panel, click the pull-down menu of the lines control and select **Browse**.

   ❑ Hold down the <Shift> key and select lines 0 through 7.

   ❑ Run the Digital Write VI.

   ❑ Use the front panel to output a low signal on the screw terminal for line 7 by making the corresponding button in the data Boolean array control unlit.

2. Test the Simple Edge Counting - Gated VI.

   ❑ Run the Simple Edge Counting - Gated VI.

   ❑ Rotate the Quadrature Encoder knob. Notice that the data numeric indicator does not change. The counting is paused when the pause trigger source is receiving a low signal.

   ❑ To unpause the counting, use the front panel of the Digital Write VI to output a high signal on the screw terminal for line 7 by making the corresponding button in the data Boolean array control lit.

   Rotate the Quadrature Encoder knob. Notice that the data numeric indicator begins to increment. Counting is resumed when the pause trigger source is receiving a high signal.

   ❑ Stop the VI and change the **Pause When** value to `High`.

   ❑ Run the VI and observe the behavior with this trigger setting.

3. Stop both VIs when finished. Close the Digital Write VI, but leave the Simple Edge Counting - Gated VI open.

Part B: Finite Buffered Counting

In finite buffered counting, you specify the total number of edges to count. Counting stops when this number is reached.

BNC-2120 Configuration

1. Use a wire to connect the **PULSES** or **CLK** screw terminal to the default Counter 0 Gate (PFI 9) screw terminal in the Timing I/O section.

Note Some models of the BNC-2120 have a **PULSES** terminal and some models have a **CLK** terminal instead. They are functionally the same.

Implementation

1. If necessary, open the Simple Edge Counting VI located in the `<Exercises>\DAQ and Signal Conditioning\Counter` directory.

2. Select **File»Save As»Copy»Substitute copy for original** and save the VI as `Simple Edge Counting - Finite Buffered.vi` in the `<Exercises>\DAQ and Signal Conditioning\Counter` directory.

3. Modify the block diagram to include finite buffering, as shown in Figure 6-5, using the following items:

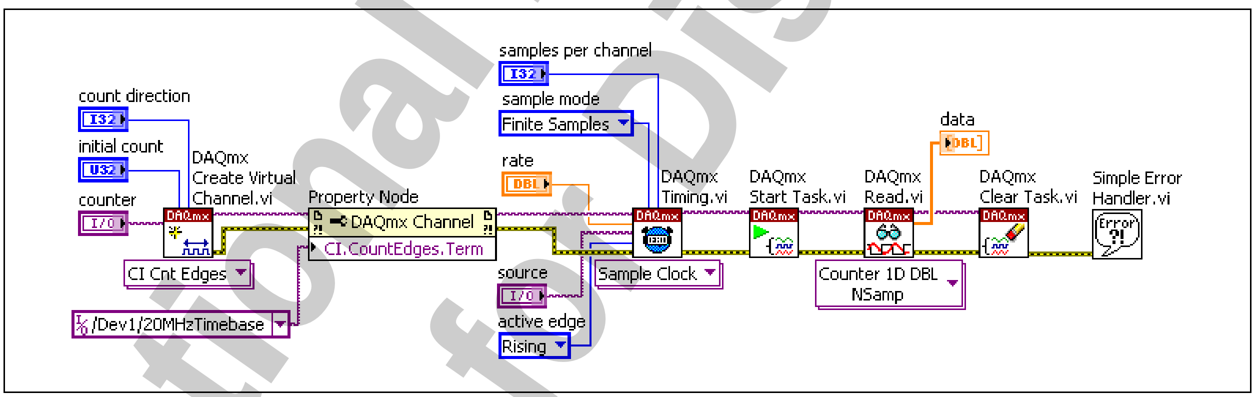
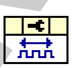


Figure 6-5. Simple Edge Counting - Finite Buffered VI Block Diagram

❑ Right-click the While Loop and select **Remove While Loop**.



❑ DAQmx Channel Property Node

– Right-click the terminal and select **Select Property»Counter Input»Count Edges»Input Terminal**.

– Right-click the terminal and select **Change All to Write**.

– Right-click the terminal and select **Create»Constant**. Set the constant to `/Dev1/20MHzTimebase`.

Instead of counting the clicks on the Quadrature Encoder, this VI will count the edges of an internal timebase of the DAQ device.

❑ DAQmx Timing VI

– Select the **Sample Clock** instance from the pull-down menu.

– Right-click the active edge and sample mode inputs and select **Create»Constant**. Set the sample mode constant to **Finite Samples**.

– Right-click the source, rate, and samples per channel inputs and select **Create»Control**.

❑ Configure the DAQmx Read VI to read one or more samples from the counter.

– Select the **Counter»Multiple Samples»1D DBL** instance from the pull-down menu to read one or more single double-precision, floating-point values from the counter.

– Delete the count numeric indicator.

– Right-click the data output and select **Create»Indicator** to display a 1D array of samples.

4. Arrange the front panel controls as shown in Figure 6-6.

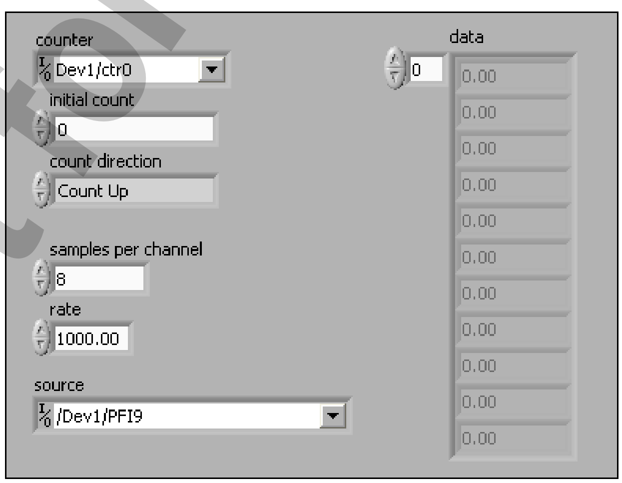


Figure 6-6. Simple Edge Counting - Finite Buffered VI Front Panel

5. Set the front panel controls of the Simple Edge Counting - Finite Buffered VI with the following values:

   - **Counter**: `Dev1/ctr0`
   - **Initial Count**: `0`
   - **Count Direction**: `Count Up`
   - **Samples per Channel**: `8`
   - **Rate**: `1000`
   - **Source**: `/Dev1/PFI9`. PFI 9 is the default pin for the Counter 0 Gate.

Test

1. Run the VI.
2. Rotate the Quadrature Encoder knob twice to see eight latched values of the count register in the data numeric array. For every rising edge of the gate signal, one value of the count register is acquired. If more than eight fields are visible in the array, notice that the additional fields remain at 0.00.

Note The **PULSES** or **CLK** terminal outputs four pulses per one mechanical click of Quadrature Encoder. Rotating the Quadrature Encoder knob two mechanical clicks will output eight pulses.

3. Save and close the VI when finished.

End of Exercise 6-2

Exercise 6-3 Pulse Generation

Objective

To build a VI that generates a single pulse using a counter.

Scenario

This VI demonstrates how to output a value to a counter. Although you output a frequency in this exercise, you also can output counter ticks and time with the same concept presented in this exercise.

BNC-2120 Configuration

1. Use a wire to connect the default Counter 0 Out (PFI 12) screw terminal in the Timing I/O section to the **USER 1** screw terminal in the user-defined signals section.

2. Connect the **USER 1** BNC connector to the BNC connector for channel 2 in the Analog Inputs section.

3. Verify that the **FS/GS** switch underneath the BNC connector for channel 2 is in the `GS` position.

Implementation

1. Open the Single Pulse Generation VI in the `<Exercises>\DAQ and Signal Conditioning\Counter` directory.

2. Modify the block diagram of the Single Pulse Generation VI to generate a single pulse, as shown in Figure 6-7, using the following items:

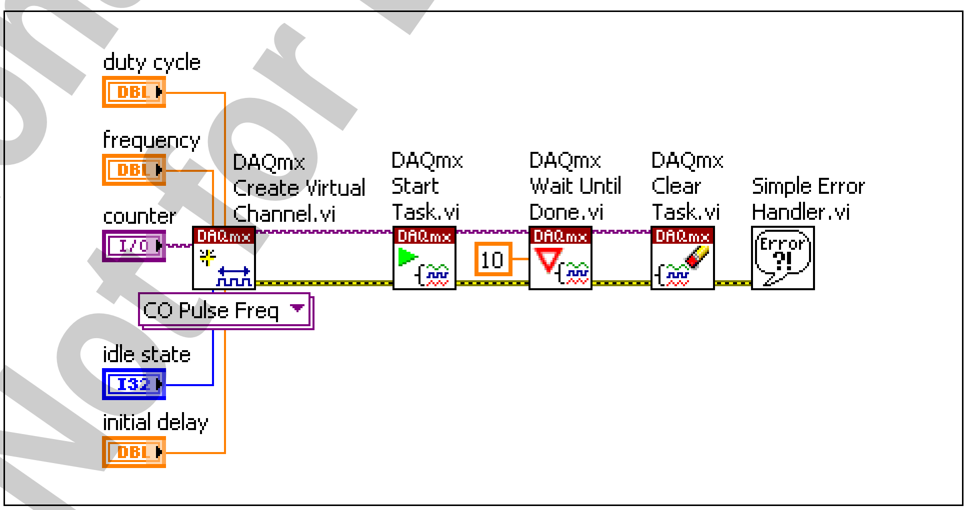
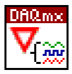


Figure 6-7. Single Pulse Generation VI Block Diagram

- ❑ DAQmx Create Virtual Channel VI
  - – Select the **Counter Output»Pulse Generation»Frequency** instance from the pull-down menu.
  - – Right-click the counter, frequency, duty cycle, idle state, and initial delay inputs and select **Create»Control**.



- ❑ DAQmx Wait Until Done VI
  - – Right-click the timeout input and select **Create»Constant**. Set the constant to `10`.

3. Arrange the front panel controls as shown in Figure 6-8.

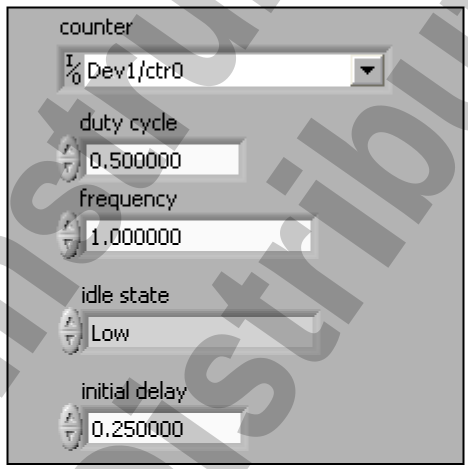


Figure 6-8. Single Pulse Generation VI Front Panel

4. Save the VI.

Test

1. Configure the VI to monitor the analog input channel 2 of the DAQ device.

   ❑ Open the NI Example Finder by selecting **Help»Find Examples**.

   ❑ Navigate to the **Hardware Input and Output»DAQmx»Analog Measurements»Voltage** folder and open `Cont Acq&Graph Voltage-Int Clk.vi`.

   ❑ Set the front panel controls of the Cont Acq&Graph Voltage-Int Clk VI with the following values:

   - **Physical Channels**: `Dev1/ai2`
   - **Sample Rate (Hz)**: `10000`
   - **Samples to Read**: `1000`

   ❑ Run the Cont Acq&Graph Voltage-Int Clk VI. The VI is now acquiring and displaying the voltage measured by analog input channel 2.

2. Test the Single Pulse Generation VI.

   ❑ On the Single Pulse Generation VI, set the front panel controls to the following values:

   - **Counter**: `Dev1/ctr0`
   - **Duty Cycle**: `0.5`
   - **Frequency**: `1`
   - **Idle State**: `Low`
   - **Initial Delay**: `0.25`

   ❑ While viewing the graph on the Cont Acq&Graph Voltage-Int Clk VI, run the Single Pulse Generation VI.

   You should see the pulse appear on the Cont Acq&Graph Voltage-Int Clk VI each time you run the Single Pulse Generation VI. Notice that the signal starts low, goes high, and returns to low.

Note It is easier to see the pulse if you verify that the Autoscale Y option is disabled by right-clicking the graph while the VI is running and deselecting **AutoScale Y** from the shortcut menu to remove the checkmark from the item. An ideal y-scale for viewing this pulse is `–1` to `6`.

3. Stop the Cont Acq&Graph Voltage-Int Clk VI when finished. Keep the Cont Acq&Graph Voltage-Int Clk VI open because you will use it again for testing in the next exercise.

4. Save and close the Single Pulse Generation VI.

End of Exercise 6-3

Exercise 6-4 Pulse Train Generation

Objective

To build a VI that generates a pulse train.

Scenario

In this exercise, you will create a VI that generates a pulse train or a series of pulses.

BNC-2120 Configuration

1. Use a wire to connect the default Counter 0 Out (**PFI 12**) screw terminal in the Timing I/O section to the **USER 1** screw terminal in the user-defined signals section.
2. Connect the **USER 1** BNC connector to the BNC connector for channel 2 in the Analog Inputs section.
3. Verify that the **FS/GS** switch underneath the BNC connector for channel 2 is in the `GS` position.

Implementation

1. Open the Pulse Train Generator VI located in the `<Exercises>\DAQ and Signal Conditioning\Counter` directory.
2. Modify the block diagram to generate a series of pulses, as shown in Figure 6-9, using the following items:

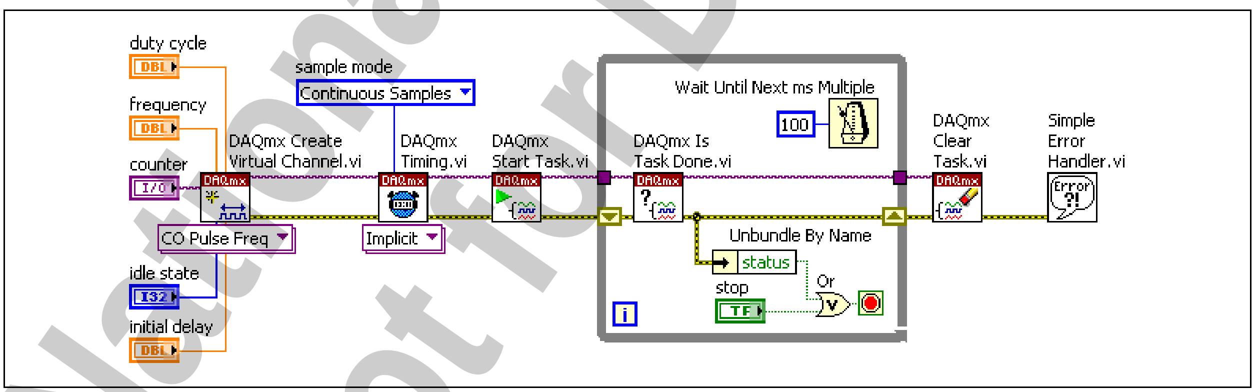
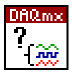


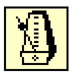
Figure 6-9. Pulse Train Generator VI Block Diagram

- ❑ DAQmx Timing VI
  - – Select the **Implicit** instance from the pull-down menu to configure timing so the task generates samples without specifying timing.

    A pulse train generation is ideal for the Implicit timing, because the pulse train itself contains all the timing parameters.
  - – Right-click the sample mode input and select **Create»Constant**. Set the constant to **Continuous Samples**.



- ❑ DAQmx Is Task Done VI



- ❑ Wait Until Next ms Multiple function
  - – Right-click the **millisecond multiple** input and select **Create» Constant**. Enter `100` as the value for the constant.

3. Arrange the front panel controls as shown in Figure 6-10.

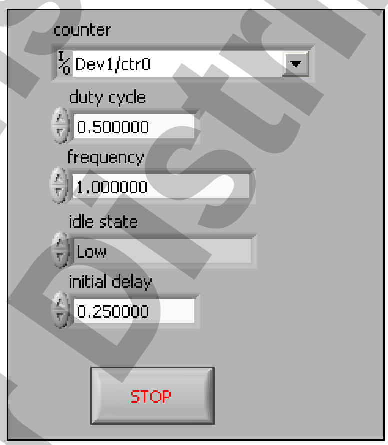


Figure 6-10. Pulse Train Generator VI Front Panel

4. Save the VI.

Test

1. Monitor the analog input channel 2 of the DAQ device.

   - ❑ If necessary, open the NI Example Finder by selecting **Help»Find Examples**. Navigate to the **Hardware Input and Output» DAQmx»Analog Measurements»Voltage** folder and open `Cont Acq&Graph Voltage-Int Clk.vi`.

❑ Set the front panel controls of the Cont Acq&Graph Voltage-Int Clk VI with the following values:

- **Physical Channels**: `Dev1/ai2`
- **Sample Rate (Hz)**: `10000`
- **Samples to Read**: `1000`

❑ Run the Cont Acq&Graph Voltage-Int Clk VI. The VI is now acquiring and displaying the voltage measured by analog input channel 2.

2. Test the Pulse Train Generator VI.

❑ On the Pulse Train Generator VI, set the front panel controls to the following values as shown in Figure 6-10.

- **Counter**: `Dev1/ctr0`
- **Duty Cycle**: `0.5`
- **Frequency**: `1`
- **Idle State**: `Low`
- **Initial Delay**: `0.25`

❑ While viewing the graph on the Cont Acq&Graph Voltage-Int Clk VI, run the Pulse Train Generator VI.

You should see a pulse train appear on the Cont Acq&Graph Voltage-Int Clk VI.

Note It is easier to see the pulse train if you verify that the Autoscale Y option is disabled by right-clicking the graph while the VI is running and deselecting **AutoScale Y** from the shortcut menu to remove the checkmark from the item. A recommended y-scale for observing the pulse train is `-1` to `6`.

3. Stop the Pulse Train Generator VI when finished. Keep the Pulse Train Generator VI open because you will modify it in Exercise 6-5.

4. Stop the Cont Acq&Graph Voltage-Int Clk VI when finished. Keep the Cont Acq&Graph Voltage-Int Clk VI open because you will use it again for testing in Exercise 6-5.

End of Exercise 6-4

Exercise 6-5 Retriggerable Pulse Train Generation

Objective

To build a VI that generates a retriggerable finite pulse train.

Scenario

You can configure single pulse generation and finite pulse train generation to be retriggerable. The counter remains armed after generating the first pulse and can respond to triggers on the gate by generating a pulse on the counter's output line. In this exercise, you modify the Pulse Train Generator VI to generate a retriggerable pulse train.

BNC-2120 Configuration

1. Use a wire to connect the **PULSES** or **CLK** screw terminal to the **PFI 1** screw terminal in the Timing I/O section.

2. Use a wire to connect the default Counter 0 Out (**PFI 12**) screw terminal in the Timing I/O section to the **USER 1** screw terminal in the user-defined signals section.

3. Connect the **USER 1** BNC connector to the BNC connector for channel 2 in the Analog Inputs section.

4. Verify that the **FS/GS** switch underneath the BNC connector for channel 2 is in the GS position.

Implementation

1. If necessary, open the Pulse Train Generator VI located in the <Exercises>\DAQ and Signal Conditioning\Counter directory.

2. Select **File»Save As»Copy»Substitute copy for original** and save the VI as Retriggerable Pulse Train.vi in the <Exercises>\DAQ and Signal Conditioning\Counter directory.

3. Modify the block diagram to generate a retriggerable pulse train, as shown in Figure 6-11, using the following items:

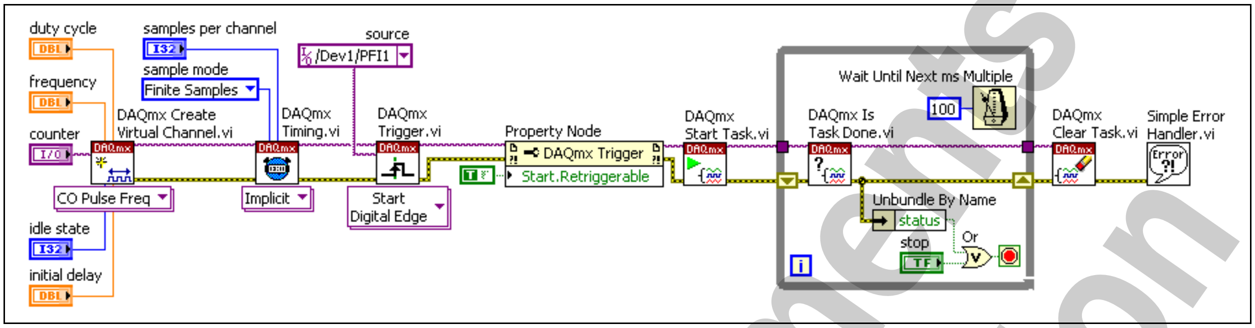


Figure 6-11. Retriggerable Pulse Train VI Block Diagram

❑ Modify the DAQmx Timing VI.

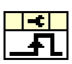
– Set the sample mode to **Finite Samples**.

– Right-click the samples per channel input and select **Create» Control**.

❑ DAQmx Trigger VI

– Select the **Start»Digital Edge** instance from the pull-down menu.

– Right-click the source input and select **Create»Constant**. Set the constant to `/Dev1/PFI1`.



❑ DAQmx Trigger Property Node-Use this Property Node to configure additional settings for the task triggering.

– Right-click the element of the Property Node and select **Select Property»Start»More»Retriggerable**.

– Right-click the Start.Retriggerable element input and select **Create»Constant** from the shortcut menu. Set the Boolean constant to **True** to allow the pulse train to be retriggerable.

4. Arrange the front panel controls as shown in Figure 6-12.

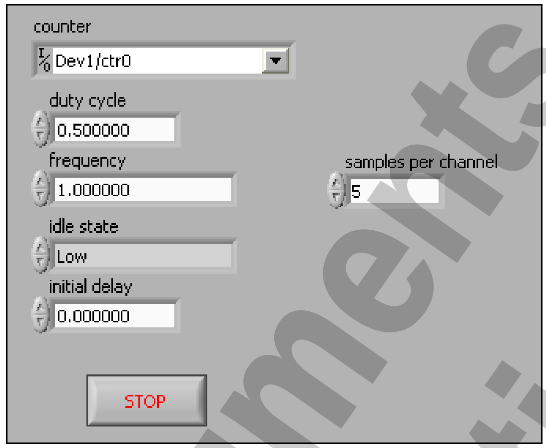


Figure 6-12. Retriggerable Pulse Train VI Front Panel

5. Save the VI.

Test

1. Monitor the analog input channel 2 of the DAQ device.

   ❑ If necessary, open the NI Example Finder by selecting **Help»Find Examples**. Navigate to the **Hardware Input and Output» DAQmx»Analog Measurements»Voltage** folder and open `Cont Acq&Graph Voltage-Int Clk.vi`.

   ❑ Set the front panel controls of the Cont Acq&Graph Voltage-Int Clk VI with the following values:

   - **Physical Channels**: `Dev1/ai2`
   - **Sample Rate (Hz)**: `10000`
   - **Samples to Read**: `1000`

   ❑ Run the Cont Acq&Graph Voltage-Int Clk VI. The VI is now acquiring and displaying the voltage measured by analog input channel 2.

2. Test the Retriggerable Pulse Train VI.

   ❑ On the Retriggerable Pulse Train VI, set the controls to the following values:

   - **Counter**: `Dev1/ctr0`
   - **Duty Cycle**: `0.5`
   - **Frequency**: `1`
   - **Idle State**: `Low`
   - **Initial Delay**: `0`
   - **Samples per channel**: `5`

   ❑ While viewing the graph on the Cont Acq&Graph Voltage-Int Clk VI, run the Retriggerable Pulse Train Generator VI.

   You should not see any pulses on the Cont Acq&Graph Voltage-Int Clk VI because you must first trigger the pulse train.

   ❑ Turn the Quadrature Encoder knob on the BNC-2120 to trigger the pulse train generation.

   Watch the graph on the Cont Acq&Graph Voltage-Int Clk VI, and count the number of pulses. The number of pulses should be equivalent to the value of the samples per channel control.

   ❑ Turn the Quadrature Encoder knob again. The pulse train is generated each time the trigger is enabled, making it a retriggerable pulse train.

Note It is easier to see the pulse train if you verify that the Autoscale Y option is disabled by right-clicking the graph while the VI is running and deselecting **AutoScale Y** from the shortcut menu to remove the checkmark from the item. A recommended y-scale for observing the pulse train is `–1` to `6`.

3. Stop and close both VIs when finished.

End of Exercise 6-5

Exercise 6-6 Pulse Width and Period Measurement

Objective

To build a VI to measure the pulse width and period of a pulse train.

BNC-2120 Configuration

1. Use a wire to connect the Counter 1 Out (**PFI 13**) screw terminal to the Counter 0 Gate (**PFI 9**) screw terminal in the Timing I/O section.

Implementation

Part A: Pulse Width

Pulse width measurements measure the time between either a rising and falling edge or a falling and rising edge.

1. Open the Pulse Width Measurement VI in the `<Exercises>\DAQ and Signal Conditioning\Counter` directory.

2. Modify the block diagram of the Pulse Width Measurement VI to measure pulse width, as shown in Figure 6-13, using the following items:

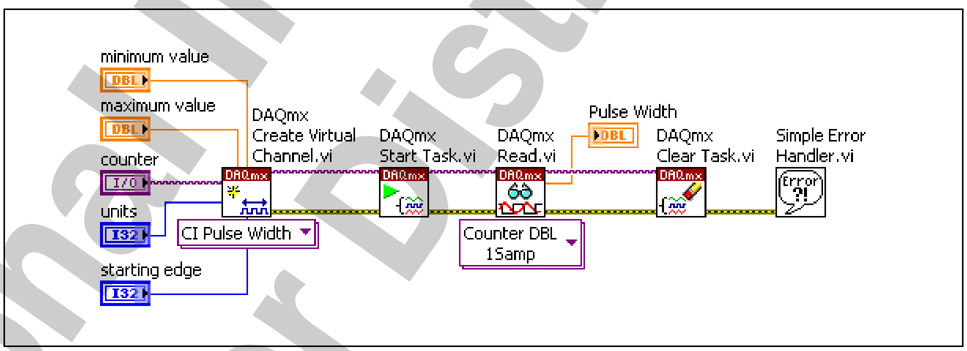


Figure 6-13. Pulse Width Measurement VI Block Diagram

❑ DAQmx Create Virtual Channel VI

– Select the **Counter Input»Pulse Width** instance from the pull-down menu.

– Right-click the minimum value, maximum value, counter, units, and starting edge inputs and select **Create»Control**.

❑ DAQmx Read VI

– Select the **Counter»Single Sample»DBL** instance from the pull-down menu to read a single, double-precision, floating-point value from the counter.

– Right-click the data output and select **Create»Indicator** from the shortcut menu. Rename the indicator as `Pulse Width`.

3. Save the VI.

Test

1. Generate a 10 Hz pulse train that outputs a high signal 20% of the time and a low signal 80% of the time.

❑ Open the Pulse Train Generator VI in the `<Exercises>\DAQ and Signal Conditioning\Counter` directory. You created this VI in Exercise 6-4.

❑ Set the front panel controls of the Pulse Train Generator VI with the following values:

- **Counter**: `Dev1/ctr1`
- **Duty Cycle**: `0.20`
- **Frequency**: `10`
- **Idle State**: `Low`
- **Initial Delay**: `0.000000`

❑ Run the Pulse Train Generator VI. This VI is now outputting a continuous pulse train.

How many seconds do you expect a pulse to remain high?

How many seconds do you expect a pulse to remain low?

2. Measure the pulse width of a pulse in the pulse train.

   ❑ On the Pulse Width Measurement VI, set the front panel controls to the following values:

   - **Counter**: `Dev1/ctr0`
   - **Starting Edge**: `Rising`
   - **Units**: `Seconds`
   - **Maximum Value**: `0.100000`
   - **Minimum Value**: `0.000001`

Note The signal you are measuring must start low when you are measuring a high pulse, so you must choose a rising starting edge when measuring a high pulse. Likewise, the signal must start high when you are measuring a low pulse, so you must choose a falling starting edge when measuring a low pulse.

   ❑ Run the Pulse Width Measurement VI.

   The VI should acquire a pulse width measurement of 0.02 seconds.

   ❑ On the Pulse Width Measurement VI, set the starting edge control to **Falling**.

   ❑ Run the Pulse Width Measurement VI.

   The VI should now acquire a pulse width measurement of 0.08 seconds.

   Do these measurements match your predictions in Step 1?

3. Stop the Pulse Train Generator VI when finished. Leave both VIs open for the next part of this exercise.

Part B: Period Measurement

Period measurements measure the time between consecutive rising edges of a pulse or consecutive falling edges of a pulse.

1. If necessary, open the Pulse Width Measurement VI in the `<Exercises>\DAQ and Signal Conditioning\Counter` directory.

2. Select **File»Save As»Copy»Substitute for original** and save the VI as `Period Measurement.vi` in the `<Exercises>\DAQ and Signal Conditioning\Counter` directory.

3. Modify the block diagram to measure period, as shown in Figure 6-14, using the following items:

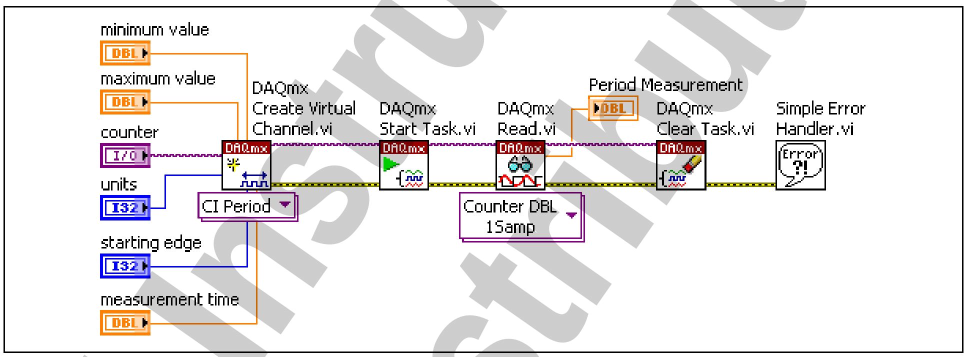


Figure 6-14. Period Measurement VI Block Diagram

   - ❑ Select the **Counter Input»Period** instance from the pull-down menu of the DAQmx Create Virtual Channel VI.
   - ❑ Right-click the measurement time input of the DAQmx Create Virtual Channel VI and select **Create»Control**.
   - ❑ Rename the Pulse Width indicator as `Period Measurement`.

4. Save the VI.

Test

1. Generate a 10 Hz pulse train that outputs a high signal 20% of the time and a low signal 80% of the time.

   - ❑ If necessary, open the Pulse Train Generator VI in the `<Exercises>\DAQ and Signal Conditioning\Counter` directory. You created this VI in Exercise 6-4.

❑ Set the front panel controls of the Pulse Train Generator VI with the following values:

- **Counter**: `Dev1/ctr1`
- **Duty Cycle**: `0.20`
- **Frequency**: `10`
- **Idle State**: `Low`
- **Initial Delay**: `0.000000`

❑ Run the Pulse Train Generator VI. This VI is now outputting a continuous pulse train.

How many seconds is the period of this 10 Hz pulse train?

2. Measure the period of a pulse in the pulse train.

❑ On the Period Measurement VI, set the front panel controls to the following values:

- **Counter**: `Dev1/ctr0`
- **Starting Edge**: `Rising`
- **Units**: `Seconds`
- **Maximum Value**: `0.100000`
- **Minimum Value**: `0.000001`
- **Measurement Time**: `0.05`

❑ Run the Period Measurement VI.

The VI should acquire a period measurement of 0.10 seconds.

Does this measurement match your prediction in Step 1?

❑ On the Pulse Width Measurement VI, set the starting edge control to **Falling**.

❑ Run the Period Measurement VI.

Does setting a starting edge of **Rising** or **Falling** change the period measurement for this signal? Why?

3. Stop the Pulse Train Generator VI and close all VIs when finished.

End of Exercise 6-6

Exercise 6-7 Frequency Measurement

Objective

To build a VI that measures frequency using a counter.

Scenario

This exercise explores three different methods of measuring frequency:

- Low Frequency with 1 Counter
- High Frequency with 2 Counters
- Large Range with 2 Counters

Part A: Low Frequency with 1 Counter Method

Use one counter that uses a constant timebase to measure the input signal.

BNC-2120 Configuration

1. Connect the **TTL Square Wave** BNC connector in the Function Generator section to the **USER 1** BNC connector in the user-defined signals section.
2. Use a wire to connect the **USER 1** screw terminal in the user-defined signals section to the Counter 0 Gate (**PFI 9**) screw terminal in the Timing I/O section.

Implementation

1. Open the Frequency Measurements - Low Frequency with 1 Counter VI in the `<Exercises>\DAQ and Signal Conditioning\Counter` directory.

2. Modify the block diagram of the VI to measure frequency using the low frequency with one counter method, as shown in Figure 6-13, using the following items:

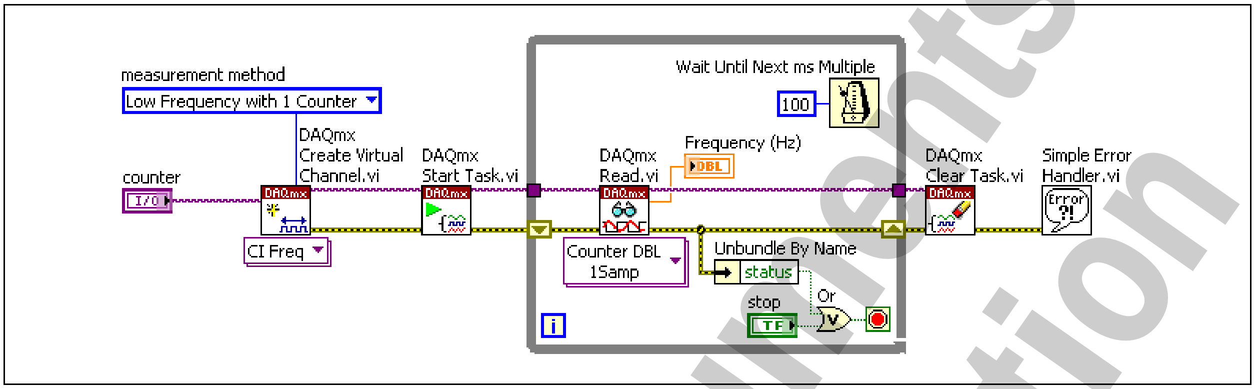
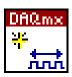


Figure 6-15. Frequency Measurements - Low Frequency with 1 Counter VI Block Diagram



- ❑ DAQmx Create Virtual Channel VI
  - Select the **Counter Input»Frequency** instance from the pull-down menu.
  - Right-click the counter input and select **Create»Control**.
  - Right-click the measurement method input and select **Create» Constant**. Set the constant to **Low Frequency with 1 Counter**.

- ❑ DAQmx Read VI
  - Select the **Counter»Single Sample»DBL** instance from the pull-down menu.
  - Right-click the data output and select **Create»Indicator**. Rename the indicator as `Frequency (Hz)`.

- ❑ Wait Until Next ms Multiple function
  - Right-click the millisecond multiple input and select **Create» Constant** from the shortcut menu. Enter `100` as the value for the constant.

3. Save the VI.

Test

1. Explore the quantization error for the Low Frequency with 1 Counter method.

    ❑ Open the *LabVIEW Help* by selecting **Help»Search the LabVIEW Help**.

    ❑ In the LabVIEW Help window, go to the **Search** tab.

    ❑ Type `quantization error` in the text field and click **List Topics**.

    ❑ Select the **Quantization Error** topic that appears and click **Display** to open the topic.

    ❑ Examine the table in the Quantization Error with One Counter Time Measurements section. Notice how the Actual Frequency of Input Signal and Counter Timebase Rate affect the Quantization Error.

    ❑ Leave the LabVIEW Help window open.

2. Measure the frequency of the TTL square wave signal using the Low Frequency with 1 Counter method.

    ❑ On the Frequency Measurements - Low Frequency with 1 Counter VI, set the front panel control to the following value:

    - **Counter**: `Dev1/ctr0`

    ❑ On the BNC-2120, set the **Frequency Selection** switch to the leftmost position. Turn the **Frequency Adjust** knob as low as possible. The function generator should now produce a 100 Hz signal.

    ❑ Run the VI. The Frequency (Hz) indicator should display approximately 100 Hz.

    ❑ Experiment with the VI by adjusting the **Frequency Adjust** knob and **Frequency Selection** switch.

3. Stop the VI.

Part B: High Frequency with 2 Counters Method

Uses two counters to count pulses of the signal to measure during the user-specified measurement time.

BNC-2120 Configuration

1. On the BNC-2120, use a wire to connect the **USER1** screw terminal in the user-defined signals section to the Counter 0 Source (**PFI 8**) screw terminal in the Timing I/O section.

Implementation

1. If necessary, open the Frequency Measurements - Low Frequency with 1 Counter VI located in the `<Exercises>\DAQ and Signal Conditioning\Counter` directory.

2. Select **File»Save As»Copy»Substitute copy for original** and save the VI as `Frequency Measurements - High Frequency with 2 Counters.vi` in the `<Exercises>\DAQ and Signal Conditioning\Counter` directory.

3. Modify the block diagram to use the High Frequency with 2 Counters method for acquiring frequency, as shown in Figure 6-16, by modifying the following item:

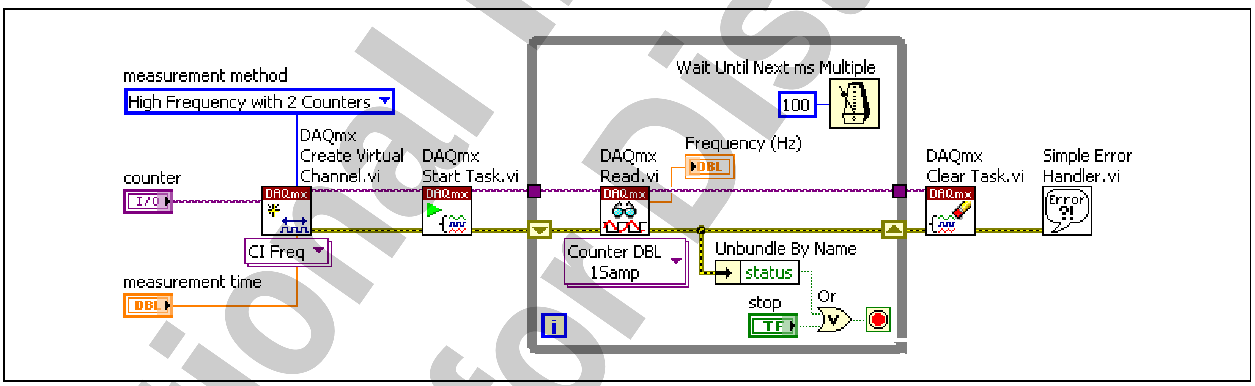


Figure 6-16. Frequency Measurement - High Frequency with 2 Counters VI Block Diagram

- ❑ DAQmx Create Virtual Channel VI
  - – Change the measurement method constant to **High Frequency with 2 Counters**.
  - – Right-click the measurement time input and select **Create» Control**.

4. Save the VI.

Test

1. Explore the quantization error for the High Frequency with 2 Counters method.

    ❑ Open the LabVIEW Help window showing the Quantization Error topic.

    ❑ Examine the table in the Quantization Error with High Frequency Two Counter Method section. Notice how the Actual Frequency of the Input Signal and Measurement Time affect the Quantization Error.

    ❑ Leave the LabVIEW Help topic window open.

2. Measure the frequency of the TTL square wave signal using the High Frequency with 2 Counters method.

    ❑ On the Frequency Measurements - High Frequency with 2 Counters VI, set the front panel controls to the following values:

    - **Counter**: `Dev1/ctr0`
    - **Measurement Time**: `1.0`

Note Measurement time is the length of time in seconds to measure the frequency or period of the signal if measurement method is High Frequency with 2 Counters. Measurement accuracy increases with increased measurement time and with increased signal frequency. If you measure a high-frequency signal for too long, however, the count register could roll over, which results in an incorrect measurement.

    ❑ On the BNC-2120, set the **Frequency Selection** switch to the rightmost position. Turn the **Frequency Adjust** knob as high as possible. The function generator should now produce a 1 MHz signal.

    ❑ Run the VI. The data indicator should display approximately 1 MHz.

    ❑ Experiment with the VI by adjusting the **Frequency Adjust** knob and **Frequency Selection** switch.

3. Stop the VI.

Part C: Large Range with 2 Counters Method

Use one counter to divide the frequency of the input signal by a divisor to create a lower-frequency signal that the second counter can more easily measure.

BNC-2120 Configuration

1. On the BNC-2120, use a wire to connect the **USER 1** screw terminal in the user-defined signals section to the Counter 0 Source (PFI 8) screw terminal in the Timing I/O section.

Implementation

1. Open the Frequency Measurements - High Frequency with 2 Counters VI located in the `<Exercises>\DAQ and Signal Conditioning\Counter` directory.

2. Select **File»Save As»Copy»Substitute copy for original** and save the VI as `Frequency Measurements - Large Range with 2 Counters.vi` in the `<Exercises>\DAQ and Signal Conditioning\Counter` directory.

3. Modify the block diagram to use the Large Range with 2 Counters method for acquiring frequency, as shown in Figure 6-17, by modifying the following item:

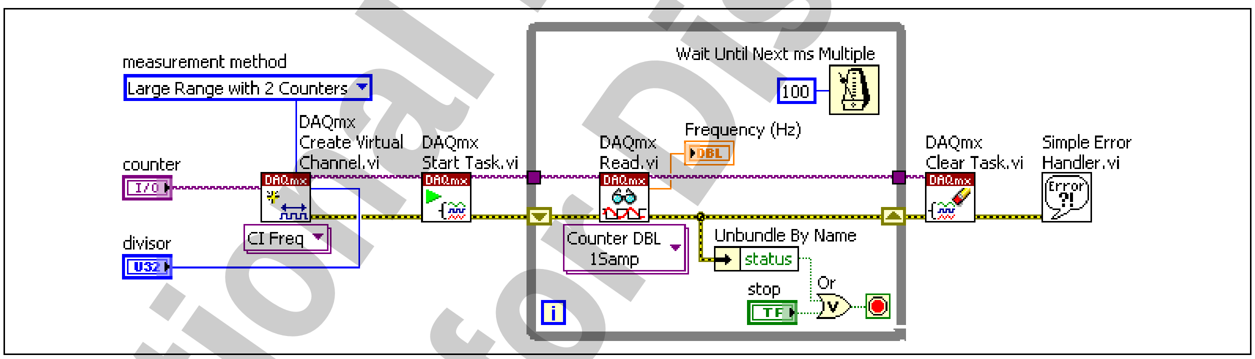


Figure 6-17. Frequency Measurement - Large Range with 2 Counters VI Block Diagram

- ❑ DAQmx Create Virtual Channel VI
  - – Change the measurement method constant to **Large Range with 2 Counters**.
  - – Delete the measurement time control.
  - – Right-click the divisor input and select **Create»Control**.

4. Save the VI.

Test

1. Explore the quantization error for the Low Frequency with 1 Counter method.

   ❑ Open the LabVIEW Help window showing the *Quantization Error* topic.

   ❑ Examine the table in the Quantization Error with Large Range Two Counter Measurement Method section. Notice how the Actual Frequency of the Input Signal and Divisor affect the Quantization Error.

   ❑ Leave the LabVIEW Help topic window open.

2. Measure the frequency of the TTL square wave signal using the Large Range with 2 Counters Method.

   ❑ On the Frequency Measurements - Large Range with 2 Counters VI, set the front panel controls to the following values:

   - **Counter**: `Dev1/ctr0`
   - **Divisor**: `4`

Note Divisor is the value by which to divide the input signal when measurement method is Large Range with 2 Counters. The larger the divisor, the more accurate the measurement. However, too large a value could cause the count register to roll over, which results in an incorrect measurement.

   ❑ On the BNC-2120, set the **Frequency Selection** switch to the rightmost position. Turn the **Frequency Adjust** knob as high as possible. The function generator should now produce a 1 MHz signal.

   ❑ Run the VI. The data indicator should display approximately 1 MHz.

   ❑ Experiment with the VI by adjusting the **Frequency Adjust** knob and **Frequency Selection** switch.

3. Stop the VI.

4. Close all VIs and the LabVIEW Help window when finished.

End of Exercise 6-7

Notes

Notes

7

Signal Conditioning Exercises

Exercise 7-1 Thermocouple Measurement

Objective

To read the temperature from a thermocouple with the NI 9219.

cDAQ Configuration

1. Connect a J-type thermocouple to **channel 1** of the NI 9219.

Implementation

1. Open the Thermocouple VI in the `<Exercises>\DAQ and Signal Conditioning\Signal Conditioning` directory.

2. Modify the block diagram of the Thermocouple VI to acquire temperature measurements from a thermocouple and monitor a user-specified temperature limit monitor, as shown in Figure 7-1, using the following items:

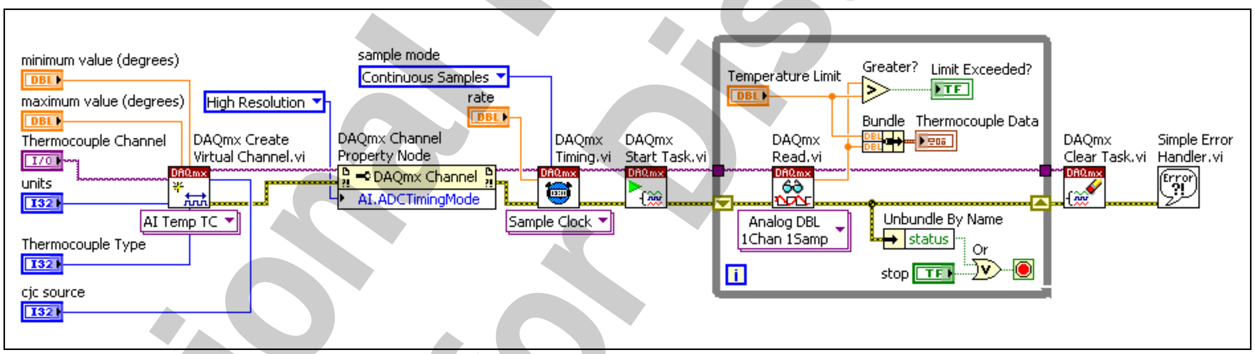
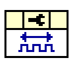


Figure 7-1. Thermocouple VI Block Diagram

- ❑ DAQmx Create Virtual Channel VI
  - Select the **Analog Input»Temperature»Thermocouple** instance from the pull-down menu.
  - Right-click the minimum value, maximum value, physical channels, units, thermocouple type, and cjc source inputs and select **Create»Control**.
  - Rename the minimum value control as `minimum value (degrees)`.
  - Rename the maximum value control as `maximum value (degrees)`.
  - Rename the physical channels control as `Thermocouple Channel`.

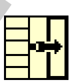


- ❑ DAQmx Channel Property Node—Use this node to filter out noise by setting the timing mode of the NI 9219.
  - Right-click the element of the Property Node and select **Select Property»Analog Input»General Properties» Digitizer/ADC»Timing Mode**.
  - Right-click the element and select **Change All to Write**.
  - Right-click the element input and select **Create»Constant**. Set the constant to **High Resolution**.

Note The NI 9219 supports four different timing options that are optimized for different types of applications by using different ADC conversion time. **High Speed** is optimized for high speed at the expense of noise rejection, **Best 60 Hz Rejection** is optimized for rejection of 60 Hz noise, **Best 50 hz Rejection** is optimized for rejection of 50 Hz noise, and **High Resolution** is optimized for maximum overall noise rejection and provides rejection of 50 Hz and 60 Hz noise.

The NI 9219 can acquire a new thermocouple measurement every 510 ms when using the High Resolution timing mode.

- ❑ Greater? function



- ❑ Bundle function

3. Arrange the front panel controls as shown and replace controls and indicators as needed. The completed front panel should resemble Figure 7-2.

Figure 7-2. Thermocouple VI Front Panel

4. Save the VI.

5. Set the front panel controls with the following values:

   - **Thermocouple Channel**: `cDAQ1Mod1/ai1`
   - **Thermocouple Type**: `J`
   - **CJC Source**: `Built-In`

Note Each channel of the NI 9219 has a built-in thermistor for cold-junction compensation (CJC) calculations. When cjc source is set to **Built-In**, the Thermocouple VI will use readings from the built-in thermistor to subtract the parasitic thermoelectric contributions from the cold junctions.

   - **Units**: `Deg C`

- **Minimum Value (degrees)**: `15.00`
- **Maximum Value (degrees)**: `35.00`

Note Input limits are the maximum and minimum values you expect to measure. Setting the input limits to a smaller range can improve your measurement. The DAQ device selects the best gain and range available according to the input limits you specify.

- **Sample Rate**: `1.0`
- **Temperature Limit**: `30`

Test

1. Test the VI.
   - ❑ Run the Thermocouple VI.
   - ❑ Place your finger or hand on the thermocouple to change the temperature.
   - ❑ Move the **Temperature Limit** control to experiment with the limit.
2. Stop the VI when you finish experimenting.

End of Exercise 7-1

Exercise 7-2 Strain Measurement

Objective

To acquire data from a strain gage connected to the NI 9219 and apply offset nulling for more accurate strain measurements.

cDAQ Configuration

1. Connect a 1/4 Bridge Strain Gage to **Channel 3** of the NI 9219. Connect one wire to **pin 3** and the other wire to **pin 5**.

Implementation

1. Open the Thermocouple VI located in the `<Exercises>\DAQ and Signal Conditioning\Signal Conditioning` directory.

2. Select **File»Save As»Copy»Substitute copy for original** and save the VI as `Strain.vi` in the `<Exercises>\DAQ and Signal Conditioning\Signal Conditioning` directory.

3. Modify the block diagram to measure strain, as shown in Figure 7-3 using the following items:

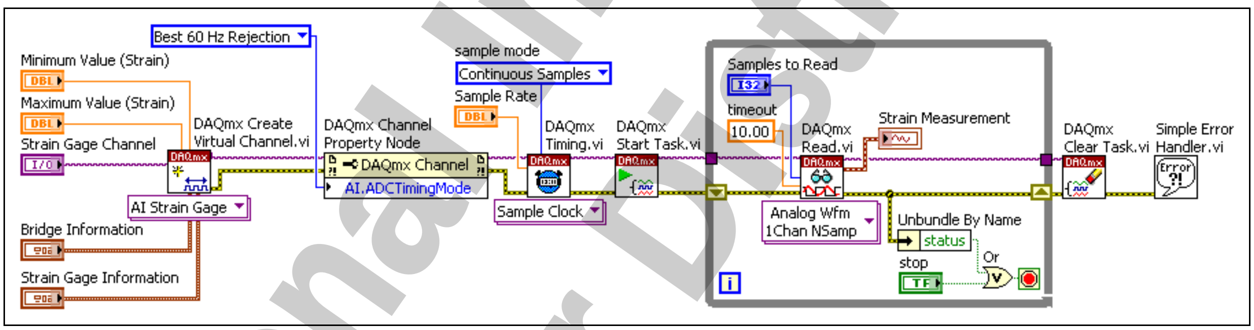


Figure 7-3. Strain VI Block Diagram

❑ Modify the DAQmx Create Virtual Channel VI.

– Select the **Analog Input»Strain»Strain Gage** instance from the pull-down menu.

– Rename the corresponding controls as `minimum value (strain)`, `maximum value (strain)`, and `Strain Gage Channel`.

– Delete the units, Thermocouple Type, and cjc source controls.

– Right-click the bridge information and strain gage information inputs and select **Create»Control**.

- ❑ Modify the DAQmx Channel Property Node.
  - – Change the AI.ADCTimingMode property to **Best 60 Hz Rejection**.

Note The NI 9219 supports four different timing options through the AI.ADC Timing Mode property. These options are optimized for different types of applications by using different ADC conversion times. **High Speed** is optimized for high speed at the expense of noise rejection, **Best 60 Hz Rejection** is optimized for rejection of 60 Hz noise, **Best 50 Hz Rejection** is optimized for rejection of 50 Hz noise, and **High Resolution** is optimized for maximum overall noise rejection and provides rejection of 50 Hz and 60 Hz noise.

The NI 9219 can acquire a measurement every 110 ms when there are no channels in Thermocouple mode and the timing mode is set to Best 60 Hz Rejection.

- ❑ Delete the Temperature Limit control, Greater? function, Bundle function, Limit Exceeded? indicator, Thermocouple Data indicator, and corresponding wires from the While Loop.
- ❑ Modify the DAQmx Read VI.
  - – Select the **Analog»Single Channel»Multiple Samples» Waveform** instance from the pull-down menu.
  - – Right-click the number of samples per channel input and select **Create»Control**. Rename the control as `Samples to Read`.
  - – Right-click the timeout input and select **Create»Constant**. Set the constant to `10`.
  - – Right-click the data output and select **Create»Indicator**. Rename the indicator as `Strain Measurement`.

4. Add offset nulling to the application. You will add code similar to Figure 7-4.

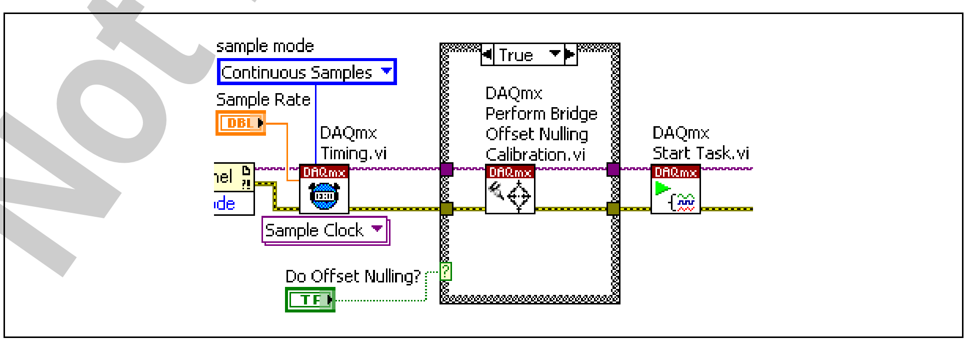
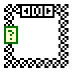


Figure 7-4. Offset Nulling Block Diagram Code

❑ Place the DAQmx Perform Bridge Offset Nulling Calibration VI between the DAQmx Timing VI and the DAQmx Start VI.



❑ Add a Case structure around the DAQmx Perform Bridge Offset Nulling Calibration VI so that you can compare uncalibrated and calibrated measurements. Notice that this VI is in the True case of the Case structure.

   – For the False case, wire the task and error wires through the Case structure.

   – Right-click the Case selector input of the Case structure and select **Create»Control**. Rename the control as `Do Offset Nulling?`.

5. Arrange the controls on the front panel in a logical manner, as shown in Figure 7-5.

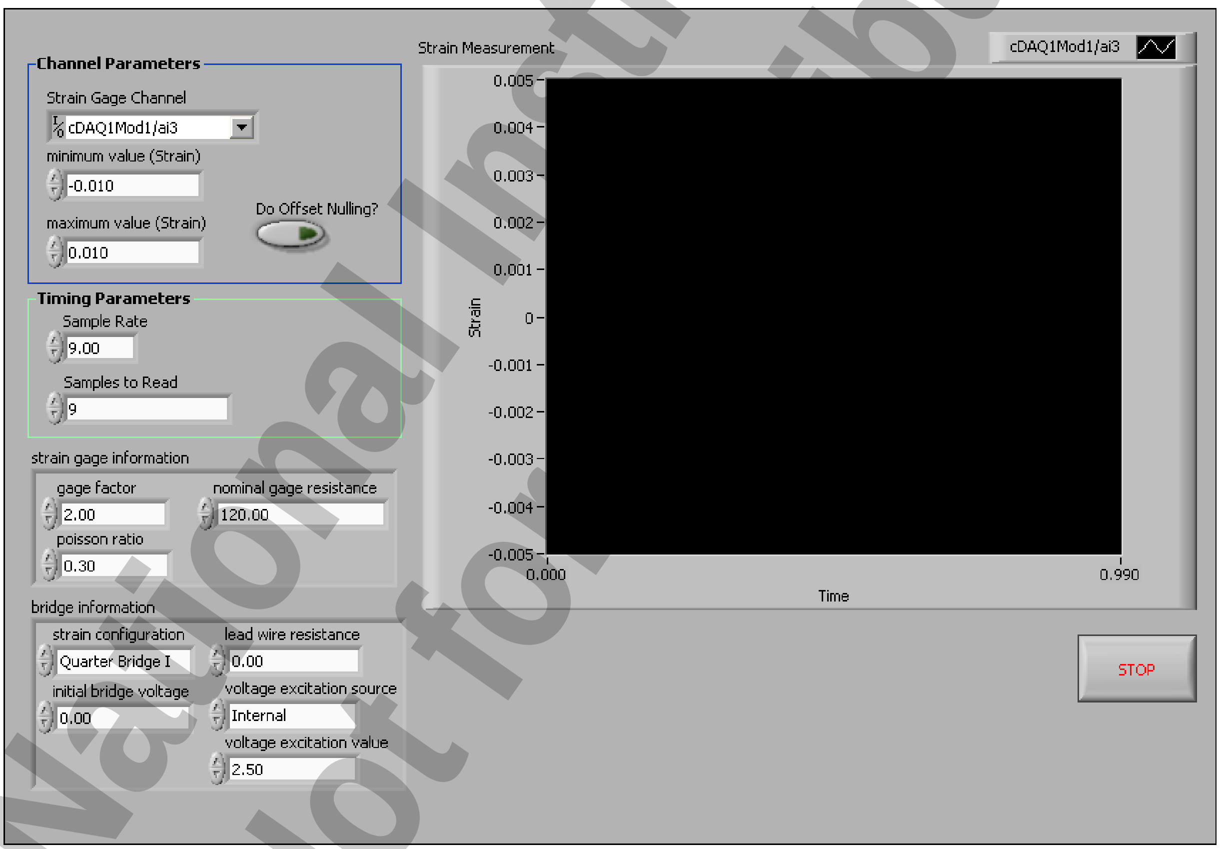


Figure 7-5. Strain VI Front Panel

- ❑ Create a waveform chart that is representative of the measurements being made.
  - – Right-click the Strain Measurement indicator and select **Replace»Graph»Waveform Chart**.
  - – Right-click the chart and select **Y scale»Autoscale Y** to disable autoscaling.
  - – Right-click the Y-axis and select **Formatting**.
  - – On the **Display Format** tab, set the following:
    - **Type**: `Floating Point`
    - **Digits**: `3`
    - **Precision Type**: `Digits of Precision`
  - – Rename the Y-axis `Strain`.
  - – Set the maximum and minimum values on the Y-axis to `0.005` and `-0.005` respectively.

6. Save the VI.

Test

1. Set the controls to the following values. If a control is not listed, then leave it at the default value.
   - **Strain Gage Channel**: `cDAQ1Mod1/ai3`
   - **Minimum Value (Strain)**: `-0.010`
   - **Maximum Value (Strain)**: `0.010`
   - **Sample Rate**: `9`
   - **Samples to Read**: `9`
   - **Nominal Gage Resistance**: `120`
   - **Strain Configuration**: `Quarter Bridge I`
2. Select **Edit»Make Current Values Default**.
3. Save the VI.

4. The first time you run the VI, leave Offset Nulling off by leaving the **Do Offset Nulling?** control in the `False` position.

5. Run the VI.

6. Bend the surface on which the strain gage is mounted.

   ❑ Is the measurement centered around 0 when not under stress?

   Most likely not, as it is very uncommon to have an accurate initial strain reading. Because of that you must perform offset nulling.

7. Stop the VI.

8. Set the **Do Offset Nulling?** control to `True`.

9. Run the VI.

10. Observe the differences between running with and without offset nulling.

11. Stop and close the VI.

End of Exercise 7-2

Notes

8

Synchronization Exercises

Exercise 8-1 Simultaneously Started Analog Input and Output

Objective

To use two different methods to simultaneously start an analog input and output operations.

BNC-2120 Configuration

1. Connect the BNC connector of channel 0 in the Analog Outputs section to the BNC connector of channel 2 in the Analog Inputs section.

2. Verify that the **FS/GS** switch underneath the BNC connector of channel 2 is in the GS position.

3. In the Timing I/O section, wire the **PULSES** or **CLK** terminal to the **PFI1** terminal.

Note Some models of the BNC-2120 have a **PULSES** terminal and some models have a **CLK** terminal instead. They are functionally the same.

Part I: Software Triggered Method

1. Open the Simultaneous Start VI in the <Exercises>\DAQ and Signal Conditioning\Synchronization directory.

2. Examine the block diagram.

   Notice that the VIs in the top row correspond to a continuous buffered analog input task and display the acquired data on the data waveform chart.

   Notice that the VIs in the bottom row correspond to a continuous buffered analog output task that outputs the waveform generated by the Sine Waveform VI.

3. Modify the VI to start two tasks at the same time without the use of a hardware trigger, as shown in Figure 8-1, using the following items:

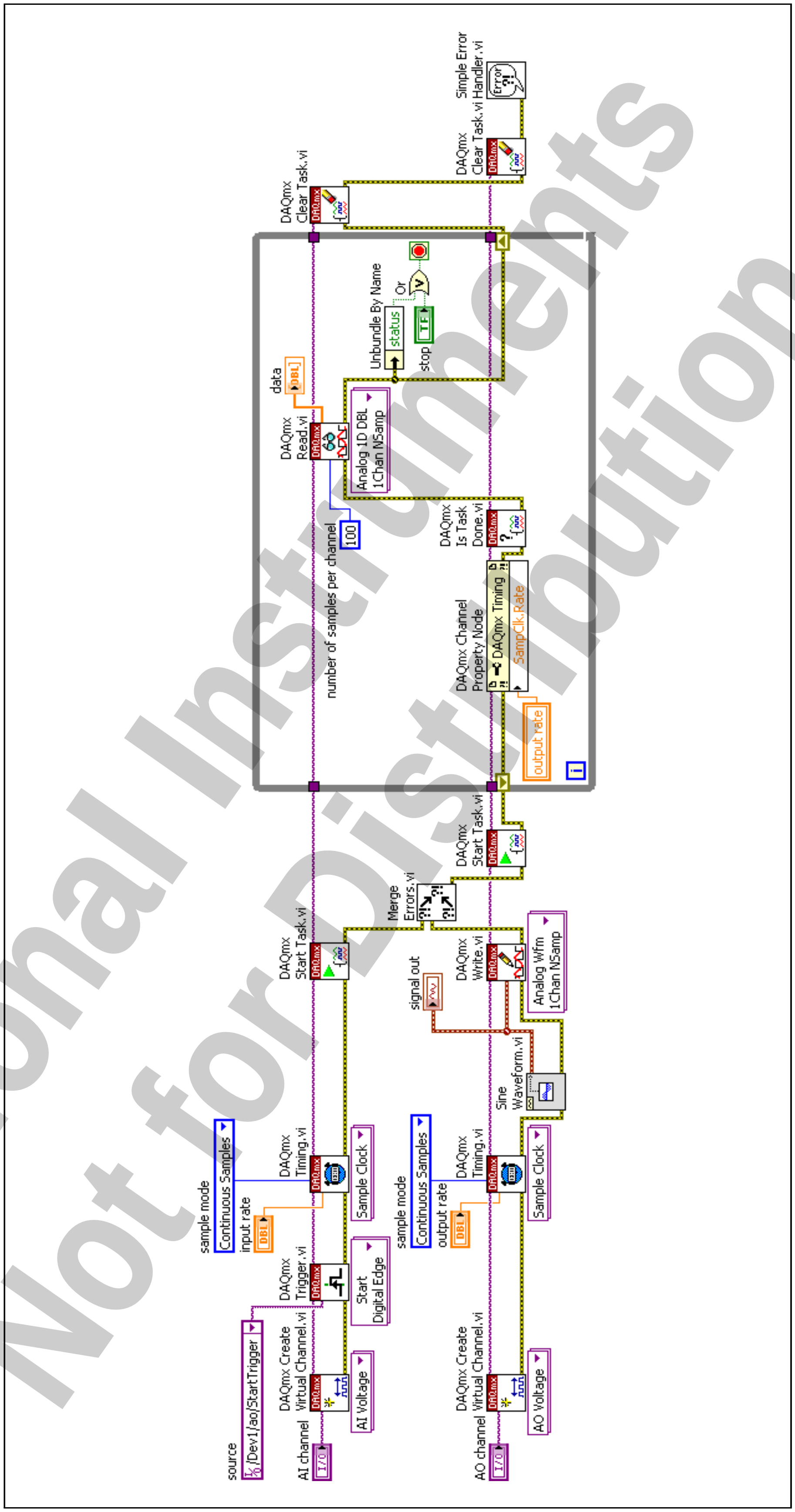
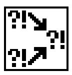


Figure 8-1. Simultaneous Start VI Block Diagram

- ❑ DAQmx Trigger VI
  - – Select the **Start»Digital Edge** instance from the pull-down menu.
- ❑ To ensure that the acquisition does not begin before the output, you must make the analog input task start before the analog output task, but be dependent upon the start of the analog output task.
  - – Create a constant for the DAQmx Trigger VI source and make it dependent upon the start of the analog output by setting the constant to `/Dev1/ao/StartTrigger`.



  - – Place a Merge Errors VI between the DAQmx Start task for the analog input and analog output tasks.
- ❑ Local Variable for Output Rate of DAQmx Timing VI
  - – Right-click the output rate control and select **Create»Local Variable**.
  - – Right-click the local variable and select **Change to Read**.
- ❑ Wire and arrange the block diagram as shown in Figure 8-1.

4. Save the VI.

5. Arrange the items on the front panel, as shown in Figure 8-2.

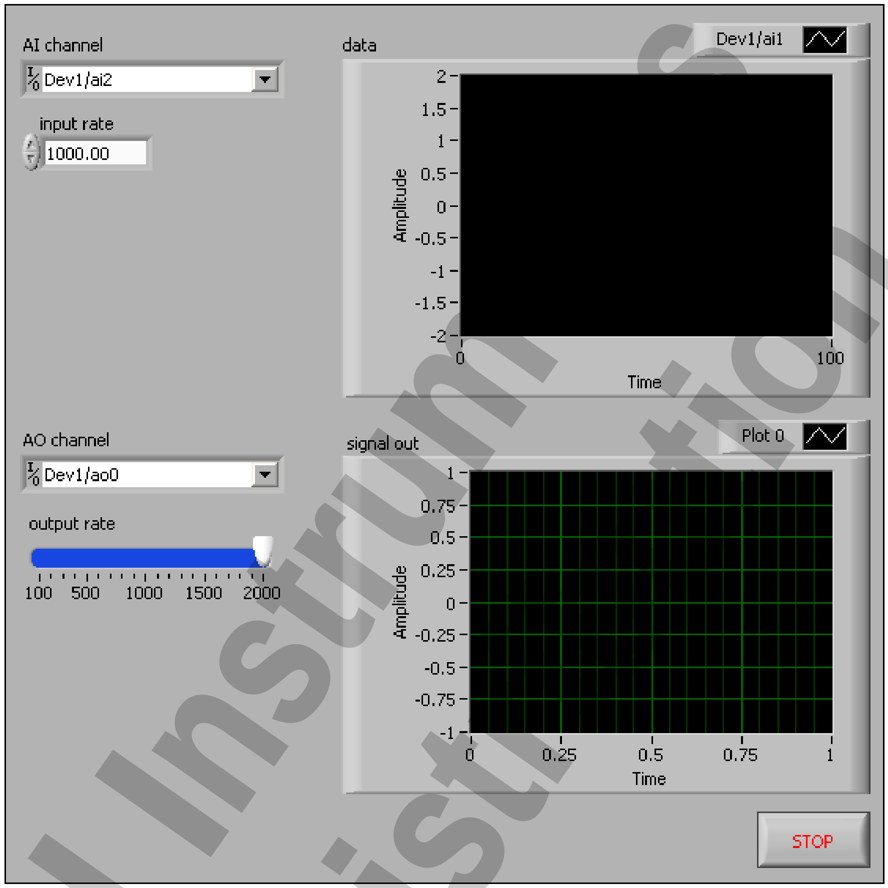


Figure 8-2. Simultaneous Start VI Front Panel

Test

1. Set the front panel controls for the Simultaneous Start VI with the following values:
   - **AI Channel**: `Dev1/ai2`
   - **Input Rate**: `1000`
   - **AO Channel**: `Dev1/ao0`
   - **Output Rate**: `1000`
2. Run the VI.

   To the human eye, these measurements appear to start at approximately the same time, but they do not behave any differently than if you simply started both tasks at the same time without linking the triggers. However, on a hardware level, both tasks begin on the exact same clock pulse.
3. Change the output rate control to increase and decrease the clock speed for the output.

Note Making the output rate and input rate different is possible because while the measurements are started at the same time, they are not required to run at the same rate.

4. Stop the VI.

Part II: Hardware Triggered Method

1. Add a hardware trigger to start two tasks simultaneously, as shown in Figure 8-3, by adding the following item:

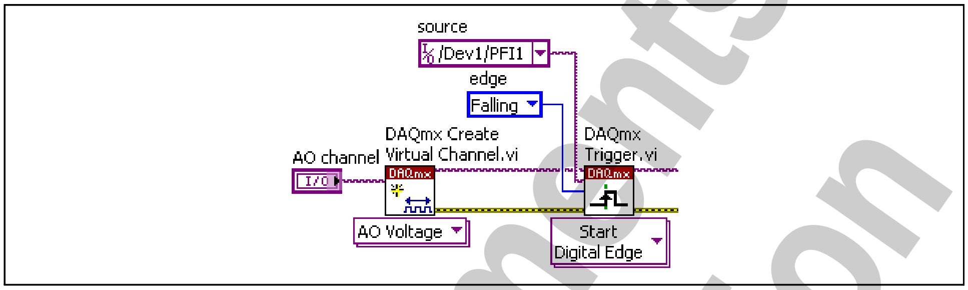


Figure 8-3. Added DAQmx Trigger VI to Analog Output Task

❑ Insert a DAQmx Trigger VI into the analog output task

– Select the **Start»Digital Edge** instance from the pull-down menu.

– Right-click the source and edge inputs and select **Create» Constant**.

– Set the source constant to `/Dev1/PFI1`.

– Set the edge constant to **Falling** so that is triggers when the Quadrature Encoder knob on the BNC-2120 is rotated.

2. Save the VI as a Copy and Substitute the copy for the original. Name the VI as `Simultaneous Start with Hardware Trigger.vi` in the `<Exercises>\DAQ and Signal Conditioning\Synchronization` directory.

Test

1. Run the VI.

2. Turn the Quadrature Encoder knob on the BNC-2120.

   Remember that when you turn the Quadrature Encoder knob, it produces a falling edge on the **PULSES** or **CLK** terminal which is connected to the PFI 1 line. The falling edge received on the PFI 1 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. Leave this VI open for the next exercise.

End of Exercise 8-1

Exercise 8-2 Synchronous Analog Input and Output

Objective

Synchronize the analog input acquisition on a cDAQ device to the analog output generation on a separate DAQ device.

BNC-2120 Configuration

1. In the Timing I/O section, wire the **PFI 5** terminal to the **USER 1** terminal in the user-defined signals section.

cDAQ Configuration

1. Connect the **TRIG 0** BNC connector on the cDAQ-9178 chassis to the **USER 1** BNC connector on the BNC-2120.

Implementation

1. If necessary, open `<Exercises>\DAQ and Signal Conditioning\Synchronization\Simultaneous Start with Hardware Trigger.vi`.

2. Save the VI as `<Exercises>\DAQ and Signal Conditioning\Synchronization\Multi-Device Synchronous AI and AO.vi`.

3. Modify the block diagram, as shown in Figure 8-4, to share the analog output sample clock on the Dev1 DAQ device with an analog input thermocouple task on the cDAQ device using the following items:

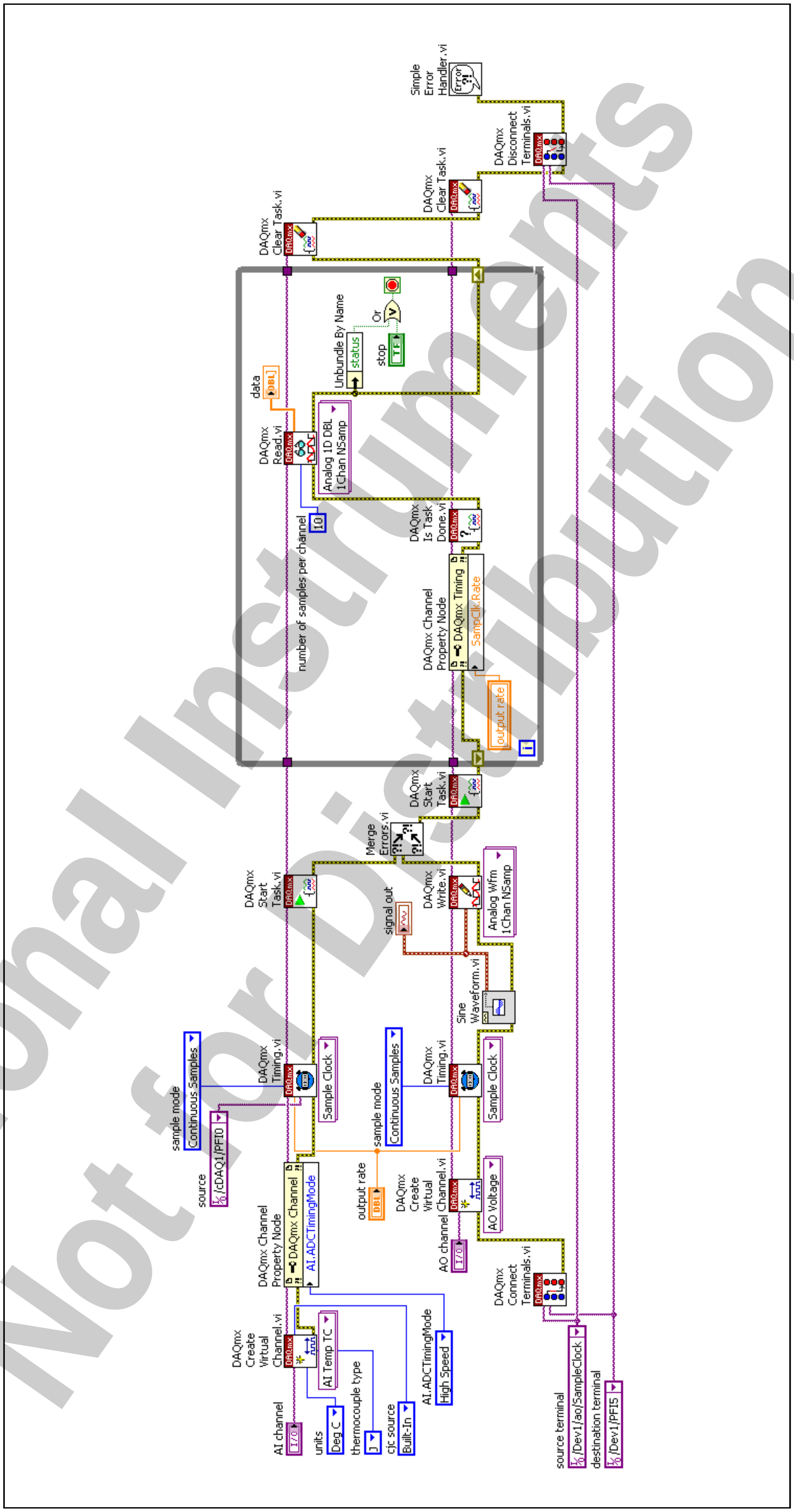
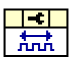


Figure 8-4. Multi-Device Synchronous AI and AO VI Block Diagram

- ❑ Delete the DAQmx Trigger VIs and all associated controls and constants.
- ❑ Delete the input rate control.
- ❑ Wire the output rate control to the rate inputs of both DAQmx Timing VIs.
- ❑ Modify the analog input DAQmx Create Virtual Channel VI to create a thermocouple task.
  - Select the **Analog Input»Temperature»Thermocouple** instance from the pull-down menu.
  - Right-click the units, thermocouple type, and cjc source inputs and select **Create»Constant**. Set these constants to **Deg C**, **J**, and **Built-In**.



- ❑ Add a DAQmx Channel Property Node.
  - Right-click the element of the Property Node and select **Select Property»Analog Input»General Properties» Digitizer/ADC»Timing Mode**.
  - Right-click the element and select **Change All to Write**.
  - Right-click the element input and select **Create»Constant**. Set the constant to **High Speed**.

Note The NI 9219 supports four different timing options that are optimized for different types of applications by using different ADC conversion time. **High Speed** is optimized for high speed at the expense of noise rejection, **Best 60 Hz Rejection** is optimized for rejection of 60 Hz noise, **Best 50 hz Rejection** is optimized for rejection of 50 Hz noise, and **High Resolution** is optimized for maximum overall noise rejection and provides rejection of 50 Hz and 60 Hz noise.

The NI 9219 can acquire a new thermocouple measurement every 20 ms when using the High Speed timing mode.

- ❑ Add a DAQmx Connect Terminals VI

  – Right-click the source terminal input and select **Create»Constant**. Set the constant to `/Dev1/ao/SampleClock`.

  – Right-click the destination terminal input and select **Create»Constant**. Set the constant to `/Dev1/PFI5`.

  When you run this VI, the PFI5 line will output the analog output sample clock signal.

- ❑ On the analog input task DAQmx Timing VI, right-click the source input and select **Create»Constant**. Set the constant to `/cDAQ1/PFI0`.

  In the BNC-2120 Configuration section, you have essentially connected the PFI5 line of Dev1 to the TRIG0 BNC connector of the cDAQ chassis. Because PFI5 will output the analog output sample clock signal of Dev1 and Trig0 is the PFI0 line of the cDAQ chassis, you can now synchronize tasks running on the cDAQ device to an analog output task on Dev1.

- ❑ On the DAQmx Read VI, change the number of samples per channel constant to `10`.

- ❑ Add a DAQmx Disconnect Terminals VI between the DAQmx Clear Task and Simple Error Handler VI.

4. Arrange the front panel controls and indicators as shown in Figure 8-5.

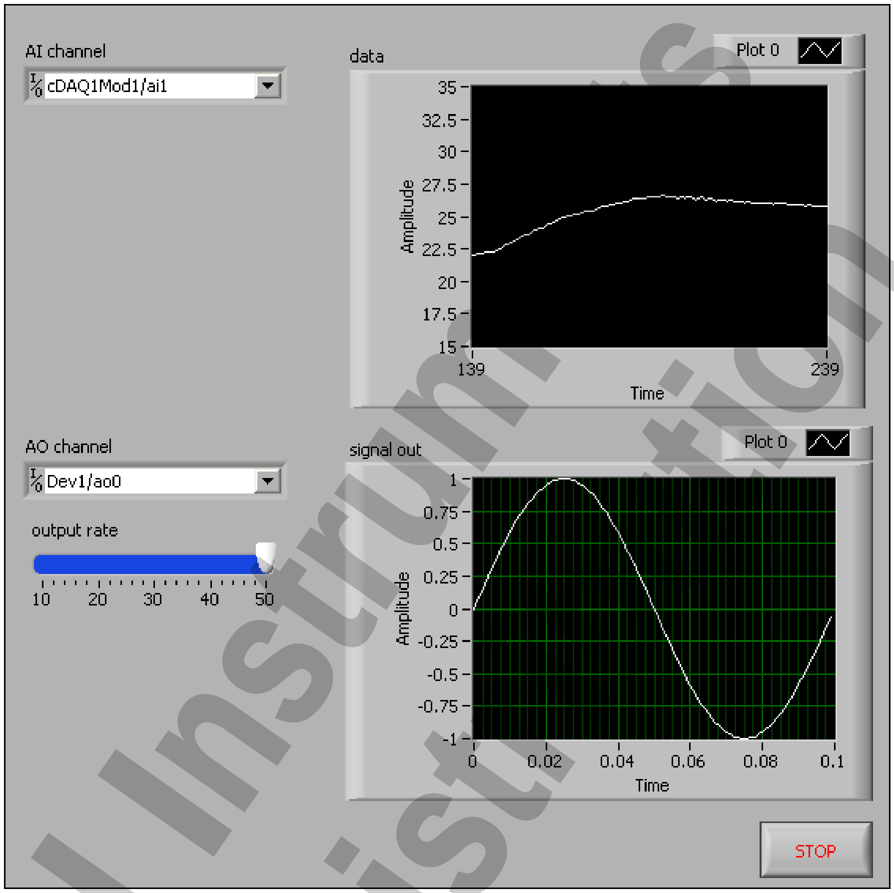


Figure 8-5. Multi-device Synchronous AI and AO VI Front Panel

- ❑ Modify the output rate indicator to allow a range of `10` to `50`.
- ❑ Modify the data waveform chart to show Y-axis values from `15` to `35`.

5. Save the VI.

Test

1. Set the front panel controls with the following values:

   - **AI channel**: `cDAQ1Mod1/ai1`
   - **AO channel**: `Dev1/ao0`
   - **Output Rate**: `50`

2. Run the VI.

3. Adjust the output rate control while changing the temperature. To continue changing the temperature, alternate holding and releasing the thermocouple.

   The analog input thermocouple task running on the cDAQ device is configured to acquire 10 samples at a time. Notice that when the output rate of the analog output task on the Dev1 device is slower, the analog input thermocouple task on the cDAQ device takes longer to acquire 10 samples because the analog input rate is synchronized with the analog output rate.

4. Stop and close the VI when finished.

End of Exercise 8-2

Exercise 8-3 Self-Study: Retriggerable Analog Input for STC2-Based Devices

Objective

To synchronize an analog input task with the internal input of a counter.

Scenario

Triggered analog input tasks for STC2-based devices can normally only receive one trigger and then read data. However, counter tasks can create retriggerable pulse trains.

BNC-2120 Configuration

1. In the Timing I/O section, wire the **PULSES** or **CLK** screw terminal to the **PFI 1** screw terminal.

Note Some models of the BNC-2120 have a **PULSES** terminal and some models have a **CLK** terminal instead. They are functionally the same.

2. Connect the **Sine/Triangle** BNC connector in the Function Generator section to the BNC connector of channel 2 in the Analog Inputs section.

3. Verify that the **Sine/Triangle** switch is in the `Sine` position.

4. Verify that the **FS/GS** switch underneath the BNC connector of channel 2 is in the `GS` position.

Implementation

1. Open `<Exercises>\DAQ and Signal Conditioning\Synchronization\Retriggerable Analog Input (STC2).vi`.

2. Examine the block diagram.

   Notice that the VIs in the top row correspond to an analog input task. Notice that the VIs in the bottom row correspond to a retriggerable pulse train task like the one you created in Exercise 6-5.

3. Modify the block diagram to use the retriggerable counter pulse train as the sample clock for the analog input task, as shown in Figure 8-6, using the following items:

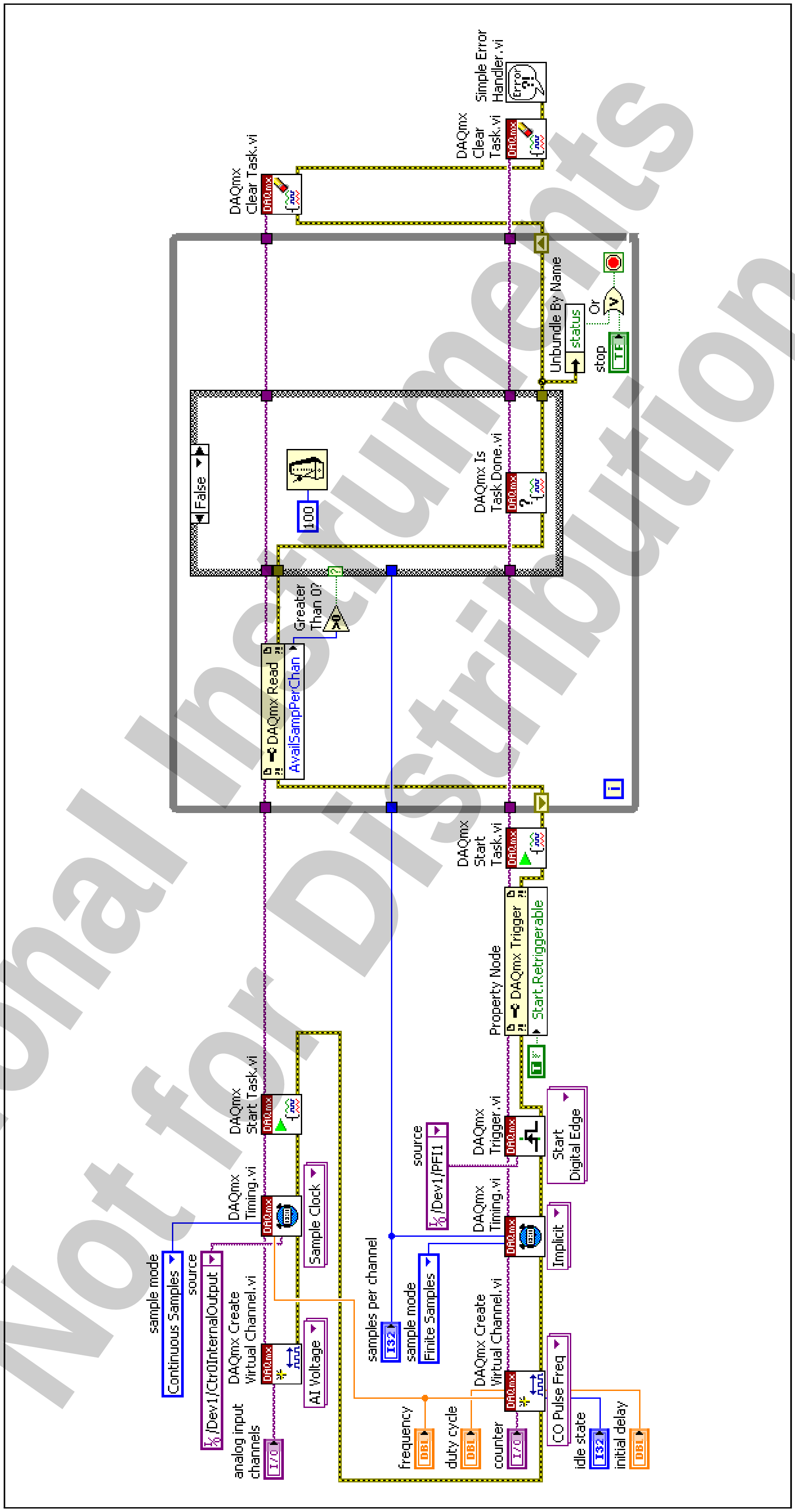


Figure 8-6. Retriggerable Analog Input (STC2)

- ❑ DAQmx Timing VI
  - – Select the **Sample Clock** instance from the pull-down menu.
  - – Wire the frequency control to the rate input of the DAQmx Timing VI.
  - – Right-click the sample mode input and select **Create»Constant**. Set the constant to **Continuous Samples**.
  - – Right-click the source input and select **Create»Constant**.
  - – Right-click the source constant and select **I/O Name Filtering**. Enable the **Include Advanced Terminals** option and click **OK**.
  - – Set the source constant to `/Dev1/Ctr0InternalOutput`.
- ❑ DAQmx Read Property Node
  - – Right-click the element of the Property Node and select **Select Property»Status»Available Samples Per Channel**.
- ❑ Greater Than 0? function

4. Create the True case of the case structure, as shown in Figure 8-7, so that the application only attempts to read if samples are available using the DAQmx Read VI.

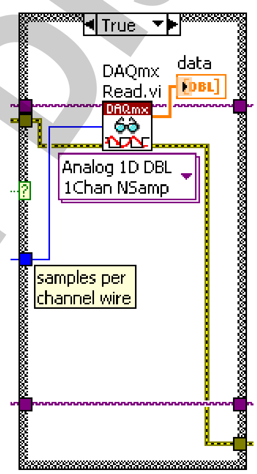


Figure 8-7. True Case of Case Structure

- ❑ DAQmx Read VI
  - – Select the **Analog»Single Sample»Multiple Samples» 1D DBL** instance from the pull-down menu.

5. Arrange the items on the front panel, as shown in Figure 8-8.

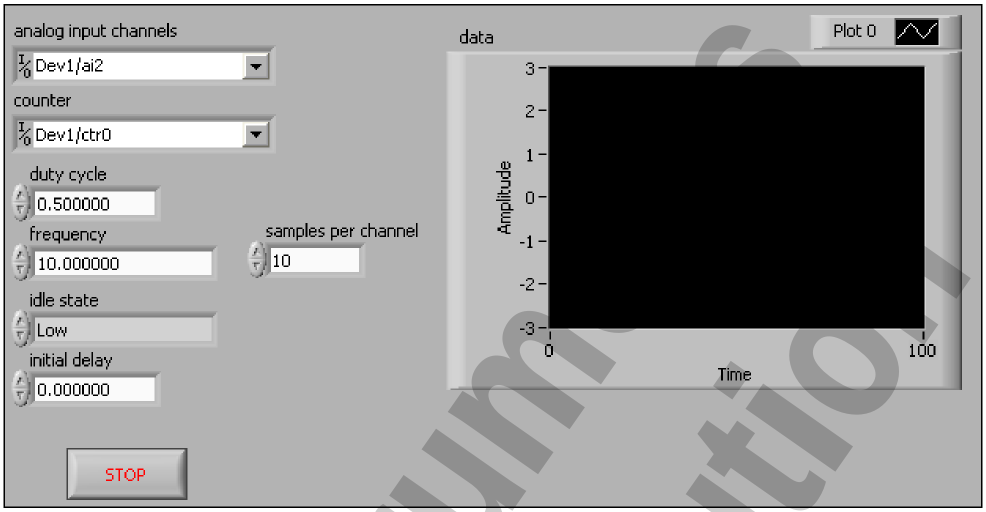


Figure 8-8. Retriggerable Analog Input (STC2) VI Front Panel

6. Save the VI.

Test

1. Set the front panel controls to the following values:
   - **Analog input channel**s: `Dev1/ai2`
   - **Counter**: `Dev1/ctr0`
   - **duty cycle**: `0.5`
   - **Frequency**: `10`
   - **Samples per channel**: `10`
   - **Idle state**: `Low`
   - **Initial Delay**: `0`
2. Run the VI.

   ❑ Notice how there are no samples in the graph. The counter output task is waiting to detect a falling edge on the `/Dev1/PFI1` channel. The analog input task is waiting its sample clock signal from the counter output.

- ❑ Turn the Quadrature Encoder knob to acquire 10 samples on the analog input channel.

  Because the samples per channel is set to 10 and the frequency is set to 10 Hz, it will take 1 second for the VI to acquire the samples after you turn the Quadrature Encoder knob.

  Every time you turn the Quadrature Encoder knob, you will acquire another 10 samples on the analog input channel because the analog input task is using a retriggerable finite counter output as the sample clock.

3. Stop and close the VI when finished.

Retriggerable Analog Input using a STC3-Based Device

In certain STC3-based DAQ devices, you can create a retriggerable analog input task without using a counter task. You only need to use a DAQmx Trigger VI and DAQmx Trigger Property Node with the **Start»More» Retriggerable** property set to true.

1. Open `<Solutions>\DAQ and Signal Conditioning\Synchronization\Retriggerable Analog Input (STC3).vi`.

2. Examine the block diagram and notice that you can use the Retriggerable property with an analog input task when using a STC3-based DAQ device.

3. Close the VI when finished.

End of Exercise 8-3

Notes

A

Signal Processing Exercises

Exercise A-1 Power Spectrum

Objective

To build a VI that determines the power spectrum of a generated signal.

Scenario

The FFT is a powerful computational method you can use to perform a spectral analysis on discrete sampled data such as that from a DAQ system. This exercise uses LabVIEW Express VIs to generate a signal and perform spectral analysis of the data.

Implementation

1. Open a blank VI and build the block diagram shown in Figure A-1 using the following items:

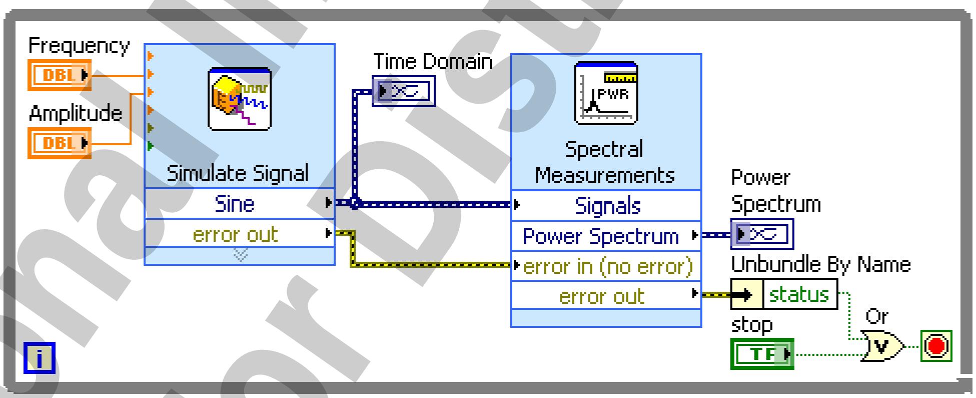
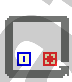
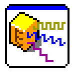


Figure A-1. Power Spectrum Express VI Block Diagram



❑ While Loop

– Create the stop control from the Loop Condition terminal of the While Loop.



- ❑ Simulate Signal Express VI—This Express VI creates an output signal of a specified type. In the **Configure Simulate Signals** dialog box that appears, select the settings shown in Figure A-2.

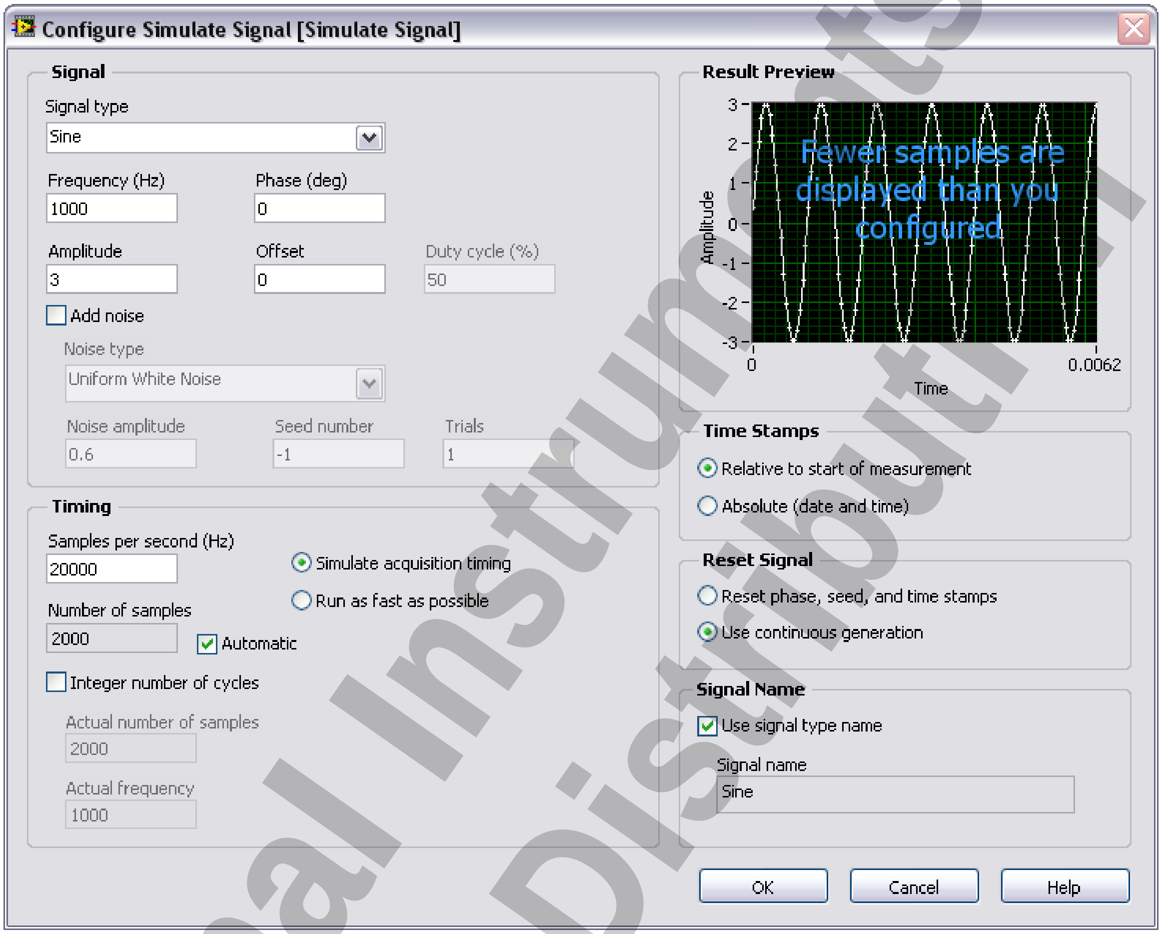


Figure A-2. Configure Simulate Signal Dialog

- – Click **OK** to close the dialog box.
- – On the block diagram, resize the Simulate Signals Express VI to display the **error out** element.
- – Right-click the **Frequency** input and select **Create»Control**.
- – Right-click the **Amplitude** input and select **Create»Control**.
- – Right-click the sine output and select **Create»Graph Indicator**. Rename this indicator as `Time Domain`.

❑ Spectral Measurements Express VI—This Express VI computes the power spectrum of a signal. In the **Configure Spectral Measurements** dialog box that displays, select the settings shown in Figure A-3.

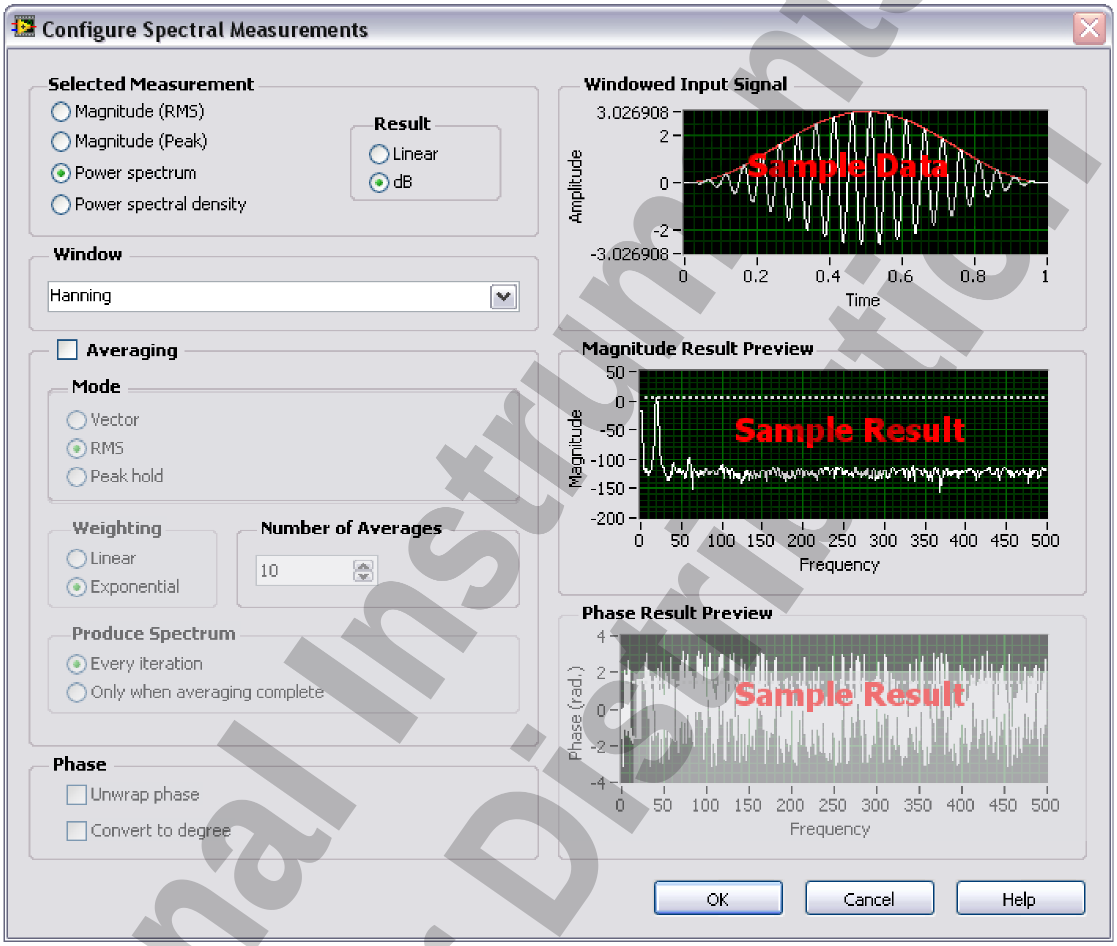
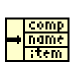


Figure A-3. Configure Spectral Measurements Dialog

– Click **OK** to close the dialog box.

– On the block diagram, resize the Spectral Measurements Express VI to display the **error in (no error)** and **error out** elements.

– Right-click the Power Spectrum output and select **Create» Graph Indicator** from the shortcut menu. Rename this indicator as `Power Spectrum`.



❑ Unbundle by Name function

❑ Or function

2. Arrange the front panel as shown in Figure A-4.

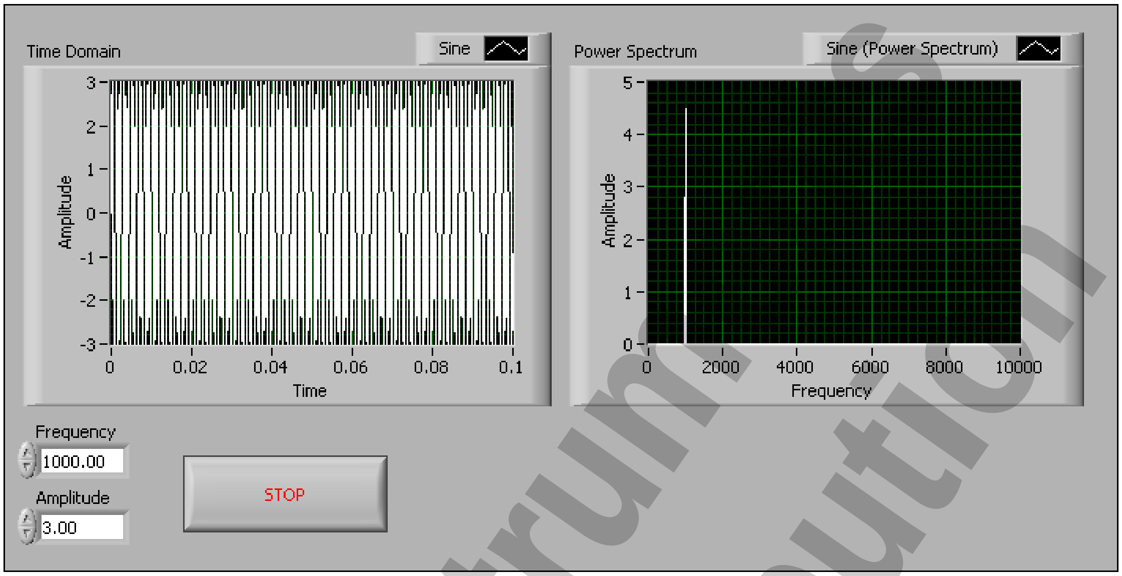


Figure A-4. Power Spectrum Express VI Front Panel

3. Save the VI as `Power Spectrum Express.vi` in the `<Exercises>\DAQ and Signal Conditioning` directory.

Test

1. Set the front panel controls of the Power Spectrum Express VI with the following values:
   - **Frequency**: `1000`
   - **Amplitude**: `3`
2. Run the VI.
3. Change the frequency that the Simulate Signal Express VI generates. Notice how the peak moves on the spectrum plot to reflect the frequency generated.
4. Save and close the VI.

End of Exercise A-1

Exercise A-2 Window Comparison VI

Objective

To build a VI to compare the effect of different window functions upon the spectral leakage and low power frequency components of a signal.

Scenario

In this exercise, two sine waves of different amplitudes are added together and transformed into the frequency domain. Sine wave 1 has a much smaller amplitude than sine wave 2. Windowing is applied which changes the shape of the signal and affects the power spectrum that you see. The frequency-domain plot shows the results so you can compare the effect of different window functions.

Implementation

1. Open a blank VI and build the block diagram shown in Figure A-5 using the following items:

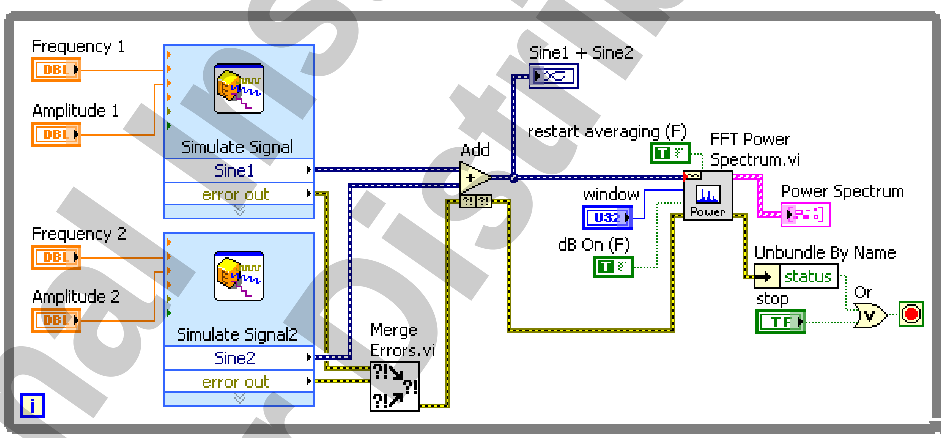
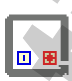
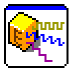


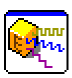
Figure A-5. Window Comparison VI Block Diagram



- ❑ While Loop
  - – Create the stop control from the Loop Condition terminal of the While Loop.

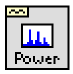


- ❑ Simulate Signal Express VI
  - – In the **Configure Simulate Signals** dialog box, configure the settings as follows:
    - **Samples per Second (Hz)**: `1000`
    - Disable the **Automatic** checkbox
    - **Number of Samples**: `1000`
    - Disable the **Use signal type name** checkbox
    - Rename the signal as `Sine1`
  - – Click **OK**.
  - – On the block diagram, resize the Simulate Signals Express VI to display the **error out** element.
  - – Right-click the frequency and amplitude terminals and select **Create»Control**. Rename the controls as `Frequency 1` and `Amplitude 1`.

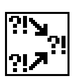


- ❑ Simulate Signal Express VI
  - – In the **Configure Simulate Signals** dialog box, configure the settings as follows:
    - **Samples per Second (Hz)**: `1000`
    - Disable the **Automatic** checkbox
    - **Number of Samples**: `1000`
    - Disable the **Use signal type name** checkbox
    - Rename the signal as `Sine2`
  - – Click **OK**.
  - – On the block diagram, resize the Simulate Signals Express VI to display the **error out** element.
  - – Right-click the frequency and amplitude terminals and select **Create»Control**. Rename the controls as `Frequency 2` and `Amplitude 2`.

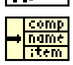
- ❑ Add function—This polymorphic function computes the sum of the two dynamic data input signals into a single output.
  - Right-click the output of the Add function and select **Create» Graph Indicator**. Rename the graph indicator as `Sine 1 + Sine 2`.



- ❑ FFT Power Spectrum VI—This VI performs an FFT on an input signal and computes the averaged auto power spectrum of the time signal.
  - Right-click the window input and select **Create»Control**.
  - Right-click the dB On (F) input and select **Create»Constant**. Set the constant to **True**.
  - Right-click the restart averaging (F) input and select **Create» Constant**. Set the constant to **True**.
  - Wire the Sine 1 + Sine 2 graph indicator to the time signals input of the FFT Power Spectrum VI.
  - Right-click the power spectra output and select **Create»Graph Indicator**. Rename the graph indicator as `Power Spectrum`.



- ❑ Merge Errors VI



- ❑ Unbundle by Name function

- ❑ Or function

2. Arrange the front panel as shown in Figure A-6.

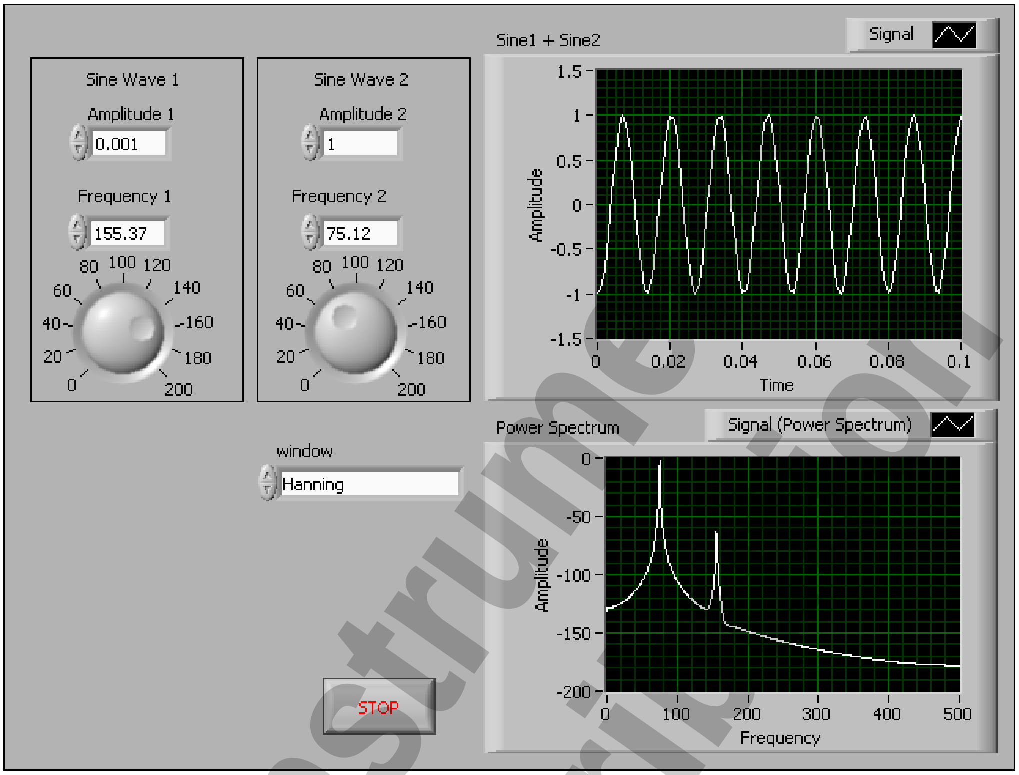


Figure A-6. Window Comparison VI Front Panel

- ❑ Replace the Power Spectrum array with a waveform graph.
- ❑ Replace the Frequency 1 and Frequency 2 numeric controls with knob numeric controls. Right-click each knob numeric control and select **Visible Items»Digital Display**.

3. Save the VI as `Window Comparison.vi` in the `<Exercises>\DAQ and Signal Conditioning` directory.

Test

1. Set the front panel controls of the Window Comparison VI with the following values:
   - **Amplitude 1**: `0.001`
   - **Frequency 1**: `155.37`
   - **Amplitude 2**: `1.00`
   - **Frequency 2**: `75.12`
   - **Window**: `Hanning`

2. Run the VI.

3. Select the Flat Top window and notice the shape of the signal in the power spectrum plot. The Flat Top window consists of more cosine terms than any other window. The Flat Top window is hence used for accurate single tone amplitude measurements.

4. Compare different window functions by choosing another window from the front panel and observe the changes in the power spectrum.

5. Experiment with the frequency values and observe the changes in the power spectrum.

6. Stop and close the VI.

End of Exercise A-2

Exercise A-3 DAQ Digital IIR Filtering

Objective

To use digital filtering in a DAQ Application

Scenario

It is common to use filtering on data that has been collected with a DAQ device. In this exercise, you will build a VI that performs digital IIR filtering on data acquired from a DAQ device. The VI compares the effects of filtering. You can view the waveform in the time domain and frequency domain of the nonfiltered and filtered waveforms to observe the advantages of digital IIR filtering.

BNC-2120 Configuration

1. Connect the **Sine/Triangle** BNC connector in the Function Generator section to the BNC connector of channel 2 in the Analog Inputs section.

2. Verify that the **Sine/Triangle** switch is in the `Sine` position.

3. Verify that the **FS/GS** switch underneath the BNC connector of channel 2 is in the `GS` position.

Implementation

1. Open a blank VI and build the block diagram shown in Figure A-7 using the following items:

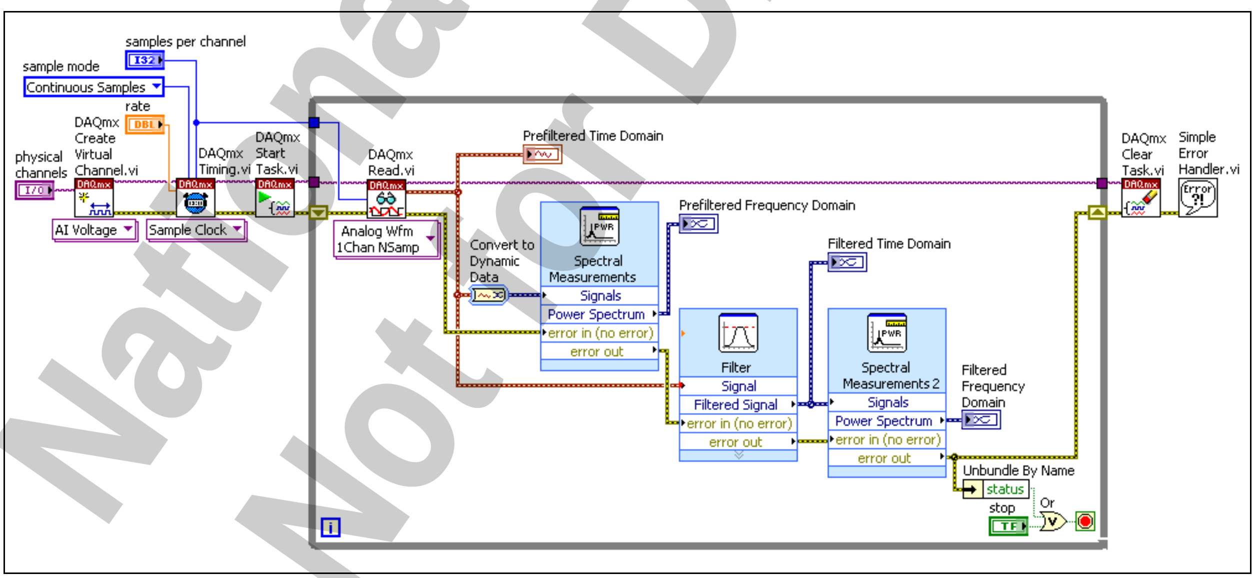
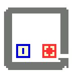


Figure A-7. IIR Digital Filter VI Block Diagram

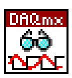
- ❑ DAQmx Create Channel VI
  - – Right-click the physical channels input and select **Create»Control**.

- ❑ DAQmx Timing VI
  - – Right-click the sample mode input and select **Create»Constant** from the shortcut menu. Set the constant to `Continuous Samples`.
  - – Right-click the samples per channel and rate inputs and select **Create»Control** from the shortcut menu.

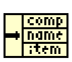
- ❑ DAQmx Start Task VI



- ❑ While Loop
  - – Create the stop control from the Loop Condition terminal of the While Loop.



- ❑ DAQmx Read VI
  - – Select the **Analog»Single Channel»Multiple Samples»Waveform** instance from the pull-down menu.
  - – Right-click the data output and select **Create»Indicator**. Rename the indicator as `Prefiltered Time Domain`.



- ❑ Unbundle by Name function

- ❑ Or function

- ❑ Convert to Dynamic Data—Converts numeric, Boolean, waveform, and array data types to the dynamic data type for use with Express VIs.
  - – In the configuration dialog that appears, set the Input data type to **Single Waveform**.
  - – Click **OK**.

❑ Spectral Measurements Express VI

– In the **Configure Spectral Measurements** dialog box, configure the settings as shown in Figure A-8.

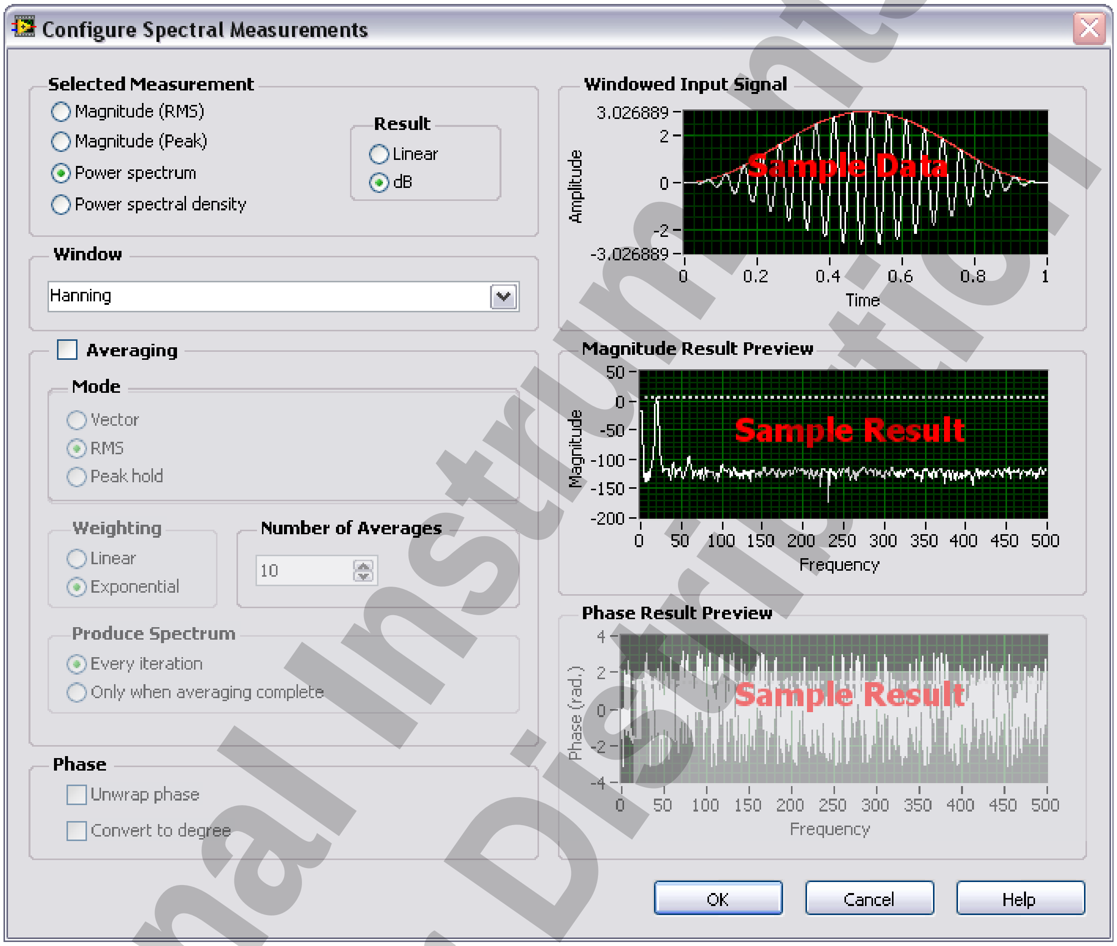
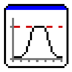


Figure A-8. Configure Spectral Measurements Dialog

– Click **OK**.

– On the block diagram, resize the Spectral Measurements Express VI to display the **error in (no error)** and **error out** elements.



- ❑ Filter Express VI—This Express VI processes signals through filters and windows.
  - – In the **Configure Filter** dialog box, set the options as shown in Figure A-9.

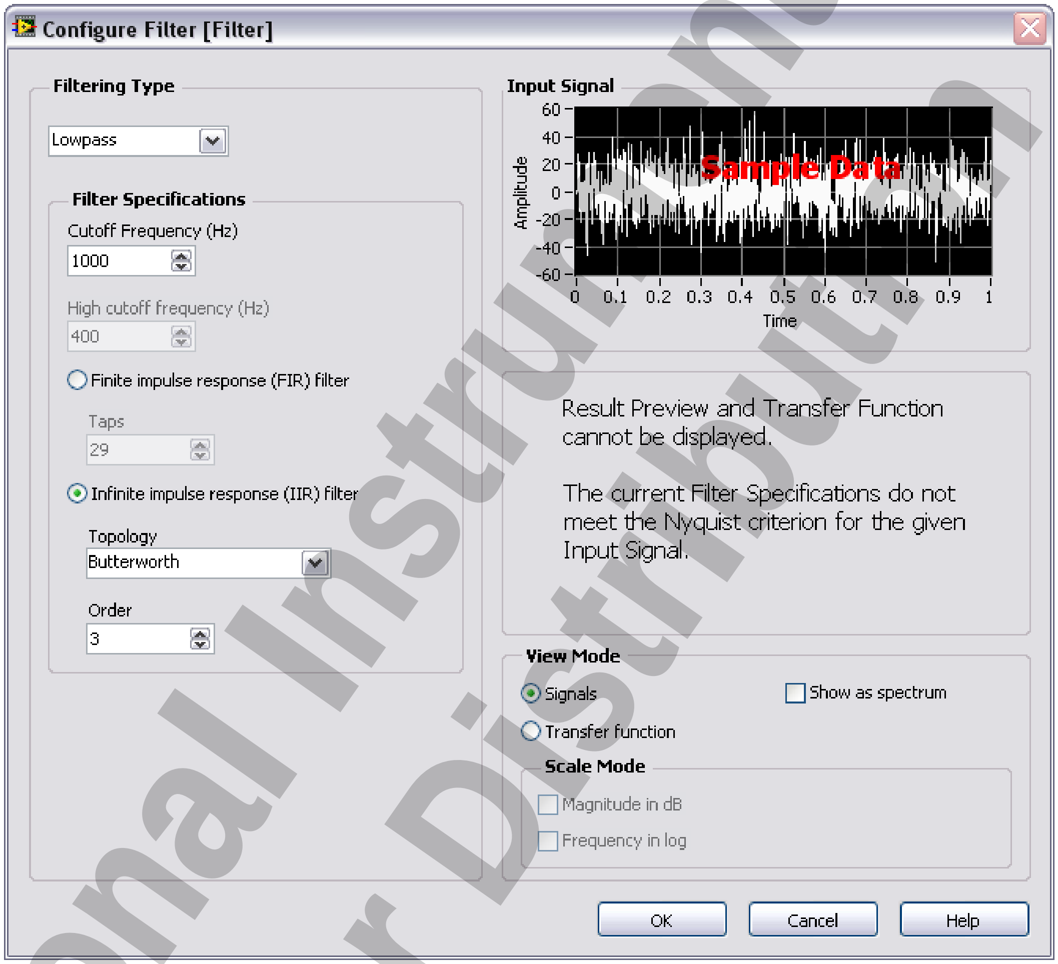


Figure A-9. Configure Filter Dialog

  - – Click **OK**.
  - – On the block diagram, resize the Spectral Measurements Express VI to display the **error in (no error)** and **error out** elements.

- ❑ Create a copy of the Spectral Measurements Express VI you configured earlier and add it to the block diagram, as shown in Figure A-7.

- ❑ DAQmx Clear Task VI

- ❑ Simple Error Handler VI on the block diagram. In the event of an error, this VI displays a dialog box with information regarding the error and where it occurred.

- ❑ Wire the block diagram as shown in Figure A-7. After the Express VIs are wired together, create the following graph indicators:

  – Right-click the Power Spectrum output on the first instance of the Spectral Measurements Express VI and select **Create»Graph Indicator**. Rename the indicator as `Prefiltered Frequency Domain`.

  – Right-click the Filtered Signal output on the Filter Express VI and select **Create»Graph Indicator**. Rename the indicator as `Filtered Time Domain`.

  – Right-click the Power Spectrum output on the second instance of the Spectral Measurements Express VI and select **Create»Graph Indicator**. Rename the indicator as `Filtered Frequency Domain`.

2. Arrange the front panel as shown in Figure A-10.

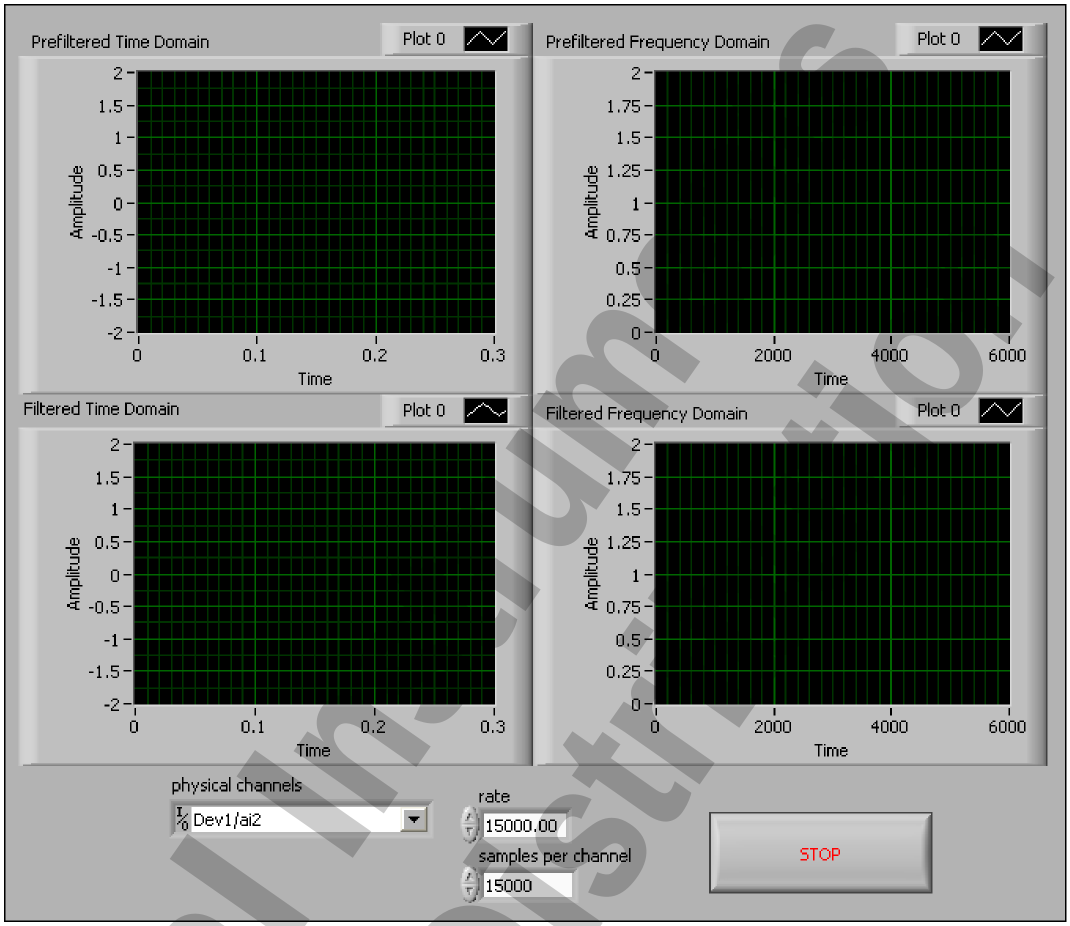


Figure A-10. IIR Digital Filter VI Front Panel

- ❑ Replace the Prefiltered Time Domain array with a waveform graph.
- ❑ Right-click each of the two Time Domain charts and select **Y scale» Autoscale Y** to disable autoscaling. Set the scale at `-1.25` to `1.25`.

3. Save the VI as `IIR DAQ Filter.vi` in the `<Exercises>\DAQ and Signal Conditioning` directory.

Test

1. Output a sine wave with a frequency of 100 Hz from the function generator on the BNC-2120.

   - ❑ On the BNC-2120, set the Frequency Selection switch in the function generator section to the leftmost position.
   - ❑ Turn the Frequency Adjust dial on the BNC-2120 to `LO`.

2. Set the front panel controls of the Continuous Buffered Acquisition VI with the following values:

   - **Physical Channels**: `Dev1/ai2`
   - **Rate**: `15000`
   - **Samples per Channel**: `15000`

3. Run the IIR DAQ Filter VI. Notice the four waveforms that are graphed.

   In the Prefiltered Time Domain waveform graph, the acquired data is a very discrete sine wave because digital data generates a discrete analog waveform. The generated sine wave contains very high frequency components beyond the fundamental. These high frequency components appear in the Prefiltered Frequency Domain waveform graph. The lowpass Digital IIR filter attenuates these high frequency components with the almost perfect sine wave in the Filtered Time Domain waveform graph and Filtered Frequency Domain graph. Always be careful with aliasing. A digital filter will not remove aliasing.

4. Change the frequency by slowly rotating the Frequency Adjust knob clockwise. You will see a predominant peak at around 1000 Hz.

   Aliasing is a problem with digital filtering. As you increase the frequency beyond 1000 Hz, the signal attenuates in the filtered waveform graphs. However, the signal is never completely attenuated. The pseudo-analog waveform that the example VI generates contains a lot of high frequency information. Because there are no hardware anti-aliasing filters, the high frequency components are aliased between 0 and 5 kHz. The digital IIR filter attenuates these high frequency aliased components but never completely eliminates them. This emphasizes the importance of hardware anti-aliasing filters.

5. Lower the cutoff frequency to 500 Hz while keeping a Butterworth filter topology and filter order of 5.

   ❑ Stop the VI and view the block diagram.

   ❑ Double-click the Filter Express VI to open the configuration dialog.

   ❑ Set **Cutoff Frequency (Hz)** at `500`.

   The input signal is not completely attenuated. The Butterworth filter transition region gets steeper as the order increases.

   ❑ Increase the **Order** value, and the filter begins to attenuate more of the signal. Increase the **Order** value until you are satisfied that the filter has attenuated the signal enough.

   A typical filter order is about 1 to 20, with 30 being an upper limit. With a higher order filter, the computer needs more processing time, and the chance increases of introducing floating point errors into the output.

   ❑ Click **OK**.

6. Run the VI.

   ❑ Experiment with different filter topologies, filter orders, and cutoff frequencies.

7. Stop and close the VI.

End of Exercise A-3

Notes

B

Additional Information and Resources

This appendix contains additional information about National Instruments technical support options and LabVIEW resources.

National Instruments Technical Support Options

Visit the following sections of the award-winning National Instruments Web site at `ni.com` for technical support and professional services:

- **Support**—Technical support at `ni.com/support` includes the following resources:
  - **Self-Help Technical Resources**—For answers and solutions, 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.
  - **Standard Service Program Membership**—This program entitles members to direct access to NI Applications Engineers via phone and email for one-to-one technical support as well as exclusive access to on demand training modules via the Services Resource Center. NI offers complementary membership for a full year after purchase, after which you may renew to continue your benefits.

    For information about other technical support options in your area, visit `ni.com/services` or contact your local office at `ni.com/contact`.
- **System Integration**—If you have time constraints, limited in-house technical resources, or other project challenges, National Instruments Alliance Partner members can help. The NI Alliance Partners joins system integrators, consultants, and hardware vendors to provide comprehensive service and expertise to customers. The program ensures qualified, specialized assistance for application and system development. To learn more, call your local NI office or visit `ni.com/alliance`.

If you searched `ni.com` and could not find the answers you need, contact your local office or NI corporate headquarters. Phone numbers for our worldwide offices are listed at the front of this manual. You also can visit the

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

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.

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.

LabVIEW Resources

The following publications offer more information about LabVIEW.

LabVIEW Publications

Many books have been written about LabVIEW programming and applications. The National Instruments Web site contains a list of all the LabVIEW books and links to places to purchase these books. Publisher information is also included so you can directly contact the publisher for more information on the contents and ordering information for LabVIEW and related computer-based measurement and automation books.

Web Sites

`ni.com/products`

`ni.com/labview`

`ni.com/dataacquisition`

`ni.com/sensors`

`ni.com/signalconditioning`

Course Evaluation

Course ______________________________

Location ______________________________

Instructor ____________________ Date ____________________

Student Information (optional)

Name ______________________________

Company ____________________ Phone ____________________

Instructor

Please evaluate the instructor by checking the appropriate circle.	Unsatisfactory	Poor	Satisfactory	Good	Excellent
Instructor's ability to communicate course concepts	❍	❍	❍	❍	❍
Instructor's knowledge of the subject matter	❍	❍	❍	❍	❍
Instructor's presentation skills	❍	❍	❍	❍	❍
Instructor's sensitivity to class needs	❍	❍	❍	❍	❍
Instructor's preparation for the class	❍	❍	❍	❍	❍

Course

Training facility quality	❍	❍	❍	❍	❍
Training equipment quality	❍	❍	❍	❍	❍

Was the hardware set up correctly? ❍ Yes ❍ No

The course length was ❍ Too long ❍ Just right ❍ Too short

The detail of topics covered in the course was ❍ Too much ❍ Just right ❍ Not enough

The course material was clear and easy to follow. ❍ Yes ❍ No ❍ Sometimes

Did the course cover material as advertised? ❍ Yes ❍ No

I had the skills or knowledge I needed to attend this course. ❍ Yes ❍ No If no, how could you have been better prepared for the course? ______________________________

____

What were the strong points of the course? ______________________________

____

What topics would you add to the course? ______________________________

____

What part(s) of the course need to be condensed or removed? ______________________________

____

What needs to be added to the course to make it better? ______________________________

____

How did you benefit from taking this course? ______________________________

____

Are there others at your company who have training needs? Please list. ______________________________

____

____

Do you have other training needs that we could assist you with? ______________________________

____

How did you hear about this course? ❍ NI Web site ❍ NI Sales Representative ❍ Mailing ❍ Co-worker

❍ Other ______________________________